Roger Dingledine
Moria Research Labs
1558 Massachusetts Ave #24
Cambridge, MA 02138

August 3, 2006

Broadcasting Board of Governors
International Broadcasting Bureau
Office of Engineering
Cohen Building, Room 4300
330 Independence Avenue, SW
Washington, DC 20237
Attn: Demetria Anderson [redacted]

Dear Demetria Anderson,

Below is my first invoice for contract BBGCON1806S6149. There are no travel costs.

Please do not hesitate to mail me at [redacted] or call me at [redacted] if there are any questions or problems.

Period	Hours	Rate	Cost
May 24 – July 24	2 months	$10000/mo	$20000
Cumulative	2 months	$10000/mo	$20000

Sincerely yours,

Roger Dingledine
Owner, Moria Research Labs

Roger Dingledine
Moria Research Labs
1558 Massachusetts Ave #24
Cambridge, MA 02138
[redacted]

September 25, 2006

Broadcasting Board of Governors
International Broadcasting Bureau
Office of Engineering
Cohen Building, Room 4300
330 Independence Avenue, SW
Washington, DC 20237
Attn: Demetria Anderson [redacted]

Dear Demetria Anderson,

Below is my second invoice for contract BBGCON1806S6149, Fiscal Data 9568-06-0206-E009601067-454000-4335-2544. There are no travel costs.

Services rendered include a new stable update for Tor; a new development snapshot for Tor; continued work on designing incentive schemes and anti-blocking schemes; meetings with various developers and activists in the U.S. and Germany; and new development work including improved resource management in Tor servers and improved auto-detection of server addresses.

Please do not hesitate to mail me at [redacted] or call me at [redacted] if there are any questions or problems.

Invoice #2:

Period	Months	Rate	Cost
July 25 – September 24	2 months	$10000/mo	$20000

Sincerely yours,

Roger Dingledine
Owner, Moria Research Labs

Roger Dingledine
Moria Research Labs
1558 Massachusetts Ave #24
Cambridge, MA 02138
[redacted]

November 30, 2006

Broadcasting Board of Governors
International Broadcasting Bureau
Office of Engineering
Cohen Building, Room 4300
330 Independence Avenue, SW
Washington, DC 20237
Attn: Demetria Anderson [redacted]

Dear Demetria Anderson,

Below is my fourth invoice for contract BBGCON1806S6149, Fiscal Data 9568-06-0206-E009601067-454000-4335-2544. There are no travel costs.

Services rendered include a new stable update for Tor (0.1.1.25); a new development snapshot for Tor (0.1.2.3-alpha); design and a first draft of the blocking-resistance Tor paper; and early development work to support Tor controllers that want to track the reasons for circuit failures.

Please do not hesitate to mail me at [redacted] or call me at [redacted] if there are any questions or problems.

Invoice #4:

Period	Months	Rate	Cost
October 25 – November 24	1 month	$27500/mo	$27500

Sincerely yours,

Roger Dingledine
Owner, Moria Research Labs

Roger Dingledine
Moria Research Labs
1558 Massachusetts Ave #24
Cambridge, MA 02138
[redacted]

January 5, 2007

Broadcasting Board of Governors
International Broadcasting Bureau
Office of Engineering
Cohen Building, Room 4300
330 Independence Avenue, SW
Washington, DC 20237
Attn: Demetria Anderson [redacted]

Dear Demetria Anderson,

Below is my fifth invoice for contract BBGCON1806S6149, Fiscal Data 9568-06-0206-E009601067-454000-4335-2544. There are no travel costs.

Services rendered include a new stable update for Tor (0.1.1.26); a new development snapshot for Tor (0.1.2.4-alpha); continued research on the blocking-resistance design; and early development work to support transparent proxy connections through Tor.

Please do not hesitate to mail me at [redacted] or call me at [redacted] if there are any questions or problems.

Invoice #5:

Period	Months	Rate	Cost
November 25 – December 24	1 month	$27500/mo	$27500

Sincerely yours,

Roger Dingledine
Owner, Moria Research Labs

Roger Dingledine
Moria Research Labs
1558 Massachusetts Ave #24
Cambridge, MA 02138
[redacted]

February 5, 2007

Broadcasting Board of Governors
International Broadcasting Bureau
Office of Engineering
Cohen Building, Room 4300
330 Independence Avenue, SW
Washington, DC 20237
Attn: Demetria Anderson [redacted]

Dear Demetria Anderson,

Below is my sixth invoice for contract BBGCON1806S6149, Fiscal Data 9568-06-0206-E009601067-454000-4335-2544. There are no travel costs.

Services rendered include new development snapshots for Tor (0.1.2.5-alpha and 0.1.2.6-alpha); continued research on the blocking-resistance design; and early development work to support encrypted directory connections through Tor.

Please do not hesitate to mail me at [redacted] or call me at [redacted] if there are any questions or problems.

Invoice #6:

Period	Months	Rate	Cost
December 25 – January 24	1 month	$27500/mo	$27500

Sincerely yours,

Roger Dingledine
Owner, Moria Research Labs

1 Tor's congestion control does not work well

One of Tor's critical performance problems is in how it combines high-volume streams with low-volume streams. We need to come up with ways to let the "quiet" streams (like web browsing) co-exist better with the "loud" streams (like bulk transfer).

1.1 TCP backoff slows down every circuit at once

Tor combines all the circuits going between two Tor relays into a single TCP connection. This approach is a smart idea in terms of anonymity, since putting all circuits on the same connection prevents an observer from learning which packets correspond to which circuit. But over the past year, research has shown that it's a bad idea in terms of performance, since TCP's backoff mechanism only has one option when that connection is sending too many bytes: slow it down, and thus slow down all the circuits going across it.

We could fix this problem by switching to a design with one circuit per TCP connection. But that means that a relay with 1000 connections and 1000 circuits per connection would need a million sockets open. That number is a problem for even the well-designed operating systems and routers out there.

More generally, Tor currently uses two levels of congestion avoidance – TCP flow control per-link, and a simple windowing scheme per-circuit. It has been suggested that this approach is causing performance problems, because the two schemes interact badly.

Experiments show that moving congestion management to be fully end-to-end offers a significant improvement in performance.

There have been two proposals to resolve this problem, but their underlying principle is the same: use an unreliable protocol for links between Tor relays, and perform error recovery and congestion management between the client and exit relay. Tor partially funded Joel Reardon's thesis [13] under Ian Goldberg. His thesis proposed using DTLS [14] (a UDP variant of TLS) as the link protocol and a cut-down version of TCP to give reliability and congestion avoidance, but largely using the existing Tor cell protocol. Csaba Kiraly *et al.* [3] proposed using IPSec [1] to replace the entire Tor cell and link protocol.

Each approach has its own strengths and weaknesses. DTLS is relatively immature, and Reardon noted deficiencies in the OpenSSL implementation of the protocol. However, the largest missing piece from this proposal is a high-quality, privacy preserving TCP stack, under a compatible license. Prior work has shown that there is a substantial privacy leak from TCP stack and clockskew fingerprinting [4, 8]. Therefore to adopt this proposal, Tor would need to incorporate a TCP stack, modified to operate in user-mode and to not leak identity information.

Reardon built a prototype around the TCP-Daytona stack [12], developed at IBM Labs, and based on the Linux kernel TCP stack. This implementation is not publicly available and its license is unclear, so it is unlikely to be suitable for use in Tor. Writing a TCP stack from scratch is a substantial undertaking, and therefore other attempts have been to move different operating system stacks into user-space. While there have been some prototypes, the maturity of these systems have yet to be shown.

Kiraly *et al.* rely on the operating system IPsec stack, and a modification to the IKE key exchange protocol to support onion routing. As with the proposal from Reardon, there is a risk of operating system and machine fingerprinting from exposing the client TCP stack to the exit relay. This could be resolved in a similar way, by implementing a user-mode IPsec stack, but this would be a substantial effort, and would lose some of the advantages of making use of existing building blocks.

Prof. Goldberg has a new student named Chris Alexander picking up where Joel left off. He's currently working on fixing bugs in OpenSSL's implementation of DTLS along with other core libraries that we'd need to use if we go this direction.

Impact: High.

Effort: High effort to get all the pieces in place.

Risk: High risk that it would need further work to get right.

Plan: We should keep working with them (and help fund Chris) to get this project closer to something we can deploy. The next step on our side is to deploy a separate testing Tor network that uses datagram protocols, based on patches from Joel and others, and get more intuition from that. We could optimistically have this testbed network deployed in late 2009.

1.2 We chose Tor's congestion control window sizes wrong

Tor maintains a per-circuit maximum of unacknowledged cells (`CIRCWINDOW`). If this value is exceeded, it is assumed that the circuit has become congested, and so the originator stops sending. Kiraly proposed [2, 3] that reducing this window size would substantially decrease latency (although not to the same extent as moving to an unreliable link protocol), while not affecting throughput.

Specifically, right now the circuit window size is 512KB and the per-stream window size is 256KB. These numbers mean that a user downloading a large file receives it (in the ideal case) in chunks of 256KB, sending back acknowledgements for each chunk. In practice, though, the network has too many of these chunks moving around at once, so they spend most of their time waiting in buffers at relays.

Reducing the size of these chunks has several effects. First, we reduce memory usage at the relays, because there are fewer chunks waiting and because they're smaller. Second, because there are fewer bytes vying to get onto the network at each hop, users should see lower latency.

More investigation is needed on precisely what should be the new value for the circuit window, and whether it should vary. Out of 100KB, 512KB (current value in Tor) and 2560KB, they found the optimum was 100KB for all levels of packet loss. However this was only evaluated for a fixed network latency and relay bandwidth, where all users had the same `CIRCWINDOW` value. Therefore, a different optimum may exist for networks with different characteristics, and during the transition of the network to the new value.

Impact: Medium. It seems pretty clear that in the steady-state this patch is a good idea; but it's still up in the air whether the transition period will show immediate improvement or if there will be a period where traffic from people who upgrade get clobbered by traffic from people who haven't upgraded yet.

Effort: Low effort to deploy – it's a several line patch!

Risk: Medium risk that we haven't thought things through well enough and we'd need to back it out or change parts of it.

Plan: Once we start on Tor 0.2.2.x (in the next few months), we should put the patch in and see how it fares. We should go for maximum effect, and choose the lowest possible window setting of 100 cells (50KB).

2 Some users add way too much load

Section 1 described mechanisms to let low-volume streams have a chance at competing with high-volume streams. Without those mechanisms, normal web browsing users will always get squeezed out by people pulling down larger content and tolerating high latency. But the next problem is that some users simply add more load than the network can handle. Just making sure that all the load gets handled fairly isn't enough if there's too much load in the first place.

When we originally designed Tor, we aimed for high throughput. We figured that providing high throughput would mean we get good latency properties for free. However, now that it's clear we have several user profiles trying to use the Tor network at once, we need to consider changing some of those design choices. Some of those changes would aim for better latency and worse throughput.

2.1 Squeeze over-active circuits

The Tor 0.2.0.30 release included this change:

```
- Change the way that Tor buffers data that it is waiting to write.
  Instead of queueing data cells in an enormous ring buffer for each
```

```
client->relay or relay->relay connection, we now queue cells on a
separate queue for each circuit. This lets us use less slack memory,
and will eventually let us be smarter about prioritizing different
kinds of traffic.
```

Currently when we're picking cells to write onto the network, we choose round-robin from each circuit that wants to write. We could instead remember which circuits have written many cells recently, and give priority to the ones that haven't.

Technically speaking, we're reinventing more of TCP here, and we'd be better served by a general switch to DTLS+UDP. But there are two reasons to still consider this separate approach.

The first is rapid deployment. We could get this change into the Tor 0.2.2.x development release in mid 2009, and as relays upgrade, the change would gradually phase in. This timeframe is way earlier than the practical timeframe for switching to DTLS+UDP.

The second reason is the flexibility this approach provides. We could give priorities based on recent activity ("if you've sent much more than the average in the past 10 seconds, then you get slowed down"), or we could base it on the total number of bytes sent on the circuit so far, or some combination. Even once we switch to DTLS+UDP, we may still want to be able to enforce some per-circuit quality-of-service properties.

This meddling is tricky though: we could encounter feedback effects if we don't perfectly anticipate the results of our changes. For example, we might end up squeezing certain classes of circuits too far, causing those clients to build too many new circuits in response. Or we might simply squeeze all circuits too much, ruining the network for everybody.

Also, Bittorrent is designed to resist attacks like this – it periodically drops its lowest-performing connection and replaces it with a new one. So we would want to make sure we're not going to accidentally increase the number of circuit creation requests and thus just shift the load problem.

Impact: High, if we get it right.

Effort: Medium effort to deploy – we need to go look at the code to figure out where to change, how to efficiently keep stats on which circuits are active, etc.

Risk: High risk that we'd get it wrong the first few times. Also, it will be hard to measure whether we've gotten it right or wrong.

Plan: Step one is to evaluate the complexity of changing the current code. We should do that for Tor 0.2.2.x in mid 2009. Then we should write some proposals for various meddling we could do, and try to find the right balance between simplicity (easy to code, easy to analyze) and projected effect.

2.2 Throttle certain protocols at exits

If we're right that Bittorrent traffic is a main reason for Tor's load, we could bundle a protocol analyzer with the exit relays. When they detect that a given outgoing stream is a protocol associated with bulk transfer, they could set a low rate limit on that stream. (Tor already supports per-stream rate limiting, though we've never found a need for it.)

This is a slippery slope in many respects though. First is the wiretapping question: is an application that automatically looks at traffic content wiretapping? It depends which lawyer you ask. Second is the network neutrality question: remember Comcast's famous "we're just delaying the traffic" quote. Third is the liability concern: once we add this feature in, what other requests are we going to get for throttling or blocking certain content? And does the capability to throttle certain content change the liability situation for the relay operator?

Impact: Medium-high.

Effort: Medium effort to deploy: need to find the right protocol recognition tools and sort out how to bundle them.

Risk: This isn't really an arms race we want to play. The "encrypted bittorrent" community already has a leg up since they've been fighting this battle with the telco's already. Plus the other downsides.

Plan: Not a good move.

2.3 Throttle certain protocols at the client side

While throttling certain protocols at the exit side introduces wiretapping and liability problems, detecting them at the client side is more straightforward. We could teach Tor clients to detect protocols as they come in on the socks port, and automatically treat them differently – and even pop up an explanation box if we like.

This approach opens a new can of worms though: clients could disable the "feature" and resume overloading the network.

Impact: Medium-high.

Effort: Medium effort to deploy: need to find the right protocol recognition tools and sort out how to bundle them.

Risk: This isn't really an arms race we want to play either. Users who want to file-share over Tor will find a way. Encouraging people to fork a new "fast" version of Tor is not a good way to keep all sides happy.

Plan: Not a good move.

2.4 Throttle all streams at the client side

While we shouldn't try to identify particular protocols as evil, we could set stricter rate limiting on client streams by default. If we set a low steady-state rate with a high bucket size (e.g. allow spikes up to 250KB but enforce a long-term rate for all streams of 5KB/s), we would probably provide similar performance to what clients get now, and it's possible we could alleviate quite a bit of the congestion and then get even better and more consistent performance.

Plus, we could make the defaults higher if you sign up as a relay and pass your reachability test.

The first problem is: how should we choose the numbers? So far we have avoided picking absolute speed numbers for this sort of situation, because we won't be able to predict a number now which will still be the correct number in the future.

The second problem is the same as in the previous subsection – users could modify their clients to disable these checks. So we would want to do this step only if we also put in throttling at the exits or intermediate relays, a la Section 2.1. And if that throttling works, changing clients (and hoping they don't revert the changes) may be unnecessary.

Impact: Low at first, but medium-high later.

Effort: Low effort to deploy.

Risk: If we pick high numbers, we'll never see much of an impact. If we pick low numbers, we could accidentally choke users too much.

Plan: It's not crazy, but may be redundant. We should consider in Tor 0.2.2.x whether to do it, in conjunction with throttling at other points in the circuit.

2.5 Default exit policy of 80,443

We hear periodically from relay operators who had problems with DMCA takedown attempts, switched to an exit policy of "permit only ports 80 and 443", and no longer hear DMCA complaints.

Does that mean that most file-sharing attempts go over some other port? If only a few exit relays permitted ports other than 80 and 443, we would effectively squeeze the high-volume flows onto those few exit relays, reducing the total amount of load on the network.

First, there's a clear downside: we lose out on other protocols. Part of the point of Tor is to be application-neutral. Also, it's not clear that it would work long-term, since corporate firewalls are continuing to push more and more of the Internet onto port 80.

To be clearer, we have more options here than the two extremes. We could switch the default exit policy from allow-all-but-these-20-ports to accept-only-these-20-ports. We could even get more complex, for example by applying per-stream rate limiting at the exit relays to streams destined for certain ports.

Impact: Low? Medium? High?

Effort: Low effort to deploy.

Risk: The Tor network becomes less useful, roughly in proportion to the amount of speedup we get.

Plan: I think we should take some of these steps in the Tor 0.2.2.x timeframe. The big challenge here is that we don't have much intuition about how effective the changes should be, so we don't know how far to go.

2.6 Better user education

We still run across users who think any anonymity system out there must have been designed with file-sharing in mind. If we make it clearer in the FAQ and our webpage that Tor isn't for high-volume streams, that might combine well with the other approaches above.

Overall, the challenge of users who want to overload the system will continue. Tor is not the only system that faces this challenge.

3 The Tor network doesn't have enough capacity

Section 1 aims to let web browsing connections work better in the face of high-volume streams, and Section 2 aims to reduce the overall load on the network. The third reason why Tor is slow is that we simply don't have enough capacity in the network to handle all the users who want to use Tor.

Why do we call this the third problem rather than the number one problem? Just adding more capacity to the network isn't going to solve the performance problem. If we add more capacity without solving the issues with high-volume streams, then those high-volume streams will expand to use up whatever new capacity we add.

Economics tells us to expect that improving performance in the Tor network (i.e. increasing supply) means that more users will arrive to fill the void. So in either case we shouldn't be under the illusion that Tor will magically just become faster once we implement these improvements. We place the first two sections higher in priority because their goals are to limit the ability of the high-volume users to become even higher-volume users, thus allowing the new capacity to be more useful to the other users. We discuss the supply-vs-demand question more in Section 7.1.

3.1 Tor server advocacy

Encouraging more volunteers to run Tor servers, and existing volunteers to keep their servers running, will increase network capacity and hence performance.

Impact: High, assuming we work on the plans from Section 1 and Section 2 also.

Effort: Medium to high, depending on how much we put in.

Risk: Low.

Plan: A clear win. We should do as many advocacy aspects as we can fit in.

3.1.1 Talks and trainings

One of the best ways we've found for getting new relays is to go to conferences and talk to people in person. There are many thousands of people out there with spare fast network connections and a willingness to help save the world. Our experience is that visiting them in person produces much better results, long-term, than Slashdot articles.

Roger and Jake have been working on this angle, and Jake will be ramping up even more on it in 2009.

Advocacy and education is especially important in the context of new and quickly-changing government policies. In particular, the data retention question in Germany is causing instability in the overall set of volunteers willing to run relays. Karsten's latest metrics[1] show that while the number of relays in other countries held steady or went up during 2008, the numbers in Germany went down over the course of 2008. On the other hand, the total amount of bandwidth provided by German relays held steady during 2008 – so while other operators picked up the slack, we still lost overall diversity of relays. These results tell us where to focus our efforts.

3.1.2 Better support for relay operators

Getting somebody to set up a relay is one thing; getting them to keep it up is another thing entirely. We lose relays when the operator reboots and forgets to set up the relay to start on boot. We lose relays when the operator looks through the website and doesn't find the answer to a question.

We've been working on a new service for relay operators called Tor Weather[2]. The idea is that once you've set up your relay, you can subscribe to get an email whenever it goes down. We need to work on the interface more, for example to let people subscribe to various levels of notification, but the basic idea seems like a very useful one.

With Tor Weather you can also subscribe to watch somebody *else*'s relay; so this service should tie in well for the people doing advocacy, to let them focus their follow-ups when a relay they helped set up disappears.

We are also considering setting up a mailing list exclusively for relay operators, to give them a better sense of community, to answer questions and concerns more quickly, etc.

We should also consider offering paid or subsidized support options so relay operators have a place to go for help. Corporations and universities running relays could get direct phone, email, or IM support options.

3.1.3 A Facebook app to show off your relay

We're currently developing a Facebook application that will allow relay operators to link their Tor relays to their Facebook profile. Volunteers who desire can therefore publicly get credit for their contribution to the Tor network. This would raise awareness for Tor, and encourage others to operate relays.

Opportunities for expansion include allowing relay operators to form "teams", and for these teams to be ranked on the contribution to the network. (Real world examples here include the SETI screensaver and the MD5 hash crack challenges.) This competition may give more encouragement for team members to increase their contribution to the network. Also, when one of the team members has their relay fail, other team members may notice and provide assistance on fixing the problem.

3.1.4 Look for new ways to get people to run relays

We are not primarily social engineers, and the people that we are good at convincing to set up relays are not a very huge group.

We need to keep an eye out for more creative ways to encourage a broader class of users to realize that helping out by operating a relay will ultimately be something they want to do.

3.2 Funding more relays directly

Another option is to directly pay hosting fees for fast relays (or to directly sponsor others to run them).

The main problem with this approach is that the efficiency is low: at even cheap hosting rates, the cost of a significant number of new relays grows quickly. For example, if we can find 100 non-exit relays providing

[1] https://www.torproject.org/projects/metrics
[2] https://weather.torproject.org/

1MB/s for as low as $100/mo (and at that price it'd be renting space on a shared server, with all the resource sharing hassles that comes with), that's $120k per year. Figure some more for maintenance and coordination, the overhead to find 100 locations that are on sufficiently different networks and administrative zones, etc.

The amount of work involved in running them as exit relays might be a few times this cost, due to higher hosting fees, more effort involved in establishing and maintaining the right relationships, having lawyers nearby, etc.

Plus the costs just keep coming, month after month.

Overall, it seems more sustainable to invest in better code, and community outreach and education.

Impact: Medium.

Effort: High.

Risk: Low.

Plan: If we end up with extra funding, sure. Otherwise, I think our time and effort are better spent on design and coding that will have long-term impact rather than be recurring costs.

3.3 Handling fast Tor relays on Windows

Advocating that users set up relays is all well and good, but if most users are on Windows, and Tor doesn't support fast relays on Windows well, then we're in a bad position.

Nick has been adapting libevent so it can handle a buffer-based abstraction rather than the traditional Unix-style socket-based abstraction. Then we will modify Tor to use this new abstraction. Nick's blog post[3] provides more detail.

Impact: Medium.

Effort: High, but we're already halfway through.

Risk: Low.

Plan: Keep at it. We're on schedule to get a test version (one that works for Nick) out in September 2009. Then iterate until it works for everybody.

3.4 Relay scanning to find overloaded relays or broken exits

Part of the reason that Tor is slow is because some of the relays are advertising more bandwidth than they can realistically handle. These anomalies might be due to bad load balancing on the part of the Tor designers, bad rate limiting or flaky network connectivity on the part of the relay operator, or malicious intent. Similarly, some exit relays might fail to give back the 'real' content, requiring users to repeat their connection attempts.

Mike has been working on tools to identify these relays: SpeedRacer[4] and SoaT[5]. Once the tools are further refined, we should be able to figure out if there are general classes of problems (load balancing, common usability problems, etc) that mean we should modify our design to compensate. The end goal is to get our tools to the point where they can automatically tell the directory authorities to leave out certain misbehaving relays in the network status consensus, and/or adjust the bandwidths they advertise for each relay.

Impact: Low.

Effort: Medium.

Risk: Low.

Plan: Keep at it. We're on schedule to get a test version (that works for Mike) out in mid 2009. Then iterate until it works for everybody.

[3] `https://blog.torproject.org/blog/some-notes-progress-iocp-and-libevent`

[4] `https://svn.torproject.org/svn/torflow/trunk/README.PerfMeasurements`

[5] `https://svn.torproject.org/svn/torflow/trunk/NetworkScanners/README.ExitScanning`

3.5 Getting dynamic-IP relays back into the relay list quickly

Currently there is a delay of 2-5 hours between when a relay changes its IP address and when that relay gets used again by clients. This delay causes two problems: relays on dynamic IP addresses will be underutilized (contributing less to the total network capacity than they could), and clients waste time connecting to relay IP addresses that are no longer listening.

There are several approaches that can mitigate this problem by notifying clients sooner about IP address changes. The first approach is to continue on our path of simplifying directory information (see Section 6.1): if we can put out "diffs" of the network status more often than once an hour, clients can get updated quicker. A second approach is for each relay to estimate how volatile its IP address is, and advertise this in its descriptor. Clients then ignore relays with volatile IP addresses and old descriptor. Similarly, directory authorities could prioritise the distribution of updated IP addresses for freshly changed relays.

As a last note here, we currently have some bugs that are causing relays with dynamic IP addresses to fall out of the network entirely. If a third to half of the relays are running on dynamic IP addresses, that's really bad.

Impact: Low-medium.

Effort: Low-medium.

Risk: Low.

Plan: Track down and fix bugs for Tor 0.2.2.x. Continue simplifying directory information so we can get new info to clients quicker.

3.6 Incentives to relay

Our blog post on this topic[6] explains our work to-date on this topic. The current situation is that we have two designs to consider: one that's quite simple but has a serious anonymity problem, and one that's quite complex.

I think we should move forward with the first (simple but flawed) design. There are several pieces to moving it forward. The first phase is changing Tor's queueing mechanisms to be able to give some circuits priority over others. This step also ties into the other development items in this document regarding cell-, circuit-, and connection-priorities. The second phase is then redesigning the "gold star" mechanism so the priority earned by relays lasts long enough that there's a sufficient anonymity set for them. We'll need to look at current and projected network metrics to discover a good upper bound on relay churn. The question to answer is: "What period of time, taken as a rolling snapshot of which relays are present in the network, guarantees a sufficiently large anonymity set for high-priority relays?" Hopefully the answer is something like 7 or 14 days. There are other missing pieces in there, like "what do we mean by sufficiently?", that we'll just have to guess about. The third phase is to actually sort out how to construct and distribute gold-star cryptographic certificates that entry relays can verify.

Notice that with the new certificates approach, we can reward users who contribute to the network in other ways than running a fast public relay – examples might include top sponsors, users who run stable bridge relays, translators, people who fix bugs, etc.

Impact: Medium-high.

Effort: Medium-high.

Risk: Medium-high: if we screw up the balance of our community-oriented infrastructure, we might end up hurting more than we help.

Plan: Accomplishing the three phases above will put us in a much better position to decide whether to deploy this idea. At the same time, the more complex options might become more within reach as other research teams investigate and refine them, so we should keep an eye on them too.

[6] `https://blog.torproject.org/blog/two-incentive-designs-tor`

3.7 Reachable clients become relays automatically

Even if we don't add in an incentive scheme, simply making suitable users into relays by default should do a lot for our capacity problems.

We've made many steps toward this goal already, with automated reachability testing, bandwidth estimation, UPnP support built in to Vidalia, and so on.

There are a few risks here though. First, relaying traffic could introduce anonymity vulnerabilities, and we need to learn more about that first. (That's on the roadmap for 2009.) Second, making clients into relays by default could make some users upset. Third, this approach could change how sysadmins view Tor. By putting ourselves into the same category as Skype, we would scale up the "blocking Tor connections" arms race by a level that's hard to predict. Also, we need to finish deployment of Section 3.3 before we can roll this out, or we'll just make a bunch of Windows machines crash.

We had originally been avoiding the "everybody a relay" design until we had a better plan for scaling the directory to be able to distribute tens of thousands of relay addresses. I think these two plans are not as related as we first thought, though. For example, a very simple answer for what to do if we get more relays than our current directory scheme can handle is to publish only the best relays, for some metric of best that considers capacity, expected uptime, etc. That should be a perfectly adequate stopgap measure. The step after that would be to consider splintering the network into two networkstatus documents, and clients flip a coin to decide which they use. Ultimately, if we are so lucky as to get into the position of having too many relays, we'll want to look at the distribution and properties of the relays we have when deciding what algorithms would best make use of them.

Impact: High.

Effort: Medium, now that we've done a lot of hard work already.

Risk: Medium.

Plan: Wrap up our investigations into the anonymity implications of being a relay, at the same time as working on a plan for exactly how the Tor client should decide if it's suitable for elevation to relay status. This is going to happen, it's just a matter of how soon.

4 Tor clients choose paths imperfectly

Even when we sort out the congestion control issues, the problem of users abusing the network with too much traffic, and the question of overall capacity, we still face a fourth problem. Users need to choose their paths in such a way that everybody is using the network efficiently.

Right now, Tor relays estimate their capacity by observing the largest traffic burst they've seen themselves do in the past day. They advertise that bandwidth capacity in the directory information, and clients weight their path selection by the bandwidth of each relay. For example, a relay that advertises 100KB/s peak bandwidth will be chosen twice as often as a relay that advertises 50KB/s peak bandwidth.

There are several problems with our current algorithm that are worth fixing.

4.1 We don't balance traffic over our bandwidth numbers correctly

Selecting relays with a probability proportional to their bandwidth contribution to the network may not be the optimal algorithm. Murdoch and Watson [10] investigated the performance impact of different relay selection algorithms, and came up with a model to describe the optimal path selection strategies based on how loaded the network is.

Tor's current selection strategy is optimal when the network is fully loaded. That is, if every single byte is going to be used, then weighting by capacity is the right way to do it. But if the network is not fully loaded, then the fast relays end up with less load than the slow relays. To compensate, clients should pick faster relays with higher probability.

In particular, we can estimate the network load because all Tor relays publish both their capacity and usage in their relay descriptor (but see Section 4.2 for problems that crop up there). The Tor network is currently loaded at around 50%. This level is much higher than most reasonable networks, indicating that our plan in Section 3 to get more overall capacity is a good one. But 50% is quite far from 100% when it becomes to optimal load balancing.

To find the optimum relay selection probabilities the model, Steven used a hill-climbing algorithm to minimize network latency, with a Tor directory snapshot as input. The results (shown in Figure 1 and Figure 2) depend on the network load relative to overall capacity. As load approaches capacity, the optimum selection probabilities converge to the one currently used by Tor: relay bandwidth proportional to network capacity. However, as load drops, the optimized selection algorithm favors slow relays less and faster relays more; many relays are not used at all.

Anecdotal evidence supports the theory that the fast relays in the Tor network have more spare capacity than they should. Several users have posted that they get much better Tor performance if they hard-code their paths to only use the fastest ten relays (and ignore the huge anonymity implications, of course).

The relay selection probabilities in these graphs are tuned to a particular level of network load. Figure 3 shows how average network latency is affected by relay selection probabilities, for different levels of network load. For all load levels examined, the optimized selection probabilities offer lower latency when compared to Tor's current selection algorithm. However, there's a downside to tailoring for a particular load level: if we see a much heavier load in practice than the one we had in mind when we tuned our selection biases, then we end up overbalancing the network in the other direction.

Specifically, each probability distribution has a cut-off point at which (according to the model) at least one relay will have a higher load than its capacity, at which its queue length, and hence latency, will become infinite. For the optimized selection probability distributions, this cut-off point is a few percent above the level they were designed to operate at. For Tor's current selection algorithm, it is when the overall network capacity equals the overall network load.

In this respect the Tor selection algorithm reaches the theoretical optimum, as no network can operate at greater than 100% utilization while maintaining finite latency. However, in a real instantiation of any of these alternative probability distributions, the network latency would not become infinite; instead a connection would time out and a different circuit would be selected. So in practice, if the wrong probability distribution was selected, the network would converge at a different one. Unfortunately the standard queuing theory models cannot handle this case; we need to move to a simulation rather than using equations and assumptions, to estimate the real effect.

Impact: Low-medium.

Effort: Medium, since we still need to get a better sense of the correct network load to expect, and we need to experiment to see if the model actually matches reality.

Risk: Low, since we can always back out the changes.

Plan: It seems clear that some adjustments should be done in terms of biasing selection toward the faster relays. The exact load level to anticipate remains an open question though. Fortunately, in our new networkstatus algorithm, the directory authorities declare the bandwidths for each relay. So we can just reweight them on the fly and clients will use the new numbers. That means once enough clients have upgraded to using the bandwidths specified in the networkstatus, we can start to experiment with shifting the biases and see what results we get.

4.2 The bandwidth estimates we have aren't very accurate

Weighting relay selection by bandwidth only works if we can accurately estimate the bandwidth for each relay.

Snader and Borisov [15] examined three strategies for estimating the bandwidth for each relay. The first strategy was Tor's current approach of looking for peaks in the actual bytes it's handled in the past day. The second strategy was active probing by the directory authorities. For their third strategy, they proposed that

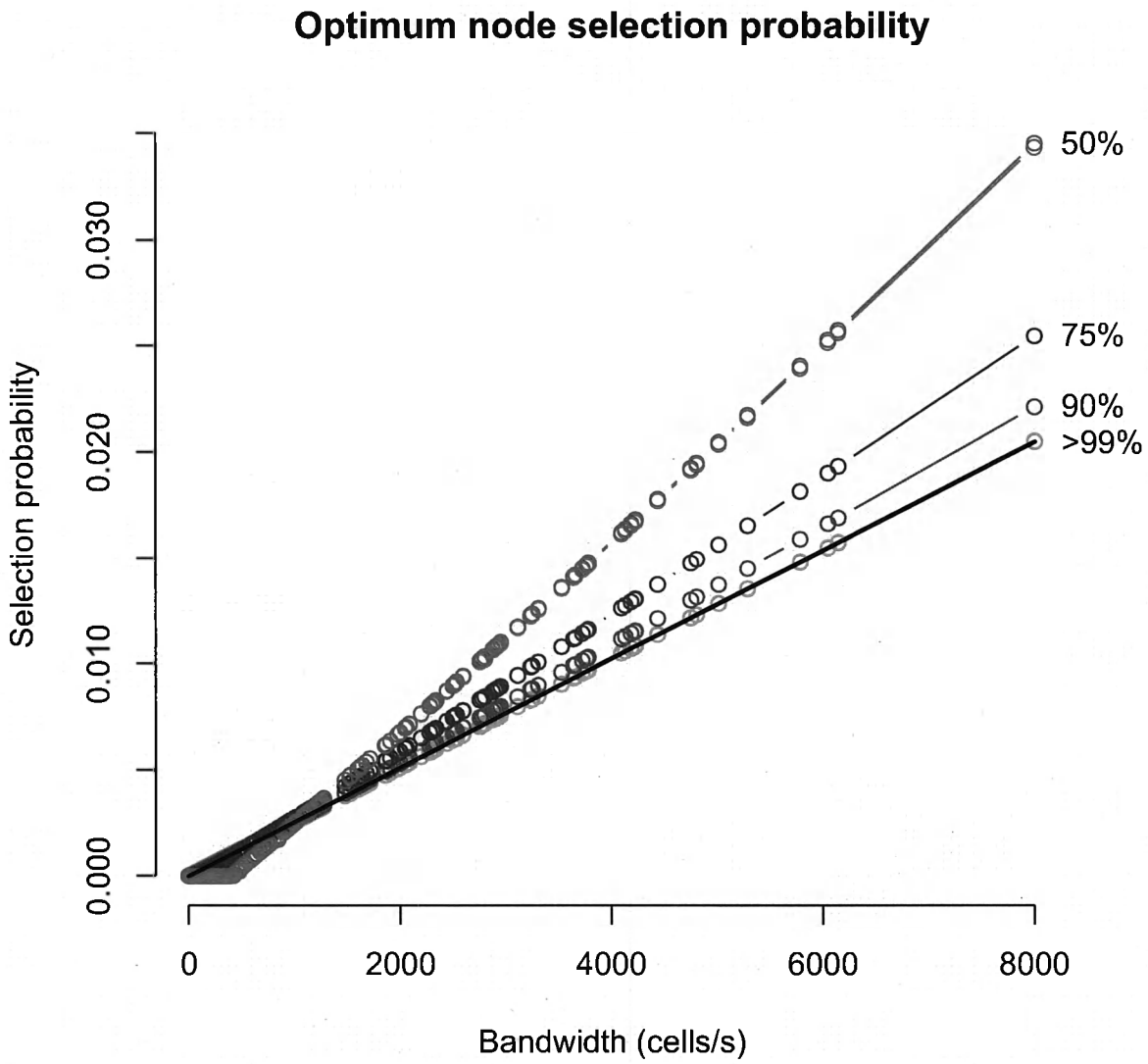


Figure 1: Optimum relay selection probabilities for a variety of network loads. Tor is currently at around 50% utilization. The relay selection probabilities currently used by Tor are shown in black.

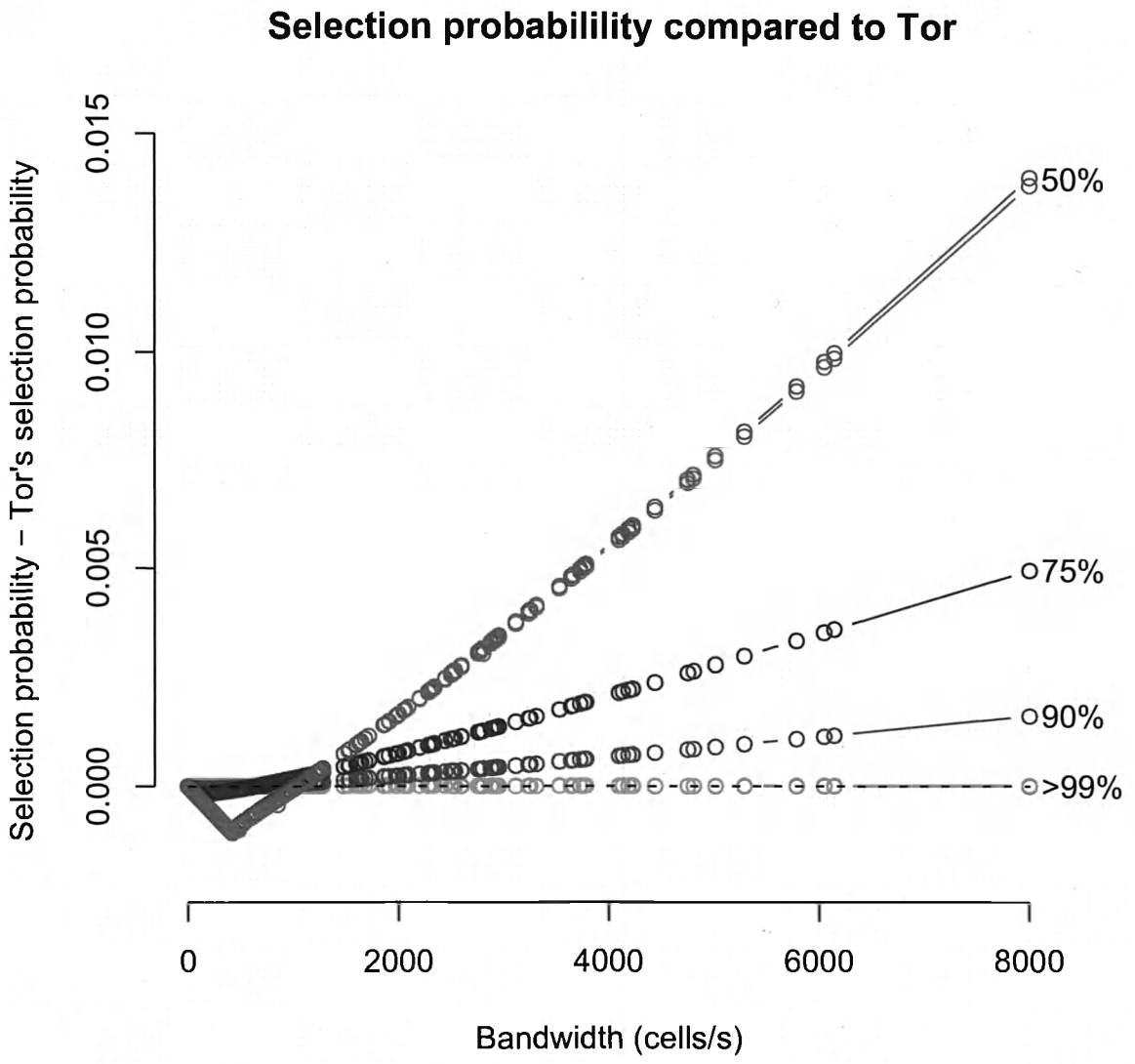


Figure 2: Difference between Tor's current relay selection probabilities and the optimum, for a variety of network loads. For Tor's current network load ($\approx 50\%$) shown in pink, the slowest relays are not used at all, and the slower relays are favoured less.

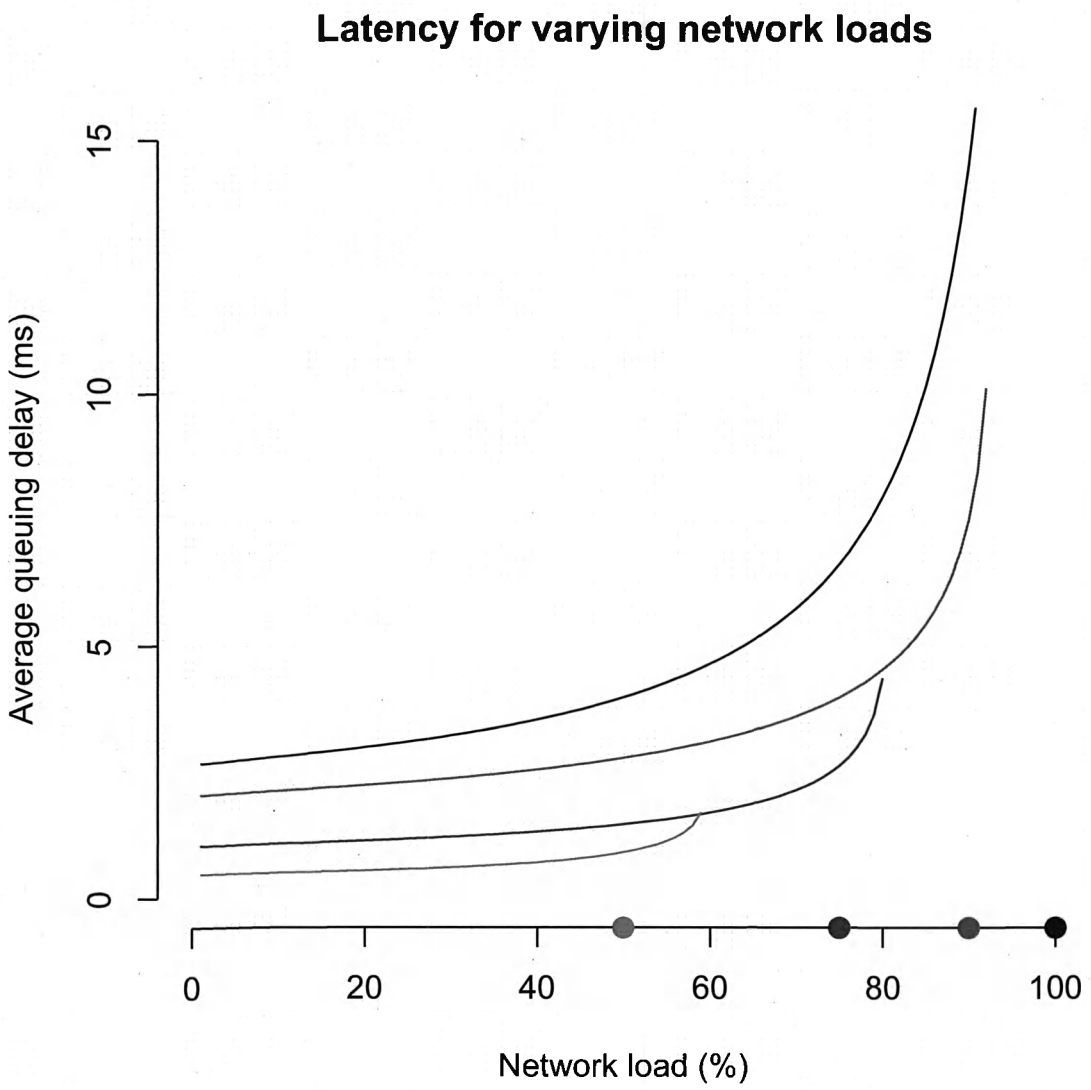


Figure 3: Average network latency against network load. Three relay selection probabilities are shown, optimized for 50%, 75%, and 90% network load. The Tor relay selection algorithm is also included (black). The dots on the x axis show the level of network load at which the relay selection probability distributions are optimized for. The line is cut off when the model predicts that at least one relay will have an infinite queue length, which occurs before load = capacity for all relay selection algorithms except for Tor's current one.

each Tor relay opportunistically monitor the data rates that it achieves when communicating with other Tor relays. Since currently Tor uses a clique topology, given enough time, all relays will communicate with all other Tor relays. If each Tor relay reports their measurements back to the directory authorities, then the median report should be a good estimate of that relay's bandwidth. As a bonus, this estimate should be difficult to game, when compared to the current approach of self-advertising bandwidth capacity.

Experiments show that opportunistic bandwidth measurement has a better systematic error than Tor's current self-advertised measure, although has a poorer log-log correlation (0.48 vs. 0.57). The most accurate scheme is active probing of capacity, with a log-log correlation of 0.63, but this introduces network overhead.

All three schemes suffer from fairly poor accuracy. Perhaps this inaccuracy is due to some relays with high variance in bandwidth capacity? We need to explore this area more to understand why our estimates are not as good as they could be.

Impact: Low-medium.

Effort: Medium, since we still need to get a better sense of the correct network load to expect, and we need to experiment to see if the model actually matches reality.

Risk: Low, since we can always back out the changes.

Plan: More research remains here to figure out what algorithms will actually produce more accurate bandwidth estimates. As with Section 4.1 above, once we do have some better numbers, we can change the weights in the directory, and clients will immediately move to the better numbers. We should also experiment with augmenting our estimates with active probes from Mike's SpeedRacer tool.

4.3 Bandwidth might not even be the right metric to weight by

The current Tor network selection algorithm biases purely by bandwidth. This approach will sometimes cause high latency circuits due to multiple ocean crossings or otherwise congested links. An alternative approach would be to not only bias selection of relays based on bandwidth, but to also bias the selection of hops based on expected latency.

Micah Sherr is finishing his PhD thesis at Penn under Matt Blaze, exploring exactly this issue. In the past we've avoided any sort of path selection algorithm that requires pairwise measurements of the network, because communicating N^2 measurements to clients would take too much bandwidth. Micah solves this problem by using a *virtual coordinate system* – a three or four dimension space such that distance between relays in the virtual coordinate space corresponds to the network latency (or other metric) between them.

His experiments show that we could see a significant speedup in the Tor network if users choose their paths based on this new relay selection algorithm. More research remains, of course, but the initial results are very promising.

On the other hand, reducing the number of potential paths would also have anonymity consequences, and these would need to be carefully considered. For example, an attacker who wishes to monitor traffic could create several relays, on distinct /16 subnets, but with low latency between them. A Tor client trying to minimize latency would be more likely to select these relays for both entry than exit than it would otherwise. This particular problem could be mitigated by selecting entry and exit relay as normal, and only using latency measurements to select the middle relay.

Impact: Medium-high.

Effort: Medium-high, since we first need to sort out how effective the algorithm is, and then we need to figure out a migration plan.

Risk: Medium, since a new selection algorithm probably carries with it a new set of anonymity-breaking papers that will only come out a few years after we deploy.

Plan: Micah is going to write a design proposal for getting relays to compute and maintain their virtual coordinates based on latency. Once we deploy that, we'll have some actual data points, and we'll be in a better position to simulate whether the idea will help in reality. Counting deployment time, that means we probably won't have clients using this scheme until 2010.

rotating to a new guard after a week or two is enough to substantially resolve the problem. We also need to consider the added risk that higher guard churn poses versus the original attack they were designed to thwart [11], but I think a few weeks should still be plenty high.

At the same time, there are fewer relays with the Guard flag than there should be. While the Exit flag really is a function of the relay's exit policy, the required properties for entry guards are much more vague: we want them to be "fast enough", and we want them to be "likely to be around for a while more". I think the requirements currently are too strict. This scarcity of entry guards in turn influences the anonymity the Tor network can provide, since there are fewer potential entry points into the network.

Impact: High.

Effort: Low.

Risk: Low.

Plan: We should do it, early in Tor 0.2.2.x. We'll need proposals first, both for the "dropping old guards" plan (to assess the tradeoff from the anonymity risk) and for the "opening up the guard criteria" plan.

5 Clients need to handle variable latency and failures better

The next issue we need to tackle is that Tor clients aren't as good as they should be at handling high or variable latency and connection failures. First, we need ways to smooth out the latency that clients see. Then, for the cases where we can't smooth it out enough, we need better heuristics for clients to automatically shift away from bad circuits, and other tricks for them to dynamically adapt their behavior.

5.1 Our round-robin and rate limiting is too granular

Tor's rate limiting uses a token bucket approach to enforce a long-term average rate of incoming and outgoing bytes, while still permitting short-term bursts above the allowed bandwidth. Each token represents permission to send another byte onto the network (or read from the network). Every second new tokens are added, up to some cap (the *bucket size*).

So Tor relays that have cells buffered waiting to go out onto the network will wait until the new second arrives, and then deliver as many cells as they can. In practice, this behavior results in traffic 'bumps' at the beginning of each second, with little network traffic the rest of the time. Mike and Karsten have been collecting data from circuit extension times (how long it takes to establish each hop of a circuit); the bumps are easily seen in Figure 5.

Our original theory when designing Tor's rate limiting was that one-second granularity should be sufficient: cells will go out as quickly as possible while the bucket still has tokens, and once it's empty there's nothing we can do but wait until the next second for permission to send more cells.

We should explore refilling the buckets more often than once a second, for three reasons. First, we'll get a better intuition about how full the buffers really are: if we spread things out better, then we could reduce latency by perhaps multiple seconds. Second, spikes-and-silence is not friendly for TCP, so averaging the flows ourselves might mean much smoother network performance. Third, sub-second precision will give us more flexibility in our priority strategies from Section 2.1.

On the other hand, we don't want to go too far: cells are 512 bytes, so it isn't useful to think in units smaller than that. Also, every network write operation carries with it overhead from the TLS record, the TCP header, and the IP packet header. Finally, network transmission unit (MTU) sizes vary, but if we could use a larger packet on the wire and we don't, then we're not being as efficient as we could be.

Impact: Low-Medium.

Effort: Medium.

Risk: Low, unless we add in bad feedback effects and don't notice.

Plan: We should continue collecting metrics to get better intuition here. While we're designing priority stategies for Section 2.1, we should keep the option of higher-resolution rate-limiting in mind.

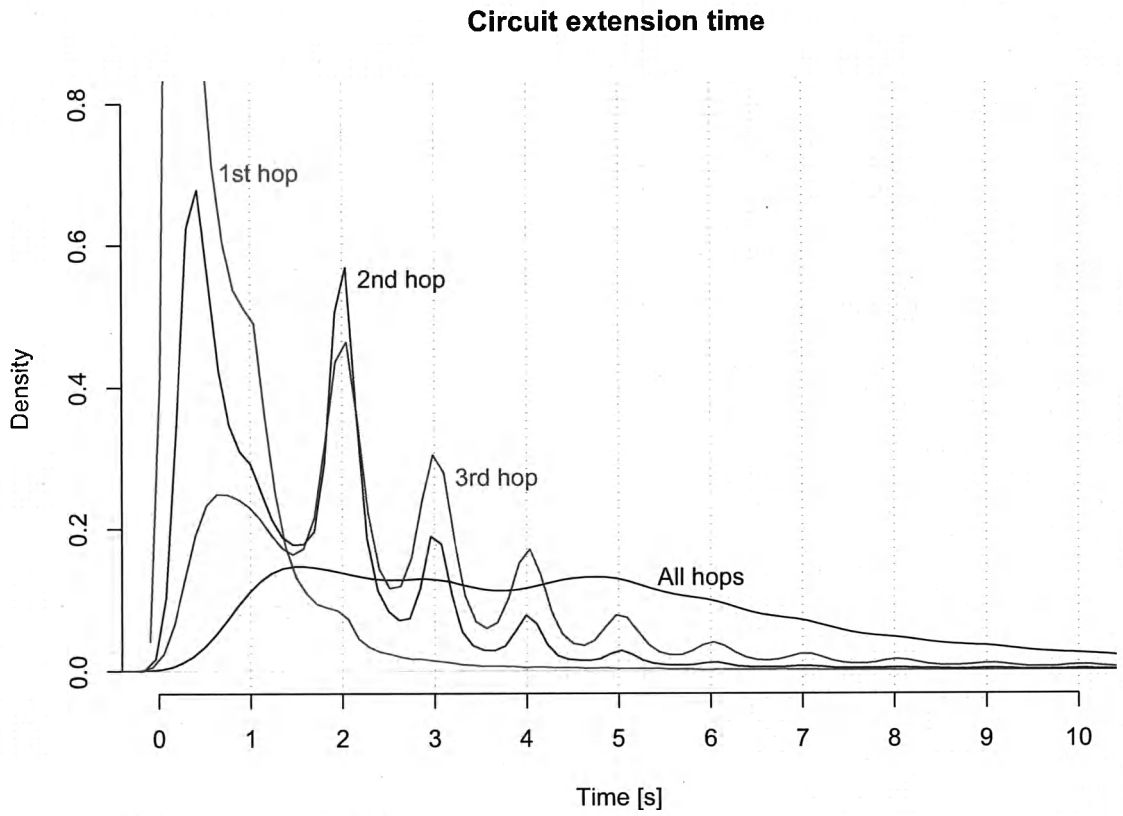


Figure 5: Number of seconds it takes to establish each hop of a 3-hop circuit. The higher density of samples around 2s, 3s, etc indicate that rate limiting at each relay is introducing extra delay into the responses.

5.2 Better timeouts for giving up on circuits and trying a new one

Some circuits are established very quickly, and some circuits take many seconds to form. The time it takes for the circuit to open can give us a hint about how well that circuit will perform for future traffic. We should discard extremely slow circuits early, so clients never have to even try them.

The question, though, is how to decide the right timeouts? If we set a static timeout in the clients, then choosing a number that's too low will cause clients to discard too many circuits. Worse, clients on really bad connections will never manage to establish a circuit. On the other hand, setting a number that's too high won't change the status quo much.

Fallon Chen worked during her Google-Summer-of-Code-2008 internship with us on collecting data about how long it takes for clients to establish circuits, and analyzing the data to decide what shape the distribution has (it appears to be a Pareto distribution). The goal is for clients to track their own circuit build times, and then be able to recognize if a circuit has taken longer than it should have compared to the previous circuits. That way clients with fast connections can discard not-quite-fast-enough circuits, whereas clients with slow connections can discard only the really-very-slow circuits. Not only do clients get better performance, but we can also dynamically adapt our paths away from overloaded relays.

Mike and Fallon wrote a proposal[8] explaining the details of how to collect the stats, how many data points the client needs before it has a good sense of the expected build times, and so on.

Further, there's another case in Tor where adaptive timeouts would be smart: how long we wait in between trying to attach a stream to a given circuit and deciding that we should try a new circuit. Right now we have a crude and static "try 10 seconds on one, then try 15 seconds on another" algorithm, which is both way too high and way too low, depending on the context.

Impact: Medium.

Effort: Medium, but we're already part-way through it.

Risk: Low, unless we've mis-characterized the distribution of circuit extend times, in which case clients end up discarding too many circuits.

Plan: We should deploy the changes in clients in Tor 0.2.2.x to collect circuit times, and see how that goes. Then we should gather data about stream timeouts to build a plan for how to resolve the second piece.

5.3 If extending a circuit fails, try extending a few other places before abandoning the circuit.

Right now, when any extend operation fails, we abandon the entire circuit. As the reasoning goes, any other approach allows an attacker who controls some relays (or part of the network) to dictate our circuits (by declining to extend except to relays that he can successfully attack).

However, this reasoning is probably too paranoid. If we try at most three times for each hop, we greatly increase the odds that we can reuse the work we've already done, but we don't much increase the odds that an attacker will control the entire circuit.

Overall, this modification should cut down on the total number of extend attempts in the network. This result is particularly helpful since some of our other schemes in this document involve increasing that number.

Impact: Low.

Effort: Low.

Risk: Low-medium. We need to actually do some computations to confirm that the risk of whole-path compromise is as low as we think it is.

Plan: Do the computations, then write a proposal, then do it.

[8] `https://svn.torproject.org/svn/tor/trunk/doc/spec/proposals/151-path-selection-improvements.txt`

5.4 Bundle the first data cell with the begin cell

In Tor's current design, clients send a "relay begin" cell to specify the intended destination for our stream, and then wait for a "relay connected" cell to confirm the connection is established. Only then do they complete the SOCKS handshake with the local application, and start reading application traffic.

We could modify our local proxy protocol in the case of Privoxy or Polipo so it sends the web request to the SOCKS port during the handshake. Then we could optimistically include the first cell worth of application data in the original begin cell. This trick would allow us to cut out an entire network round-trip every time we establish a new connection through Tor. The result would be quicker page loads for users.

Alas, this trick would involve extending the SOCKS protocol, which isn't usually a polite strategy when it comes to interoperating with other applications. On the other hand, it should be possible to extend it in a backwards-compatible way: applications that don't know about the trick would still behave the same and still work fine (albeit in a degraded mode where they waste a network round-trip).

Impact: Medium.

Effort: Medium.

Risk: Low.

Plan: Overall, it seems like a delicate move, but with potentially quite a good payoff. I'm not convinced yet either way.

6 The network overhead may still be high for modem users

Even if we resolve all the other pieces of the performance question, there still remain some challenges posed uniquely by users with extremely low bandwidth – for example, users on modems or cell phones. We need to optimize the Tor protocols so they are efficient enough that Tor can be practical in this situation too.

6.1 We've made progress already at directory overhead

We've already made great progress at reducing directory overhead, both for bootstrapping and maintenance. Our blog post on the topic provides background and details[9].

Proposal 158 further reduces the directory overhead, and is scheduled to be deployed in Tor 0.2.2.x.[10]

Impact: Low for normal users, high for low-bandwidth users.

Effort: Medium, but we're already a lot of the way through it.

Risk: Low.

Plan: We should roll out proposal 158. Then we'll be in good shape for a while. The next directory overhead challenge will be in advertising many more relays; but first we need to get the relays.

6.2 Our TLS overhead can also be improved

OpenSSL will, by default, insert an empty TLS application record before any one which contains data. This is to prevent an attack, by which someone who has partial control over the plaintext of a TLS stream, can also confirm guesses as to the plaintext which he does not control. By including an empty application record, which incorporates a MAC, the attacker is made unable to control the CBC initialization vector, and hence does not have control of the input to the encryption function [7].

This application record does introduce an appreciable overhead. Most Tor cells are sent in application records of their own, giving application records of 512 bytes (cell) + 20 bytes (MAC) + 12 bytes (TLS padding) + 5 bytes (TLS application record header) = 549 bytes. The empty application records contain

[9] https://blog.torproject.org/blog/overhead-directory-info%3A-past%2C-present%2C-future

[10] https://svn.torproject.org/svn/tor/trunk/doc/spec/proposals/158-microdescriptors.txt

only 20 bytes (MAC) + 12 bytes (TLS padding) + 5 bytes (TLS application record header) = 37 bytes. There is also a 20 byte IP header and 32 byte TCP header.

Thus the overhead saved by removing the empty TLS application record itself is $37/(549+37+20+32) = 5.8\%$. This calculation is assuming that the same number of IP packets will be sent, because currently Tor sends packets, with only one cell, far smaller than the path MTU. If Tor were to pack cells optimally efficiently into packets, then removing the empty application records would also reduce the number of packets, and hence TCP/IP headers, that needed to be sent. The reduction in TCP/IP header overhead would be $37/(549 + 37) = 6.3\%$.

Of course, the empty application record was inserted for a reason – to prevent an attack on the CBC mode of operation used by TLS, so before removing it we must be confident the attack does not apply to Tor. Ben Laurie (one of the OpenSSL developers) concluded that in his opinion Tor could safely remove the insertion of empty TLS application records [5]. Steven was able to come up with only certificational weaknesses (discussed in the above analysis), which are expensive to exploit and give little information to the attacker.

Impact: Low.

Effort: Low.

Risk: Medium, since our initial analysis might be wrong.

Plan: Do it in the Tor 0.2.2.x or 0.2.3.x timeframe. Not critical.

7 Last thoughts

7.1 Lessons from economics

Imagine we implement all the solutions above, and it doubles the effective capacity of the Tor network. The naïve hypothesis is that users would then experience twice the throughput. Unfortunately this is not true, because it assumes that the number of users does not vary with bandwidth available. In fact, as the supply of the Tor network's bandwidth increases, there will be a corresponding increase in the demand for bandwidth from Tor users. Simple economics shows that performance of Tor and other anonymization networks is controlled by how the number of users scales with available bandwidth; this relationship can be represented by a demand curve.[11]

Figure 6 is the typical supply and demand graph from economics textbooks, except with long-term throughput per user substituted for price, and number of users substituted for quantity of goods sold. As the number of users increases, the bandwidth supplied by the network falls.

In drawing the supply curve, we have assumed the network's bandwidth is constant and shared equally over as many users as needed. The shape of the demand curve is much harder to even approximate, but for the sake of discussion, we have drawn three alternatives. The number of Tor users and the throughput they each get is the intersection between the supply and demand curves – the equilibrium. If the number of users is below this point, more users will join and the throughput per user will fall to the lowest tolerable level. Similarly, if the number of users is too high, some will be getting lower throughput than their minimum, so will give up, improving the network for the rest of the users.

Now assume Tor's bandwidth grows by 50% – the supply curve shifts, as shown in the figure. By comparing how the equilibrium moves, we can see how the shape of the demand curve affects the performance improvement that Tor users see. If the number of users is independent of performance, shown in curve A, then everyone gets a 50% improvement, which matches the naïve hypothesis. More realistically, the number of users increases, so the performance gain is less. The shallower the curve gets, the smaller the performance increase will be. For demand curve B, there is a 18% increase in the number of Tor users and a 27% increase

[11]The economics discussion is based on a blog post published in Light Blue Touchpaper [9]. The property discussed was also observed by Andreas Pfitzmann in response to a presentation at the PET Symposium [16].

in throughput. On the other hand, with curve C there are 33% more users and so only a 13% increase in throughput for each user.

The above analysis glosses over many topics. One interesting analysis is reaching equilibrium – in fact it could take some time between the network bandwidth changing and the user population reaching stability. If this period is sufficiently long and network bandwidth is sufficiently volatile it might never reach equilibrium. We might also consider effects which shift the demand curve. In normal economics, marketing makes people buy a product even though they considered it too expensive. Similarly, a Slashdot article or news of a privacy scandal could make Tor users more tolerant of the poor performance. Finally, the user perception of performance is an interesting and complex topic. In this analyis we assumed that performance is equivalent to throughput; but actually latency, packet loss, predictability, and their interaction with TCP/IP congestion control are important components too.

So what does all this tell us?

The above discussion has argued that the speed of an anonymity network will converge on the slowest level that the most tolerant users will consider usable. This is problematic because there is significant variation in levels of tolerance between different users and different protocols. Most notably, file sharing users are subject to high profile legal threats, and do not require interactive traffic, so they will continue to use a network even if the performance is considerably lower than the usable level for web browsing.

In conventional markets, this type of problem is solved by differential pricing, for example different classes of seat on airline flights. In this model, several equilibrium points are allowed to form, and the one chosen will depend on the cost/benefit tradeoffs of the customers. A similar strategy could be used for Tor, allowing interactive web browsing users to get higher performance, while forcing bulk data transfer users to have lower performance (but still tolerable for them). Alternatively, the network could be configured to share resources in a manner such that the utility to each user is more equal. In this case, it will be acceptable to all users that a single equilibrium point is formed, because its level will no longer be characterized in terms of simple bandwidth.

Section 2 is an example of the former strategy. Web browsing users will be offered better performance, so we should attract more of them, but hopefully not so many that the performance returns to current levels. In constrast, bulk-traffic users will be given poorer performance, but since they are less sensitive to latency, it could be that they do not mind. Section 1 could be used to implement the latter strategy. If web-browsing users are more sensitive to latency than bandwidth, then we could optimize the network for latency rather than throughput.

7.2 The plan moving forward

Our next steps should be to work with funders and developers to turn this set of explanations and potential fixes into a roadmap: we need to lay out all the solutions, sort out the dependencies, assign developers to tasks, and get everything started.

At the same time, we need to continue to work on ways to measure changes in the network: without 'before' and 'after' snapshots, we'll have a much tougher time telling whether a given idea is actually working. Many of the plans here have a delay between when we roll out the change and when the clients and relays have upgraded enough for the change to be noticeable. Since our timeframe requires rolling out several solutions at the same time, an increased focus on metrics and measurements will be critical to keeping everything straight.

Lastly, we need to be aware that ramping up development on performance may need to push out or downgrade other items on our roadmap. So far, Tor has been focusing our development energy on the problems that funders are experiencing most severely at the time. This approach is good to make sure that we're always working on something that's actually important. But it also means that next year's critical items don't get as much attention as they should, and last year's critical items don't get as much maintenance as they should. Ultimately we need to work toward having consistent funding for core Tor development and maintenance as well as feature-oriented funding.

References

[1] KENT, S., AND SEO, K. Security architecture for the internet protocol. RFC 4301, IETF, December 2005.

[2] KIRALY, C. Effect of Tor window size on performance. Email to [redacted] February 2009. http://archives.seul.org/or/dev/Feb-2009/msg00000.html.

[3] KIRALY, C., G., B., AND LO CIGNO, R. Solving performance issues in anonymization overlays with a L3 approach. Tech. Rep. DISI-08-041, University of Trento, September 2008. version 1.1, http://disi.unitn.it/locigno/preprints/TR-DISI-08-041.pdf.

[4] KOHNO, T., BROIDO, A., AND CLAFFY, K. Remote physical device fingerprinting. In *IEEE Symposium on Security and Privacy* (Oakland, CA, US, May 2005), IEEE Computer Society, pp. 211–225.

[5] LAURIE, B. On TLS empty record insertion. Email to [redacted] in thread "Re: Empty TLS application records being injected in Tor streams", December 2008. http://archives.seul.org/or/dev/Dec-2008/msg00005.html.

[6] MCCOY, D., BAUER, K., GRUNWALD, D., KOHNO, T., AND SICKER, D. Shining light in dark places: Understanding the Tor network. In *Proceedings of the Eighth International Symposium on Privacy Enhancing Technologies (PETS 2008)* (Leuven, Belgium, July 2008), N. Borisov and I. Goldberg, Eds., Springer, pp. 63–76.

[7] MÖLLER, B. Security of CBC ciphersuites in SSL/TLS: Problems and countermeasures, May 2004. http://www.openssl.org/~bodo/tls-cbc.txt.

[8] MURDOCH, S. J. Hot or not: Revealing hidden services by their clock skew. In *CCS '06: Proceedings of the 9th ACM Conference on Computer and Communications Security* (Alexandria, VA, US, October 2006), ACM Press, pp. 27–36.

[9] MURDOCH, S. J. Economics of Tor performance. Light Blue Touchpaper, 18 July 2007. http://www.lightbluetouchpaper.org/2007/07/18/economics-of-tor-performance/.

[10] MURDOCH, S. J., AND WATSON, R. N. M. Metrics for security and performance in low-latency anonymity networks. In *Proceedings of the Eighth International Symposium on Privacy Enhancing Technologies (PETS 2008)* (Leuven, Belgium, July 2008), N. Borisov and I. Goldberg, Eds., Springer, pp. 115–132.

[11] ØVERLIER, L., AND SYVERSON, P. Locating hidden servers. In *Proceedings of the 2006 IEEE Symposium on Security and Privacy* (May 2006), IEEE CS.

[12] PRADHAN, P., KANDULA, S., XU, W., SHAIKH, A., AND NAHUM, E. Daytona: A user-level TCP stack, 2002. http://nms.lcs.mit.edu/~kandula/data/daytona.pdf.

[13] REARDON, J. Improving Tor using a TCP-over-DTLS tunnel. Master's thesis, University of Waterloo, September 2008. http://hdl.handle.net/10012/4011.

[14] RESCORLA, E., AND MODADUGU, N. Datagram transport layer security. RFC 4347, IETF, April 2006.

[15] SNADER, R., AND BORISOV, N. A tune-up for Tor: Improving security and performance in the Tor network. In *Network & Distributed System Security Symposium* (February 2008), Internet Society.

Tor Blocking-Resistance Roadmap

Roger Dingledine Nick Mathewson Shava Nerad

March 14, 2007

1 Introduction

This document lays out the tasks we need to complete to finish designing, specifying, and deploying our blocking-resistance design, as well as related tasks that will improve Tor's efficiency and performance for the target users.

The general strategy is to finish the prototyping tasks early, so we can have a deployed working system even if it's rough around the edges. Then we can focus on refining it, improving Tor, and research and design for the required next steps.

In particular, we break the project tasks down into three categories: tasks for blocking-resistance; improved Tor efficiency and scalability; and improved user tools and documentation. The below milestones represent what we hope to achieve given enough developer resources. Then the remaining sections provide more details, as well as listing other tasks we ought to tackle.

Phase one: First deployed usable-by-engineers system. Aiming for May.

Phase two: Vidalia-side interfaces, deploy strategy pools, have started research and design on all the harder parts. Aiming for Sep.

Phase three: First-draft designs for improved descriptor fetching design and safer network partitioning, scanning-resistance, incentives, etc. Aiming for Nov.

2 Blocking resistance

2.1 Design for blocking resistance

We have written a first draft design document explaining our general approach to blocking resistance. We should workshop it with other experts in the field to get their ideas about how we can improve Tor's efficacy as an anti-censorship tool.

(Implementation draft, phase one.)

2.2 Implementation: bridge-side

Our anticensorship design calls for some nodes to act as "bridges" that are outside a national firewall, and others inside the firewall to act as pure clients. This part of the design is quite clear-cut; we're probably ready to begin implementing it. To **implement bridges**, we need to have servers publish themselves as limited-availability relays to a special bridge authority if they judge they'd make good servers. We will also need to help provide documentation for port forwarding, and an easy configuration tool for running as a bridge.

(Tor-side changes, phase one.)

(Vidalia-side changes, phase two.)

2.3 Implementation: bridge authority

The design here is also reasonably clear-cut: we need to run some directory authorities with a slightly modified protocol that doesn't leak the entire list of bridges. Thus users can learn up-to-date information for bridges they already know about, but they can't learn about arbitrary new bridges.

(Tor-side changes, phase one.)

2.4 Implementation: user-side

To **implement clients**, we need to provide a flexible interface to learn about bridges and to act on knowledge of bridges. We also need to teach them how to know to use bridges as their first hop, and how to fetch directory information from both classes of directory authority.

(Tor-side changes, phase one.)

(Vidalia-side changes, phase two.)

Clients also need to **use the encrypted directory variant** added in Tor 0.1.2.3-alpha. This will let them retrieve directory information over Tor once they've got their initial bridges. (We may want to get the rest of the Tor user base to begin using this encrypted directory variant too, to provide cover, but then we lose our statistics gathering abilities; perhaps wait until GeoIP solution in Section 2.7 is ready?)

(Tor-side changes, phase one.)

2.5 Normalizing the Tor protocol on the wire

Additionally, we should **resist content-based filters**. Though an adversary can't see what users are saying, some aspects of our protocol are easy to fingerprint *as* Tor. We should correct this where possible. Look like Firefox; or look like nothing? This task requires an overhaul of the Tor cell protocol to include versions, new cell types, etc.

(First round of fixes, phase one.)

(Research and second round of fixes, phase two/three?)

Future research: investigate timing similarities with other protocols. **(Expensive; fortunately optional.)**

2.6 Research/Design/Impl: how users discover bridges

Our design anticipates an arms race between discovery methods and censors.

Part one: personal bridges. Included in Section 2.4 above. **(Phase one.)**

Part two: families of personal bridges. **(Phase two.)**

Part three: public bridge pools with different strategies. We need to begin the infrastructure on our side quickly, preferably in a flexible language like Python, so we can adapt quickly to censorship. **(Implement pools one-five, phase two.)**

Part four: social network reputation system. **(Further design and research, phase three.)**

Ongoing milestones for social network reputation system. **(...)**

Part five: research and prepare new strategies. **(...)**

2.7 Research: User statistics; and how to know if a bridge has been blocked?

GeoIP maintenance, and "private" user statistics. How to know if the whole idea is working?

(Design, deploy, and gather stats, phase two.)

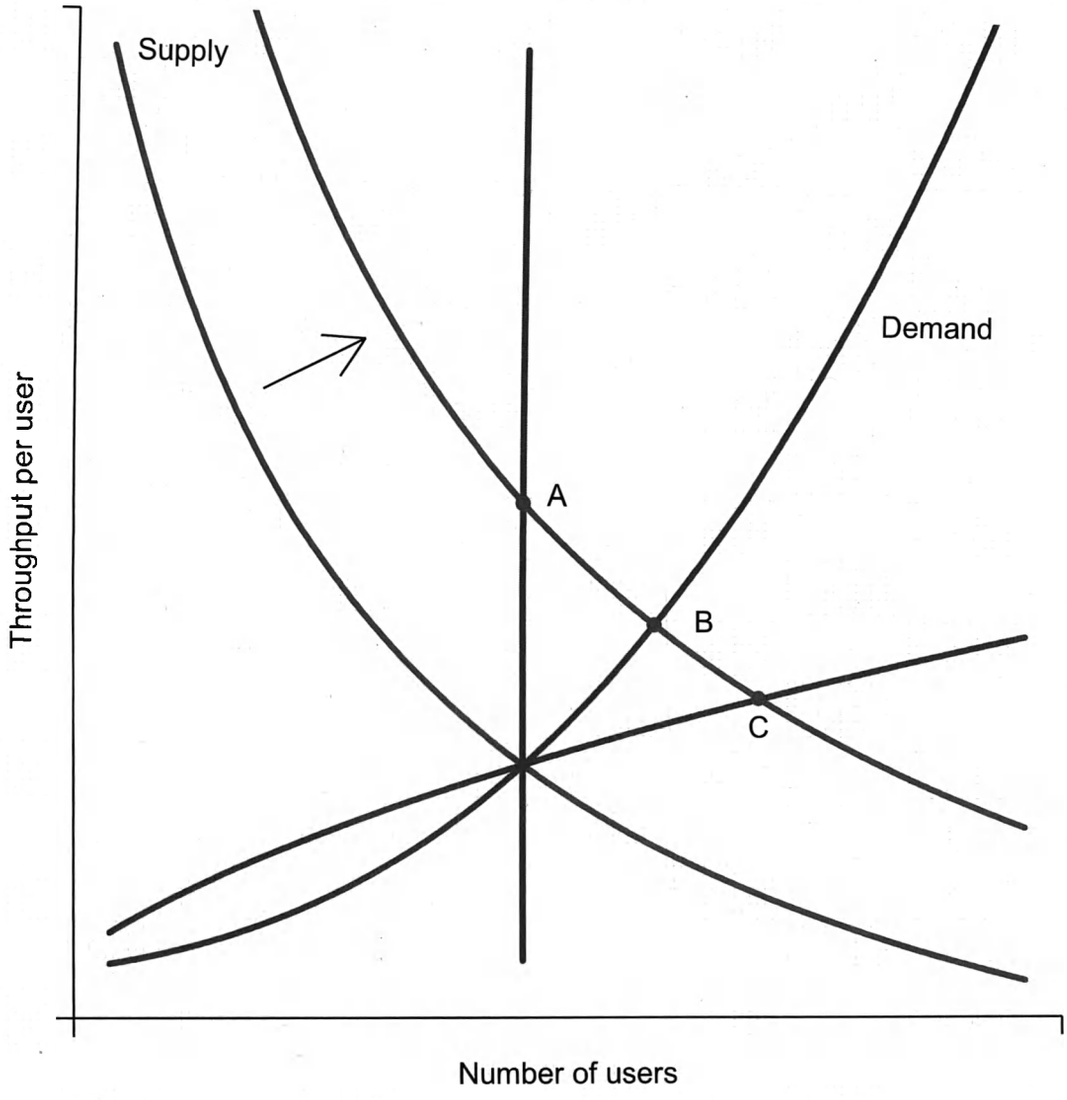


Figure 6: Hypothetical supply and demand curves for Tor network resources. As supply goes up, point A corresponds to no increase in users, whereas points B and C represent more users arriving to use up some of the new capacity.

2.8 Research: anonymity implications from becoming a bridge

Need to do at least a brief investigation, to help us design bridges correctly from an anonymity perspective and also to see if we've missed critical issues. For example, we need to decide if users should use the bridge's entry guards or not. (Could be a whole thesis.)

(Brief investigation, optional, phase one.)

2.9 Research: hiding whether the user is reading or publishing?

People publishing large documents will look different from people reading websites, even if they're all using Tor. If we introduce periodic convincing-looking bursts of uploaded bytes, perhaps we can obscure this correlation a bit. It won't be statistically perfect, but it will at least start this particular arms race.

(Initial research and simple fixes, optional, phase two/three?)

2.10 Research: how many bridges do you need to know to maintain reachability?

Early answer is to just guess and hope we're right.

Better answer is an actual study and analysis of bridge stability in practice, and go from there.

(Better analysis leads to more stable connections. When to do?)

2.11 Research: how to decentralize the bridge authorities for greater robustness?

Need to do it in a way that doesn't increase the adversary's ability to learn bridges linearly as we grow the number of bridge authorities.

(Initial research and design, phase two/three?)

2.12 Resisting censorship of the Tor website, docs, and mirrors

We should take some effort to consider **initial distribution of Tor and related information** in countries where the Tor website and mirrors are censored. (Right now, most countries that block access to Tor block only the main website and leave mirrors and the network itself untouched.) Falling back on word-of-mouth is always a good last resort, but we should also take steps to make sure it's relatively easy for users to get a copy.

(Simple fixes, phase two?)

We will probably need more complex steps down the road. **(...)**

2.13 Measuring quality of each bridge

With the success of our network, we've attracted servers in many locations, operated by many kinds of people. Unfortunately, some of these locations have compromised or defective networks, and some of these people are untrustworthy or incompetent. Our current design relies on authority administrators to identify bad nodes and mark them as nonfunctioning. We should **automate the process of identifying malfunctioning nodes** as follows:

We should create a generic **feedback mechanism for add-on tools** like Mike Perry's "Snakes on a Tor" to report failing nodes to directory authorities. **(Phase one.)**

We should write tools to **detect more kinds of innocent node failure**, such as nodes whose network providers intercept SSL, nodes whose network providers censor popular websites, and so

We should have **better support for portable devices**, including modes of operation that require less RAM, and that write to disk less frequently (to avoid wearing out flash RAM). **(Optional; end of Nov?)**

3.3 Performance: resource usage

We've been working on **using less RAM**, especially on servers. This has paid off a lot for directory caches in the Tor 0.1.2 release, which in some cases are using 90% less memory than they used to require. But we can do better, especially in the area around our buffer management algorithms, by using an approach more like the BSD and Linux kernels use instead of our current ring buffer approach. (For OR connections, we can just use queues of cell-sized chunks produced with a specialized allocator.) This could potentially save around 25 to 50% of the memory currently allocated for network buffers, and make Tor a more attractive proposition for restricted-memory environments like old computers, mobile devices, and the like. **(Implementation plus measurement. Phase two/three?)**

We should improve our **bandwidth limiting**. The current system has been crucial in making users willing to run servers: nobody is willing to run a server if it might use an unbounded amount of bandwidth, especially if they are charged for their usage. We can make our system better by letting users configure bandwidth limits independently for their own traffic and traffic relayed for others; and by adding write limits for users running directory servers. **(Must do; design, specify, implement, phase one.)**

3.4 Performance: network usage

We know too little about how well our current path selection algorithms actually spread traffic around the network in practice. We should **research the efficacy of our traffic allocation** and either assure ourselves that it is close enough to optimal as to need no improvement (unlikely) or **identify ways to improve network usage**, and get more users' traffic delivered faster. Performing this research will require careful thought about anonymity implications.

We should also **examine the efficacy of our congestion control algorithm**, and see whether we can improve client performance in the presence of a congested network through dynamic 'sendme' window sizes or other means. This will have anonymity implications too if we aren't careful.

(For both of the above: research, design and write a measurement tool, later. See if we can interest a graduate student.)

We should work on making Tor's cell-based protocol perform better on networks with low bandwidth and high packet loss. **(Make a good start; phase three.)**

3.5 Running Tor as both client and server

Many performance tradeoffs and balances that might need more attention. We first need to track and fix whatever bottlenecks emerge; but we also need to invent good algorithms for prioritizing the client's traffic without starving the server's traffic too much. **(Ongoing analysis and experimentation. Early fixes, phase one. Better fixes, phase two/three.)**

4 User experience

4.1 All-in-one bundle

We need a well-tested, well-documented bundle of Tor and supporting applications configured to use it correctly. We have an initial implementation well under way, but it will need additional work

Tor Development Roadmap, 2008-2011

Roger Dingledine

September 1, 2008

1 Introduction

This document describes Tor research and development items that should happen on a three-year timeframe to move Tor forward at being both a useful circumvention tool and a useful privacy tool.

There are two goals to the items in this roadmap. First, we want to make sure to continue adapting Tor to changing environments so it can continue to be a useful tool right now and for the next few years — people are trying to deploy it and use it today, and we want to make sure it stays working well. Second, we want to tackle some of the long-term issues that have been holding Tor back from being useful to a broad set of people — issues that will require some investment now, but will ultimately pay off in creating a more sustainable and automated tool. The tasks here aim to include the right mix between these near-term-useful items and the items that will need several years of design and analysis before they can become useful.

1.1 Outline

There are many pieces to this project. Section 2 describes the core development work that needs to be done to allow us to extend to new features and designs. Sections 3 and 4 cover performance and robustness improvements: load balancing so we can use the available Tor relays in a way that keeps all the traffic moving quickly, making Tor relays work on Windows without crashing, encouraging more users to become relays or bridges, some preliminary scalability work, and work to continue reducing the overhead of directory information. Section 5 adds more circumvention features, focusing on the critical gaps in the current design. Section 6 works on client safety, that is, steps to make sure that educated and prepared users can possibly be secure while using Tor. Section 7 is then client usability: how to make it easier for more ordinary users to still have safety while using Tor, and how to make it more convenient for them to set it up. Last, Section 8 covers not-so-technical development and support work that needs to be done in parallel with the technical development.

1.2 What the various annotations mean

This roadmap aggregates funding and priorities from several different sources. The main sources are funding from BBG and iFree to focus on Tor as a circumvention tool. The others smaller pieces are funding from Google to work on free software development, funding from NLnet to work on directory scalability and on hidden services, and funding from NRL to work on more basic research questions about anonymity and path selection. Getting funding from multiple sources who care about related goals lets us focus not just on the core development, but also on the future features and development that can move us past 'maintenance' level.

In the timelines below, "Year0.5" runs from the beginning of September 2008 to the end of February 2009; "Year1.0" runs from beginning of March 2009 to end of August 2009; etc. For tasks within the one year timeframe, we have also specified which key engineer will be taking the lead on that task. We'll choose key engineers to lead later tasks once we get closer to them.

2 Core development

In order for The Tor Project to continue putting out high-quality software, we need to decide on a roadmap for new features, improvements, and cleanups for each release, prioritize them, and periodically refine the roadmap[1]. Part of the iterative cleanup process is fixing bugs, maintaining existing features, and interacting with users and relay operators to refine our requirements. Another part of the process is coordinating with volunteer and part-time developers to make good use of their energies. More generally, we must be ready to solve whatever other problems and related projects come up. This core development work is what keeps Tor moving forward and keeps new releases coming out.

We have two special focuses for our core development work over the course of this project.

The first focus is on users, especially users in blocked and partially censored environments. How can we give them reliable and adequate performance when they connect to the Tor network, including when they are using obsolescent hardware? This step also includes adding other features that don't warrant their own sections below, such as IPv6 support for destinations and many of the Tor Proposals[2]. **(BBG: Year0.5=30k, Year1.0=30k, Year1.5=30k)**

The second focus is on relays, especially relays and bridge relays run by ordinary Tor users on Windows who use Vidalia for configuration. We've added a lot of features lately that make this process simpler, but we need to make sure to keep these features working well as we work on the next components. For example, we must make sure to keep our resource load (especially memory and CPU) low so Tor relays remain practical on cheap and old hardware — as we add new features that demand more resources, we need to be constantly working on ways to maintain good efficiency and size. **(BBG: Year0.5=30k, Year1.0=30k, Year1.5=30k)**

While this section is funded at an adequate level to keep everything moving, we could still make use of more core development funding: there are many good volunteer developers who would be much more productive if we could give them more attention and help them become core developers. Getting these people up to speed is also a necessary step of figuring out which ones we should pay if we end up with more money, or if we end up with funders with specific target results in mind. **(Unfunded: Year1.0=50k. Medium priority.)**

3 Improve performance through better use of the current network

3.1 Improve Path Selection and Load Balancing

There are many steps to the path selection and load balancing problem. Research needs to be done on integrating load balancing feedback with path selection. Johannes Renner did some initial work here during his 2007 Summer of Code project to extend the Torflow library, and Nikita Borisov of University of Illinois published a paper at NDSS 2008 in this direction [8]. Most recently, Steven Murdoch published a paper at PETS 2008 in Leuven refuting the load balancing aspect of Nikita's design [7].

We shouldn't expect immediate results on this research, because it first requires more measurements and more analysis. We could make some great progress on this in the next year or two though — and since making good use of available resources in the network is one of the critical next steps in making Tor scale better, we should focus on it.

First we need to examine all these proposed algorithms more closely, design new ones as necessary, and simulate the effects of the ones that seem most promising. One example here is looking at 2-hop paths vs 3-hop paths, and comparing the usability, performance, and average network load benefits against the potential for decreased anonymity. Another example is looking at choosing the middle hop of the path based on latencies, or geographical or network locality, to again trade off performance for anonymity. The goal is to identify strategies where we can make a small compromise in expected anonymity for a large expected

[1] See e.g. https://svn.torproject.org/svn/tor/trunk/doc/TODO.021

[2] https://svn.torproject.org/svn/tor/trunk/doc/spec/proposals/000-index.txt

performance gain. Part of the challenge here is to understand what sort of anonymity metrics we should use, so they both reflect reality and are practical to compute.

Good load balancing improvements could double or more the average throughput of Tor users — mainly because the variances that Tor clients are seeing right now are so high, perhaps because a small number of relays are extremely overloaded. Yet other topics to look at include a distributed bandwidth estimation protocol based on Nikita's NDSS paper, and ways to prioritize traffic or circuits to be more fair to as many users as possible.

Another design component we need to consider while we're working on these load balancing algorithms is susceptibility to attack. That is, right now Tor relays self-advertise whatever bandwidth they want. This has led to a variety of attack papers where an attacker signs up an allegedly high-resource relay and attracts a lot more traffic than he otherwise could have handled [2]. So while we design load balancing schemes above, we should aim for ones where the weight for each relay is measured rather than just declared – this could be accomplished for example via community consensus or via a threshold of authorities.

Another topic to tackle is finishing Fallon Chen's circuit timeout work[3]. Her 2008 Google Summer of Code work helped characterize the distribution of circuit build times — the goal is to discard circuits that take more than a standard dev longer than they should, because we already know that they are going to be particularly crummy circuits. Alas, the implementation and verification remain to be done. This step could conceivably reduce the variance of circuit latency a whole lot.

3.1.1 Organize, sort, and prepare

For the Year0.5 deliverable, the main goal here is to lay out all the directions we need to explore, and for each item guess the amount of work it'll take and how useful we think it'll be. Put priorities on each, and for each one describe a path of how to get from here to there. We should also brainstorm about what other measurements and metrics we should gather over time that will let us make better decisions on this topic.

(BBG: Year0.5=20k and iFree: Year0.5=10k led by Steven.)

For the Year1.0 deliverable, we will work on the low hanging fruit from the Year0.5 report, and also get some longer term projects going.

(BBG: Year1.0=20k and iFree: Year1.0=15k led by Mike.)

(iFree: Year1.5=25k and Year2.0=25k led by ???.)

3.1.2 Measure circuits and streams in the wild

At the same time we should build an automated infrastructure to measure and track performance (both bandwidth and latency) in the network over time, so we can observe trends and see what effects our modified algorithms are producing in the network once we deploy them.

This data will be useful for this section in terms of advising us on how to change the designs, and it will also be used in the Metrics work in Section 5.7 so we can graph the health and progress of the network over time and notice trends.

(iFree: Year1.5=10k, Year2.0=10k, Year2.5=20k. Led by Mike.)

3.1.3 Implement and deploy

The final step would involve implementing and deploying new better load balancing algorithms that improve (or at least don't hinder) both performance and anonymity. **(iFree: Year2.0=15k, Year2.5=30k.) (Unfunded: 50k+. High priority.)**

3.2 Tor over UDP, UDP over Tor

Moving to using UDP transport in Tor will provide huge advantages to performance, since user connections will do end-to-end congestion control and we should be able to fit many more connections onto a Tor network

[3] https://svn.torproject.org/svn/tor/trunk/doc/spec/proposals/151-path-selection-improvements.txt

with a given capacity, and since we will tolerate dropped packets without slowing down every stream over the connection that dropped the packet. Moving to UDP transport in Tor will also provide scalability advantages, because each Tor relay doesn't need to hold open a TCP socket to each other Tor relay – meaning we can increase the network capacity, and thus again increase performance. Further, more relays (in particular more diversity of relays) leads to better anonymity for users.

Another advantage of moving to UDP transport is that Tor would now be able to handle connections for UDP-based applications like Skype (VoIP in general), OpenVPN, DNS, etc.

We've been working with Ian Goldberg at Waterloo. He was the Chief Scientist at Zero-Knowledge Systems, a company that deployed a UDP-based anonymity system called the Freedom network that was quite like Tor. He has several grad students who want to work on exactly this problem, and Joel Reardon has written his Master's thesis investigating it.

The first step would be to rederive how the Freedom network worked, combine that with ideas from the above two research groups, and produce a design and specification for how to pass Tor traffic over UDP. This step involves adding sequence numbers and MACs to each packet as it traverses the Tor network, handling and retrying dropped packets during the circuit-level crypto handshake, etc. The second step would be to analyze the design with respect to Tor's current security properties, including perfect forward secrecy from the current circuit handshake, not partitioning traffic across multiple connections, etc. Step three is to figure out a migration plan that allows us to move to the new design within a year or so, and doesn't harm user security or performance too much during migration. (For example, some designs we've seen involve a "flag day" where all users and servers stop using one network and start using a different one.) Then we iterate steps one, two, and three until we get a realistic design that still has adequate security properties.

(BBG: Year0.5=5k to fund Joel's thesis work with supervision by Ian (matched 4x by the Canadian government!) and Year0.5=5k for brainstorming the roadmap here.)

However, there's still a big gap: since we're transporting TCP and UDP packets end-to-end and just writing them onto the network on each side, the details of the TCP stack used on the client side becomes relevant. Operating systems like Windows, OS X, and Linux choose sequence numbers, source ports, and other connection properties in a predictable way, meaning the exit relay, the destination site, or somebody in between can observe the traffic and discover that two connections are coming from the same user. This attack probably works across different exit relay, and probably works across time (e.g. users with a given timestamp skew will probably retain it later too). See https://wiki.torproject.org/noreply/TheOnionRouter/TorFAQ#PhysicalFingerprint

(BBG: Year1.5=20k to take the work Joel and others have been doing and try to turn it into a realistic design and development roadmap. What are the critical missing pieces and what are the steps required to resolve them? Just how hard would it be to maintain a user-space TCP library? Etc.)

So step four is to find, adapt, and/or write a user-space TCP stack that can rewrite and normalize TCP and UDP packets and streams so they no longer contain these identifying properties. This is a huge task, and we shouldn't really plan it until the above phase has given us better intuition. **(Unfunded: User-space TCP stack. 200k-500k?, Low priority.)**

At this point, we will have a convincing design for how to migrate to UDP for connections between Tor relays, and for Tor clients that are in a position to use UDP. But clients in censored areas may still not be able to use UDP for their first hop, because it will have an unusual network footprint. These users would still benefit from most of the above advantages (improved performance inside the Tor network, improved scalability and thus improved anonymity), but they wouldn't get what is perhaps the most important benefit for them, which is tolerating high packet loss to their first hop.

So step five is to reverse engineer Skype, or pick a different popular UDP-based app that is allowed through most firewalls, figure out what security properties it's missing (for example, I bet its crypto handshake doesn't provide perfect forward secrecy), and then either figure out how to achieve our security properties while looking like Skype traffic, or decide to have a second inferior handshake that would provide less security to these users and also partition them from the rest of the Tor user base. Then we iterate steps one-three above until our new protocol seems like the right one to deploy. **(Unfunded: Design and papers and review. 75k, Low priority.)**

Step six is then to implement, deploy, and manage the migration. **(Unfunded: Implement and deploy. 100k, Low priority.)**

3.3 Hidden service performance and reliability

Tor Hidden Services allow users to set up anonymous information services, like websites, that can only be accessed through the Tor network and are protected against identification of the host that runs the services. The most critical limitations of Tor Hidden Services are the time it takes until a Hidden Service is registered in the network and the latency of contact establishment when accessed by a user. Due to design issues in the original Tor protocol, the connection to a new Hidden Service can take several minutes, which leads most users to give up before the connection has been established. Using Tor Hidden Services for direct interactive user-to-user communication (e.g. messaging) is nearly impossible due to the high latency of Hidden Service circuit setup.

This project aims at speeding up Tor Hidden Services by improving the way Tor circuits are set up between the user and the Hidden Service as well as the way a Hidden Service is registered in the Tor network. In a first step precise diagnostics of the behavior of the Hidden Services in lab setups and real world situations will be conducted to find the root causes of the bad timing effects. Based on these diagnostics, optimization strategies will be designed and verified for unwanted implications for the security and anonymity of the Tor network. Precise success metrics will be developed in the diagnostics phase, after it becomes clear where the time is lost and what improvements are realistic.

(NLnet: 45k in Year0.5. Led by Karsten.)

Still need to add in authorization, many further steps to improving hidden service performance and reliability. [say more] **(Unfunded: 30k in Year1.0. Led by Karsten.)**

4 Improve performance through more capacity

Better performance comes from increased network capacity, and better security comes from increased network diversity.

4.1 Relay stability on Windows

Tor relays still don't work well or reliably on Windows XP or Windows Vista, because we don't use the Windows-native "overlapped IO" approach. Christian King made a good start at teaching libevent about overlapped IO during Google Summer of Code 2007, and many volunteers have been working lately on related modifications to the libevent library. The next steps are to A) finish that work, B) teach Tor to do OpenSSL calls on buffers rather than interacting directly with the network, and C) teach Tor to use the new libevent buffers approach. In particular, we can imagine the following roadmap:

- 1. Update libevent to include support for an IOCP backend for its buffering logic.
  - a. Update libevent's bufferevents code to support multiple backends.
  - b. Add a generic multithreading support mechanism for use by libevent backends that need multiple threads.
  - c. Make libevent's bufferevents base logic threadsafe, as needed.
  - d. Merge, adapt, and rewrite IOCP code.
  - e. Evaluate performance and optimize.
- 2. Update libevent's buffering logic so that it can support SSL-over-buffers, as Tor requires.
  - a. Determine appropriate APIs for buffering abstraction. Niels and Nick are discussing filtering versus function tables as an appropriate mechanism.
  - b. Implement these APIs.
  - c. Implement an OpenSSL wrapper with the flexibility we need.

- 3. Other libevent hacking as may be required.
  - a. Minimize uses of "pullup" function on libevent buffers.
  - b. Evaluate whether any of Tor's buffer RAM hacks (like freelists) are actually useful for libevent.
  - c. Make sure that all of the functions currently implemented in Tor's buffers can be trivially cloned by libevent's.
- 4. Update Tor to take advantage of new libevent features as available.
  - a. Rewrite Tor's basic networking layers so that it contains an element that behaves equivalently to libevent's buffers. This process will inform 3c and 2a above, and may require us to revisit those steps with iterative refinements.
  - b. Have this layer use new libevent code when it's present and efficient.
  - c. Evaluate performance; continue to optimize.

(iFree: Year1.0=45k for a first cut of the above steps. Then Year1.5=15k, Year2.0=15k, Year2.5=15k to make sure it works smoothly. Led by Nick.)

4.2 Tor clients that find themselves reachable and reliable should automatically become a bridge or relay.

We've made a lot of progress towards letting an ordinary Tor client also serve as a Tor relay, and we will continue to make progress as we move forward. There are several more topics that need investigation still:

4.2.1 Risks from being a relay

Understand the risks from letting the attacker send traffic through your relay while you're also initiating your own anonymized traffic. Three different research papers [1, 5, 6] describe ways to identify the relays in a circuit by running traffic through candidate relays and looking for dips in the traffic while the circuit is active. These clogging attacks are not that scary in the Tor context so long as relays are never clients too. But if we're trying to encourage more clients to turn on relay functionality too (whether as bridge relays or as normal relays), then we need to understand this threat better and learn how to mitigate it.

One research direction is to investigate the RelayBandwidthRate feature that lets Tor rate limit relayed traffic differently from local traffic. Since the attacker's "clogging" traffic is not in the same bandwidth class as the traffic initiated by the user, it may be harder to detect interference. Or it may not be.

We aren't really comfortable setting users up en masse as bridges or relays until we understand these issues more.

At the end of year1, we're going to start on the proposal in Section 4.2.2 below for how exactly Tors will measure themselves and decide that they should elect to be a bridge relay and/or normal relay. That is, at the end of year1 we'd like to have a clear plan for how users who become relays will be safe. We want to be confident that we can build this plan.

This means the main tasks for the research in year1.0 are: a) evaluate all the various attacks that are made possible when an attacker can use you as a relay, and b) identify ways to make them not a big deal. Then we should c) pick the right way or ways, and spec them out such that we believe they will work and be possible to implement.

This work might involve simulation, might involve analysis, will probably involve messing with Tor's round-robin read/write algorithms, etc.

(BBG: Year1.0=10k and iFree: Year1.0=30k led by Steven.)
(iFree: Year2.0=30k and Year2.5=40k led by ???.)

4.2.2 First a bridge, then a relay.

I want all clients to start out automatically detecting their reachability and opting to be bridge relays. Then if they realize they have enough consistency and bandwidth, they should automatically upgrade to being non-exit relays.

(iFree: Year1.5=5k led by Roger for a proposal.)

(BBG: Year1.5=15k for a development roadmap on what internal Tor pieces will need to change and how they should change.)

(iFree: Year2.0=10k and Year2.5=15k led by ???.)

(Unfunded: lots)

4.3 Incentives design

Roger has been working with researchers at Rice University to simulate and analyze a new design where the directory authorities assign gold stars to well-behaving relays, and then all the relays give priority to traffic from gold-starred relays. The great feature of the design is that not only does it provide the (explicit) incentive to run a relay, but it also aims to grow the overall capacity of the network, so even non-relays will benefit.

However, the current incentives design we invented has a serious flaw, which is that the set of gold-starred relays is known to the adversary, and over time he can narrow down which gold-star users are always the ones online when a certain activity (e.g. posting to a blog) happens. We need to revamp the design so the set of high-priority users and the set of currently online relays is less clearly related.

Also under this heading are better algorithms for giving priority to local traffic. Proposal 111 made a lot of progress at separating local traffic from relayed traffic, so Tor users can rate limit the relayed traffic at a stricter level. But since we want to pass both traffic classes over the same TCP connection, we can't keep them entirely separate. The current compromise is that we treat all bytes to/from a given connection as local traffic if any of the bytes within the past N seconds were local bytes. But a) we could use some more intelligent heuristics, and b) this leaks information to an active attacker about when local traffic was sent/received.

The deliverable here is a revised and peer-reviewed design paper, but we hope to augment this with external funding to implement and deploy too.

(BBG: Year0.5=10k and iFree: Year0.5=10k led by Roger to revise and publish the current draft and start gathering comments and new designs.)

(iFree: Year3.0=20k led by ??? to put out a new design doc.)

(Unfunded: we could move up the timeline here with more funding and more attention. This is potentially a very large area to tackle.)

4.4 Continue research on how to splinter the network as it grows so we can maintain a good balance of both anonymity and scalability.

This topic is probably the hardest open research problem in the field right now. We need to enumerate and analyze the various solutions we've come up with already, and work on new better solutions.

Step one is to specify the details for the simple version: partition the networkstatus documents as they get too large, and have clients fetch and use only a single partition, and have mirrors only cache descriptors from within their partition. Analyze the scalability, performance, and anonymity properties therein. We've made a start in the recent PET 2008 paper by Danezis and Syverson [3].

(iFree: Year1.5=10k and Year2.0=15k to write a proposal, led by Peter.)

(Unfunded: We'll look for more money in Year3 to revise and/or build it.)

4.4.1 Blending the network partitions

Step two is to research variants of this design that "blend" multiple partitions together. [more detail here]
(Unfunded: 100k, Medium priority.)

4.5 Clients download less directory info. Especially useful for clients on modems.

See "piece one" in https://www.torproject.org/svn/trunk/doc/spec/proposals/ideas/xxx-grand-scaling-plan.txt

The challenge here is that many of the design decisions for this topic impact other scalability and partitioning decisions down the road: that is, what we do here will decide what options we have for all the other designs. So we need to think very carefully so we don't introduce new vulnerabilities and so we don't preclude other design changes.

The start of this design is Proposal 138[4] which will cut the size of the networkstatus from about 2500 relays to 1500 relays by removing the ones that most clients won't download; Proposal 140[5] which will let clients only download a diff of one consensus to the next, rather than downloading a whole new consensus; and Proposal 141[6] which lays out a plan for the just-in-time descriptor download design, which would cut out a huge amount of the directory overhead.

(NLnet: Year0.5=40k for analysis of the various designs and tradeoffs.)

(iFree: Year0.5=10k for support of the analysis, then Year1.0=10k to figure out a transition plan, and Year1.5=10k to deploy. Led by Peter.)

4.6 Advocacy for running more relays

There are several components here.

First, we need to reach out to the right communities. There are lots of technical organizations out there who would be happy to help if only they realized the need and understood how straightforward it is.

Second, we need to streamline the instructions, including coming up with an easy set of screenshots for configuring a relay with Vidalia. Right now the web pages we point people to are years old. They're still correct, but in the mean time we have made many of the steps easier, and we should update the docs to reflect these changes. As an example, Tor on Debian now starts as root, so it can bind to port 443 directly without needing any clunky iptables rules to do port forwarding — but we haven't written this down anywhere useful.

Third, we need to maintain connections to the people who have set up the relay. I think the number one reason relays disappear is because the people who set them up don't think anybody cares anymore. To this end, Jacob has been working on a "Tor weather" website[7] that lets people sign up to get email when a relay disappears. This could be used either by the relay operator, or by our advocates to get reminders when somebody needs a nudge.

Fourth, we need to work on ways to inform people about our progress at growing the network. Some of this task will be started in the Metrics work in Section 5.7. We need to come up with flashier ways to show our growth, perhaps through Tor network maps that visualize the data in a slick way, through Facebook reputation for relay operators, or through other ideas that come along.

We also need to investigate how well our "get people to offer their ORPort on 443" strategy is working, because those relays are most useful for folks in firewalled areas.

(BBG: Year0.5=20k to get Tor weather up, stable, and in use by some relay operators and some relay advocates. Year1.0=10k and Year1.5=10 for the network maps work and also maintenance. Led by Jacob.)

5 Suitability for circumvention

5.1 Normalize our network fingerprint even more

Play the TLS handshake arms race as needed.

The Year0.5 milestone here is to produce a list of the likely avenues we anticipate for blocking, and for each avenue build a plan for how we should respond to get Tor unblocked again. We don't intend to deploy fixes until pushed by the arms race to do so, but we should work towards having the fixes ready or close to ready so we don't look like idiots for a month.

[4] https://svn.torproject.org/svn/tor/trunk/doc/spec/proposals/138-remove-down-routers-from-consensus.txt
[5] https://svn.torproject.org/svn/tor/trunk/doc/spec/proposals/140-consensus-diffs.txt
[6] https://svn.torproject.org/svn/tor/trunk/doc/spec/proposals/141-jit-sd-downloads.txt
[7] https://weather.torproject.org/

Then at each milestone we'll revisit our list and revise our plans as needed.

(BBG: Year0.5=10k, Year1.0=10k. iFree: Year1.0=10k, Year2.0=10k, Year3.0=5k. Periodic adjustments as Smartfilter and Websense do their thing. Led by Nick, with red teaming from Steven.)

5.2 More bridge address distribution strategies

Assess more bridge address distribution strategies, based on a broader set of technologies like SMS, radio, WoW, etc. Many of these approaches will require more manual ongoing attention than our first few approaches.

In the meantime, we need to design and deploy other bridge address distribution strategies that make the process a bit more automated for the users. In particular, we're thinking about a "bridge loop" design where bridge identities form a "loop" at the bridge directory authority, and if you know any bridge in the loop you can learn all the others. This approach will allow Tor clients who know a few bridges to be updated with new bridges as their old ones rotate, without opening up the list to full enumeration.

(iFree: Year2.5=25k.)

5.3 Layered guard nodes for bridge users

Decide whether bridge users need to choose a second "layer" of entry guards, so it's harder for an ordinary Tor middle server to enumerate bridge relays just by seeing who connects. Start solving this problem somehow, for example by making bridge users do the above.

I think this is going to be necessary in the near term, since if it turns out to be a real attack, it is a very practical one. **(iFree: Year2.0=10k for analysis led by Roger, Year3.0=40k for deployment led by ???.)**

5.4 Tracking bridge reachability

Better and more automated measurement tools for whether bridges are actually up, and actually reachable from inside target countries.

"Actually up" is quite straightforward: we already do simple reachability testing from the bridge authority. Tracking reachability from inside target countries will be a statistical game based on how many geoip details we can collect from the bridges themselves. See the "better user metrics" item below.

5.5 Email auto-responder

Email auto-responder so for example gmail users can fetch the Tor software via email. Social network distribution techniques. Continue beating on this problem.

For Year0.5, we'd like a design proposal that describes what commands it responds to and how it responds; how it interacts with DKIM; and a plan for supporting many languages. We should put out a prototype so experts can use it, and to get more intuition about our requirements.

For Year1.0, we should polish it some more and get some outside help with translation, how to phrase the messages, and how to make it more intuitive to users.

(BBG: Year0.5=10k, Year1.0=5k. iFree: Year0.5=10k, Year1.0=5k for design and setup.)

(iFree: Year2.0=5k and Year3.0=5k for operation. Led by Jake.)

5.6 Research: how many bridges do you need to know to maintain reachability?

We need to track the churn of bridges over time and then analyze how many bridges are smart to know, or how often it is smart to learn new bridges, in order to stay connected. If a few bridges are likely to last a long time, we can focus on other problems. On the other hand, if even a half dozen bridges are likely to

all vanish soon, we need to work both on encouraging bridge stability and on better algorithms for learning about new bridges.

Then we need to watch for trends and changes over time to see if our job is getting easier or harder.

(iFree: Year1.5=5k and Year2.0=15k to assess what and how to store data and then store it, Year3.0=30k for operation and analysis. Led by Karsten.)

5.7 Better user metrics and measurements

The proposal wants us to demonstrate that we'll be able to measure progress. There are a number of ways to do this, but most of them have anonymity implications so they must be approached carefully.

Item one is to automate the collection of current network traffic statistics: volume of data over time, number of available relays each day, total advertised capacity for the day, etc. We've been collecting some of this data already so we can have some baselines, but we need to collect it in a more reliable fashion (i.e. pay somebody to be sure to do it), and we need to redesign and rewrite our scripts for processing it into useful graphs and figures since we're talking dozens of gigabytes of input data.

Item two is to clean up and maintain the "exitlist" service we are aiming to offer, so it's easy for other sites to check if a given IP address was a Tor exit relay at a given point. This will let websites measure how many of their users are coming from Tor. We will also need to provide some glue, and maybe training, to let people convert apache logs into useful information about the fraction (and variety?) of Tor users. To be most effective, we will want to answer not just "is this IP address a Tor exit relay now?", but also "was it an exit relay in the recent past?"

Item three is to implement the "geoip lookup" designs that Nick has been designing in 2008. These should give us a better estimate of the overall set of current Tor users; and as we migrate to the directory guards design, it should still be able to give us some rough numbers.

Item four is to try to handle the skew in our user geoip stats from dynamic IP addresses. Currently we're forced to just measure a rolling 24 hour window, since most users in Germany and China get a new IP address daily. Thus our metrics leave out most of the people who use Tor only infrequently – which could be a big part of our target population. Can we find or generate a database of Internet address blocks that are associated with frequent cycling? If so, we could discard the sightings of those users after 24 hours, while accumulating the other sightings over a longer time period. We might be able to bypass this step if we can do the geoip lookup designs based on measuring something we know users do with a certain frequency, like fetching a new copy of the consensus networkstatus – in this way we are measuring the number of Tor clients running rather than trying to measure the number of IP addresses in use.

Item five is to investigate whether we can safely switch the current website logs over to doing a GeoIP lookup before discarding the IP address. Being able to track downloads or page views over time is not a perfect metric, because it doesn't include downloads from other sites (e.g. Debian repositories) or downloads shared among friends, but it is still a metric. Do we need to randomize or otherwise lock down the logs to make this increased data gathering acceptably safe?

Item six is to track and graph the countries and number of users that bridges report. Right now we're accumulating a big pile of "extrainfo" descriptors for bridges, but we aren't doing anything with them. We should write scripts to parse them, remember the important details, and maintain periodic snapshots so we can look for trends over time.

Item seven is to do some analysis about how to detect country-wide blocking events based on trends in the above data. For example, if the website sees a sharp drop-off in hits from a given country, perhaps the site is blocked by that country's firewall or ISPs. Similarly, a sharp drop-off in reported GeoIP stats from bridges or directory guards could show that the country is starting to block access to bridges and/or block Tor's network signature. The challenge here is that when there are very few data points, it's hard to learn something statistically significant from losing a few. Is there a way to aggregate the overall data points in a way that makes it easier to notice changes?

Since Karsten will be finishing his PhD at the end of 2008, he will be ramping up on this topic in 2009.

(iFree: Year0.5=10k, Year1.0=30k, Year1.5=25k, Year2.0=25k, Year2.5=30k, Year3.0=25k. Led by Karsten.)

(Unfunded: the current funding level is enough for a bit less than half of Karsten. We could easily make use of a full-time smart researcher on this topic.)

5.7.1 Year 0.5

The deliverable that we imagine for year 0.5 is a report that answers the following questions:

a) What information do we want to present? What are useful statistics? This includes all metrics mentioned above, and also the metrics described in Sections 3 and 5.4.

b) What data are available for these metrics? This includes public data (network consensus, router descriptors, extrainfos) as well as non-public data (directory requests, website requests, bridge usage).

c) What data are missing that are required to provide these statistics? Can this data be provided publicly or not? Is it a good idea anonymity-wise to collect that data?

d) What systems are available that present some of these statistics? Can we extend one or more of them, or should we start from scratch? (How likely is it that we'll be able to work together with Kasimir Gabert on his TorStatus project? Further, it might be a good move to have Sebastian Hahn, one of our successful Google Summer of Code 2008 students, work on the PHP part at some time.)

5.7.2 Year 1.0

The idea for the remaining six months of the first year is to implement the metrics concerning public data. These are 1) network statistics and 2) the exit script and log aggregator.

a) The current plan is to extend the implementation of torstatus to provide more information about network statistics. Torstatus currently takes all of its information from current network documents (consensus, router descriptors, extrainfos), but does not keep a history itself. The first step would be to extend the database to keep a history. (The previous analysis should help in estimating what kind of data is required.) Further, there should be some basic statistics that display the historical collected data. [This is a huge milestone, and our original idea was to separate collecting data and creating statistics; but a deliverable that just fills a database isn't as shiny as one that actually displays new information.]

b) We should try to integrate historical data (e.g. weasel's descriptor archives) into the database to make the network statistics complete.

c) Extend the exit list and write a script to locally analyze webserver logs. It *might* be better to use and extend the exitlist functionality of torstatus for this instead of our own exitlist. Otherwise, there is a certain overhead for getting comfortable with two systems and putting in a history database. We will know more about that after finishing the analysis.

d) Integrate historical data into the exitlist database.

It's hard to estimate how much time these tasks will take before actually having performed the analysis from Year0.5. We may end up moving some of them to the Year1.5 milestone. Item d is the one most likely to be punted.

5.7.3 Year 1.5 and 2.0

The plan for the second year is to tackle the metrics based on non-public data: 1) directory requests, 2) website requests, 3) bridge usage. This includes collecting data at their source (if not already done), aggregating them, and transferring them to the statistics portal. The portal would keep a history of these data in its own database and display the data nicely. Handling non-public data is more complicated than public data, because we can't just grab the data ourselves, but need to preprocess and collect them at different places. (The assumption here is to make statistics available for the public. If we should change our opinion on that, another (non-web-based) application might keep the history and generate reports.)

This phase would also involve gathering the stream and circuit measurement data from Section 3.1.2, so we can look for trends over time.

Yet another part for year two is researching how many bridges are needed to maintain reachability (Section 5.6).

5.7.4 Year 2.5 and 3.0

Year three would be used to handle the more advanced topics: 1) Filter out dynamic IP addresses from logs and 2) detect country-wide blocking events.

The other task for year three is operation and analysis of Section 5.6.

5.8 Distributed bridge authorities

Make our 'single bridge authority' design into a 'redundant bridge authorities' design, so bridge publish/lookup can't be knocked over or broken into by attacking a single location. ...

(Unfunded. This will be hard to do right.)

5.9 Research: scanning-resistance

5.10 Research: hiding whether the user is reading or publishing?

6 Client safety

6.1 Automatic update

Tor currently has no mechanism for updating clients in the event of security vulnerabilities and changes to blocking mechanisms. To be effective in the arms race, we will need one.

Step one is to work on auto-update-of-Tor features inside Vidalia:

- looking at the majority-signed networkstatus consensus to decide when to update and to what version (Tor already lists what versions are considered safe, in each networkstatus document),
- doing the update either via Tor or via the directory mirror update protocol (proposal 127) when possible, for additional privacy,
- checking package signatures,
- giving the user an interface for these updates, including letting her opt to migrate from one major Tor version to the next.

This should work on both Windows and OS X. Ideally we would adapt some third-party lib to do parts of this for us – but we haven't found any good free-software security-oriented auto-update lib out there yet. Then we need to work on auto-update for Vidalia itself too, as well as for other supporting applications like Polipo. We also need to produce better plans around security issues like signing key rotation so we can be sure to provide safe and secure service over time. **(Google: Year0.5=50k.)**

(BBG: Year1.0=20k to get it rolled out to experimental users.)

Then we should take a step back and revise our design into a thin client that's separate from Vidalia. It should know how to fetch new versions of the components it wants and check their signatures appropriately. We could imagine this approach as a tiny bundle stub that bootstraps itself into a full Tor bundle; it would also replace our current stopgap workaround for modem users who can't fetch the entire Tor Browser Bundle without losing their connection. We would still want to ship the full bundle too for folks who want to get the whole set and carry it around with them, though. **(iFree: Year1.5=10k, Year2.0=10k.)**

Then we need to look at how Firefox's automatic update scheme works with our automatic update scheme in the context of the Tor Browser Bundle. What about other FF extensions inside the bundle? Which app(s) should be the one managing the automatic update? Does Firefox do anything smarter than check the SSL certificate of the site that it updates from? **(iFree: Year2.5=30k.)**

6.2 Vidalia development work

a) Integration for PlaintextPorts warnings and other status events. Vidalia should walk the user through recognizing that connections to plaintext ports (e.g. 110, 143) probably indicate a bad move on the user's part.

b) More generally, there are a variety of status events that Tor currently sends to Vidalia (e.g. for reachability testing as a server), but Vidalia has no way to present them usefully to the user. We should come up with a generalized way to interact with the user when errors or warnings occur.

c) We should put Polipo into the main bundle, and teach Vidalia to launch it and close it.

d) Bridge usage display. If you're a bridge relay, then you know what countries you've got users from — you publish the aggregated lists in your extrainfo descriptor. We should also put that in the Vidalia display, so you can get a warm fuzzy feeling about saving the world.

e) Look into better error and crash reporting mechanisms for Vidalia under Windows. Google had a nice-looking one called Breakpad[8] that we should try out.

f) Let Vidalia change languages without needing to quit and start again.

g) Other development items and GUI support items as they come up.

(BBG: Year0.5=10k, iFree: Year0.5=10k for a-d. Led by Matt.)

(BBG: Year1.0=10k, iFree: Year0.5=15k for e-g. Led by Matt.)

(iFree: Year1.5=15k, Year2.0=10k. Led by Matt.)

6.3 Node/Network scanning

We've written a prototype node scanner called 'SoaT' to scan the Tor network for malfunctioning and malicious relays; we presented it at Black Hat and Defcon 2007. Currently, it scans for malicious content injection at the exit relays by checking MD5 sums of documents. It would be nice to produce a more flexible fingerprint of a page, perhaps based on the javascript/object/image content only. Additionally, it would be nice to integrate scan results into the Tor directory consensus, so clients can use the information in their routing decisions to avoid malicious or failing relays. Passive scanning and reliability reporting from clients and relays to the directory servers is also a possibility, but some of this may need experimentation and research[9].

There are several components here. The big first steps are A) a more automated scanning mechanism, B) coming up with plausible tests that are hard to distinguish from "normal" fetches, and C) integrating the results into the directory authorities so Tor clients get quick feedback about problems we discover.

The first cut at this tool would focus on making it easy to configure and run on a variety of platforms, giving it a good suite of plausible-looking tests, and feeding the results into a directory authority. **(iFree: Year0.5=10k, Year1.0=10k, Year1.5=15k, Year2.0=15k. Led by Mike.)**

Then we should make the tool more automated and easier to run for long periods unattended, including ways for users to submit tests directly to the database ("I'm having problems with the following page or website, please add it to the test suite") and other ways to add the client reliability reporting described above. **(iFree: Year2.5=15k, Year3.0=10k. Led by Mike.)**

6.4 Torbutton development

We need to stabilize and then continue the "Torbutton-dev" work so we can give people a robust Firefox extension that not only lets them toggle Tor on/off, but also protects them from many application-level threats on the web.

We've made a huge amount of progress already[10], but many more issues remain — in particular, while we've been focusing lately on making sure that Torbutton *can* provide safe browsing, it probably is at odds with usability and useful browsing in many ways. We will need to work with users to find the right balance between safety and usefulness.

Further, Torbutton will need constant attention to maintain the current features as Firefox's internals change. We have filed several dozen Firefox bugs, and are working with Mozilla to resolve them. Alas, there will continue to be more.

[8] http://code.google.com/p/google-breakpad/

[9] http://fscked.org/transient/SecuringTheTorNetwork.pdf

[10] https://www.torproject.org/torbutton/design/

Look into ways for Vidalia and Torbutton to communicate. For example, some link seems necessary for the "new identity" button to function properly when both Firefox and Polipo are determined to do keepalive on their current connections.

(iFree: Year0.5=20k, Year1.0=20k, Year1.5=15k, Year2.0=10, Year2.5=10k. Led by Mike.)

6.5 Evaluate new anonymity attacks

There have been a variety of new anonymity attacks published recently, and more on the path to publication.

We need to work with the authors of these papers to help them show the efficacy of their attacks in ways that we can grasp / believe / reproduce; and we need to brainstorm practical solutions or mitigations for the scenario where the attacks do in fact work (or can be made to work).

(iFree: Year1.0=20k, Year2.0=20k, Year3.0=10k. Led by Steven.)

(BBG: Year1.5=20k.)

(Unfunded: This is an immense but critical task, and we plan to get most of the funding for it from other sources, but this item will make sure that it gets at least some attention.)

6.6 Evaluating traces from Tor Browser Bundle, Tor VM, etc

We've made a good start at enumerating the traces we see from the current Tor Browser Bundle running Firefox 2. As we note in Section 7, it will need ongoing attention as we move to Firefox 3 and as other changes are made. Further, other mechanisms for launching Tor, such as the QEMU approaches for Incognito and Tor VM, will leave still different sets of traces.

Beyond just enumerating the traces and evaluating their severity, we also need to work to reduce the impact of each trace. How many of them are impossible to get rid of? How many issues are resolved by running most of the sensitive programs inside a VM? What about if we moved to Vista?

(BBG: Year0.5=5k. Led by Steven.)

(Unfunded: We could probably make more progress on this, but we really need to bring in outside forensic experts. And even then, the prognosis doesn't look good about actually being able to do anything about the traces. Windows is bad like that. Maybe this is something iFree wants to pick up?)

6.7 Is your browser config safe?

We need test websites that can evaluate at a glance whether the user is vulnerable to the wide variety of web-based exploits we've already discovered. This "check" website would be useful for several goals:

- We could use it for regression tests as we make new versions of Torbutton, and as Firefox makes upgrades that might reopen new problems.
- We could use it to evaluate other browsers and other extensions that people might try to use instead of the recommended Firefox + Torbutton combination.
- We could use it to help users confirm that they are in fact using Torbutton in the correct configuration.

Some of these steps, especially the last, could even be done by shipping a tiny webserver inside Tor, and letting the browser interact with it locally. This approach has the benefit of being a much more controlled environment, where we can avoid confusion due to outside variables; but it has the drawback that we can't automatically update it for new or better tests in the way that we can update a central test website.

(Unfunded: We'd like to get to this in the Year2 range; but doing it right will require a lot of attention.)

6.8 Documentation and explanations of how and when to use Tor

There are a few starts to these docs out there, like Ethan's GV tutorial[11]. But I bet people would like some more detailed, more updated walkthroughs too — also covering other activities like setting up instant messaging. Some of this is hopefully going to be covered by the NGO-In-A-Box folks, but I don't know how indepth they go.

(iFree should work on this with its own funding.)

6.9 Recommended configurations and applications

Figuring out recommended configurations for various apps and figuring out which apps in each category to recommend. We're doing better at this now that we have the Tor IM Browser Bundle, since it handles IM, IRC, etc. But many people still want to use their own apps, or at least apps that aren't quite as klunky at Pidgin. We need to investigate the default configurations of apps that users want to use in the wild, and see if we can either a) come up with some instructions for how to use them safely, and/or b) come up with some recommendations for which ones we better than others. Some explanations ("here's why you shouldn't use IE") would probably help give people some intuition.

(iFree should work on this with its own funding.)

6.10 Safe path selection when you know more than other users

NRL Research and development project. [say more]

(NRL: Year0.5=27k. Led by Nick.)

7 Client usability

7.1 Transparently intercepting connections

Currently, Tor clients need to actively configure their applications to use Tor as a SOCKS proxy. This step results in many confused users, as well as (probably) many users who are using Tor in an unsafe configuration. Even for web browsing, many of the challenges that Torbutton attempts to solve come from ways that websites can trick users into bypassing their proxy settings. And simply disabling all of these avenues results in usability problems, e.g., preventing users from watching Flash videos at Youtube.

Some tools like Xerobank VM and JanusVM use a virtual machine (generally VMWare running a Linux OS) to run the Tor client and web proxy, and they use the VM's ability to intercept outgoing connections so they can redirect them into Tor.

It might be that the VM approach is necessary, in which case we should work on adapting QEMU so our bundles are not subject to VMWare's restrictive redistribution licenses. Or it might be that we can just reuse drivers like the ones OpenVPN uses to transparently intercept the outgoing connections and pass them into the Tor client.

We need to research the options and implications, and document the problems that can come up — for example, trying to pass outgoing email connections into Tor will not work very well, since few Tor relays have an exit policy that will allow outgoing email. We may end up needing an administrative interface for which addresses and ports should be sent into Tor; but we'd like to find a more usable solution than that.

Ultimately we need to deploy a Windows Tor client that's wrapped in a VM with transparent proxying. Done right, this should be the default way that Windows users interact with Tor.

(BBG: Year0.5=10k and Year1.0=10k; iFree: Year0.5=20k and Year1.0=20k for research, analysis, and prototyping. Led by Martin.)

(BBG: Year1.5=15k. iFree: Year1.5=15k, Year2.0=10k, Year2.5=10k for deployment. Led by Martin.)

[11] http://advocacy.globalvoicesonline.org/projects/guide/

7.2 Tor Browser Bundle

Lock down the Tor Browser Bundle, make it more robust, etc. Deal with deployment issues, updating features and maintenance for new versions of the component software, translations, etc.

Year 0.5 items:

- Work on a new launcher out of Vidalia rather than using Portable Firefox. This new launcher would let Vidalia recognize better when Firefox has closed.
- We'd like to make it possible to run a TBB Firefox and a normal Firefox in parallel, but without putting a lot more money in this item, we can't promise it in case there is some weird FF bug that blocks us. We will either do it or determine exactly what weird FF bug it is that blocks us.
- Switch TBB to Firefox 3, once Torbutton 1.2 is stable enough on it.
- Gather a new set of traces, once we have a new launcher and once we switch to Firefox 3.

Year 1.0 items:

- Port Tor Browser Bundle to OS X.
- Make TBB the recommended Tor download for Win / OSX.
- Make sure it's easy to switch to an unbranded Firefox (e.g. called "Tor Browser") on short notice, since we might find ourselves in exactly that situation.
- Evaluate CCC's "Freedom Stick" version of TBB. Decide if we like the approach they took, and if we want to recommend it or suggest some changes.

Year 1.5 should include investigating what else to put on a USB image for TBB. Can we do some disk encryption, so if we decide to put out USB images with a different bridge relay for each image, simply swiping the USB key isn't enough to learn about the bridge? This would also be useful if we want to give out per-key hidden service authorization.

(iFree: Year0.5=10k led by Steven.)

(BBG: Year1.0=10k. iFree: Year1.0=10k, Year1.5=10k, Year2.0=10k, Year2.5=10k led by Jake.)

7.2.1 Deploying USB keys with Tor Browser Bundle on them

Now that we've got the software itself working pretty well, we should figure out all the details like where to buy USB keys in bulk, how to silk-screen a logo onto them, what size they should be, whether we should aim for those cool USB form factors that look like smart cards or buttons or whatever. Should we include an auto-start file for the Windows users when they pop the key in? Should we include a few tutorials on how to stay safe on the Internet? What else? I wouldn't recommend burning tens of thousands of these yet though, since we're hoping to have improved versions in a year or two (e.g. using QEMU and/or with an auto updater).

(iFree should work on this with its own funding (with periodic advice from us).)

7.3 Playing nicely with websites

Right now Wikipedia and some other services block posts from Tor users, due to a few abusers. Tools like Nymble[4] allows these services to recognize the abusers later without ever needing to learn who or where they are – thus allowing the non-abusers to start using the service like usual again.

The first step here is to write down a clear explanation of why anonymity is not at odds with open access to sites like Wikipedia and Slashdot. Roger gave a talk at Wikimania 2006 to explain to the Mediawiki developers about other options they have besides blacklisting IP addresses[12]. Once we have this essay ready, we can start to educate other people in the community about the fact that practical options do exist. The Wikipedia community is diverse though, and can be stubborn, so this will not be accomplished quickly. At

[12] `http://freehaven.net/~arma/slides-wiki-tor.pdf`

the same time we need to work with the legitimate Tor users (e.g. the ones who would like to edit Wikipedia through Tor) and help them to adopt some interim solutions, if any. **(iFree: Year2.0=15k for advocacy, led by Roger. Year3.0=15k for operations, led by ???.)**

The next step (after this project) is to help deploy a system like Nymble, which is an infrastructure for letting websites blacklist abusive users without needing to (or being able to) unveil their location or identity. They are apparently working on an implementation right now, but it will definitely need help with deployment, usability, and sustainability.

7.4 Let users specify their exit country

As the network grows, and censorship gets more varied and widespread, exiting from a censored country becomes more of a hassle.

The first steps are a) changing the Vidalia interface to let you communicate which country you want to exit from, b) turning countries into sets of Tor servers that we're pretty sure are in those countries, and c) making sure that the Tor client does in fact do the right thing with the set of preferred servers (the basic features have been in for a while, but nobody uses them for much and they likely have some problems). **(iFree: Year2.5=20k, led by ???.)**

The next steps after this contract are to tackle making the Tor client scale well when you specify really large lists of server IDs either in the "use these" or the "don't use these" side, work on more complex interface approaches ("use these two countries but not this one"), specify other constraints the user has in mind ("please only pick servers with certain uptime, certain bandwidth, certain operating system, etc"), look into the anonymity issues with choosing a smaller set of options at each step, look into the security questions of using geoip data vs whois data for country codes, let us build a more efficient interface (that doesn't require interacting with the GUI) by specifying country properties in the url (like www.google.com.cn.exit) without actually broadcasting this url to the destination website in the Host: http header, and figure out various other issues that come up as we start deploying a solution. We may get follow-on (external) funding to move this part forward.

7.5 LiveCD

We need a nice bootable LiveCD containing a minimal OS and a few applications configured to use it correctly. The Anonym.OS project has demonstrated that this is quite feasible, and there is a nice student in Sweden who is currently trying to maintain it for free.

It needs more documentation, more analysis about whether its configuration choices are the right ones, and more thought for what applications to include so it has enough that it's useful — and what apps to *exclude* to keep its complexity from dragging it down.

Most appealingly, Anonym.OS has recently added a feature allowing it to be booted from inside Windows as a standalone system that provides a set of self-contained correctly-configured applications chosen for their good security. This approach could provide the best combination of the transparently intercepted connections item above with the Tor Browser Bundle item. The tradeoff is download size.

(BBG: Year0.5=5k, Year1.0=5k, Year1.5=5k. Responsive support and feedback for the volunteer maintainer.)

(Unfunded: At some point we should ramp up development and support for Incognito. 40k security analysis, 25k maintenance. Medium priority.)

7.6 Translation coordination and automation

We've set up a prototype online translation website[13] based on Pootle. We've also set up some preliminary documentation[14] for how to interact with the website and provide translations.

[13] https://translation.torproject.org/

[14] https://www.torproject.org/translation-portal

First, we need to continue to coordinate and maintain our current volunteer translators. They do a tough job basically for free, so we need to keep making sure their questions get answered promptly and making them know they're appreciated. We should keep working on ways to let them stay notified when new translations are needed, and to let them track their translation progress. **(BBG: Year0.5=5k, Year1.0=5k, year1.5=5k)**

Second, while we've successfully transitioned most of our file formats over to the Pootle interface, we still haven't finished our "wml2po" (website to .po) converter, so there's no way in the Pootle web interface currently to translate the Tor website pages. We were hoping to get a good start on that from one of our Google Summer of Code 2008 students, but that didn't work out. So, we should get that going ourselves. The really tricky parts are: A) we need to automatically break up wml files (basically html files) in such a way that we get one 'idea' per piece: we're currently thinking that splitting on ¡p¿ and similar tags should do it. B) If we change a paragraph just a little bit, we would like some way to recognize that the previously translated string is a good place to start when translating the new paragraph. In the obvious way to break up a web page into strings, the new string would not be the same as the old string, so we'd have no way of knowing they're related. We need to either label the paragraphs in a way such that the label stays the same as we edit the paragraph, or do something even smarter to recognize when a new paragraph is 'likely' to be a derivative of an old one. **(BBG: Year0.5=5k, Year1.0=5k, Year1.5=5k)**

(Unfunded: Another 25k would get this second step done more reliably.)

7.7 Usability testing of Tor

Especially the browser bundle, ideally amongst our target demographic. That would help a lot in knowing what needs to be done in terms of bug fixes or new features. We get this informally at the moment, but a more structured process would be better.

(iFree should tackle this through their own funding.)

8 Non-development supporting tasks

8.1 Making our website more usable and useful

This is especially important in the context of circumvention. Chris had some suggestions, and there are plenty more suggestions where those came from. We also have a big pile of neat stuff on our wiki (including the faq), but nobody will find it there. Coordinating the translators comes into this somewhere. One of the things I keep noticing is that we have screenshots for things like Tor Browser Bundle, but the Farsi TBB page still points to English screenshots.

(iFree should do some of this, especially the parts that matter to them, through their own funding.)

(Unfunded: Even if iFree picks up a lot of the website reworking, we need to get the common knowledge from inside Roger's head to the people doing the rework. So we should allocate some funding for this. 25k would make a great start.)

8.2 Making the website more available in blocked countries

This involves keeping up with the steps on the "finding tor" page: https://www.torproject.org/finding-tor and we plan to be working on some technical solutions such as the email autoresponder, but it also involves taking a more community-oriented approach, e.g. setting up mirrors only known by certain groups. More generally, we should coordinate the community of people running our mirrors – make them feel appreciated, help answer their questions, etc. We have some volunteers[15] already doing that a bit but it sure isn't as much attention as it could be. We also need to figure out how to make better use of mirrors inside censored areas, e.g. http://tor.anonymity.cn/

[15] https://www.torproject.org/mirrors

What ways would people who can't reach the Tor website like to get Tor? We've talked of putting Tor Browser Bundle on USB sticks and distributing them manually. We're working on an email auto responder. We also have an IRC auto responder, but we're concerned that no blocked users know what IRC is. What are they used to using?

(iFree should tackle this through their own funding.)

8.3 Legal advocacy for anonymity

Tor relies upon volunteers running relays, so advocacy that can improve the legal climate for those volunteers strengthens the diversity and capacity of the network. Several countries, mainly in the EU but soon in the US too, are pondering new laws about "data retention". The goal is to record "traffic headers", in the hopes that when something bad happens they'll have all the data just waiting to be pieced together. The reality is that this is just another huge database waiting to be leaked or broken into, while at the same time the bad guys are already using enough stepping stones and decoys that they won't be found. "The trail will lead back to my grandmother's computer, and then what?"

(Unfunded. Ultimately if we ignore this it will impact our ability to get relays, in ways we can't easily predict right now.)

8.4 Training the trainers

All around the world there are people teaching other people how to safely use Tor and related applications. This training will be ramping up with projects like iFree, NGO-in-a-box, and the Global Voices seminars.

We should help train the trainers about Tor, so they better understand the technology, issues, and tradeoffs and can then do a better job of training the users. **(iFree: Year1.0=20k, Year1.5=15k, Year2.0=15k, Year2.5=20k, Year3.0=30k. Led by Roger and Jake.)**

8.5 Teaching journalists about Tor

Figuring out how to phrase all this for the media, so they understand Tor. Crafting a message that the media can understand is a critical piece of this, especially because of how many different angles Tor has. This isn't so much about getting good press about Tor as it is about preparing journalists so if they see bad press and consider spreading it further, they'll stop and think "hey, our guy down the hall got hassled less in Thailand because of Tor." While crafting Tor's image in the media is clearly up to Tor to do, figuring out how it fits in the bigger picture of privacy and circumvention is something we should all do together.

(iFree should work on this with their funding.)

8.6 Teaching other human rights groups about Tor

Groups like Human Rights Watch, HRIC, and many others really need to understand how Tor works so they can recognize situations in which it is smart to recommend it, and situations in which it is not so smart. More generally, we'd like to give them a better intuition about the limitations of the various circumvention and/or anonymity models out there. Hopefully with this knowledge they'll be better prepared to deal with the next snake oil circumvention tool they run across.

This goal means we need to teach them, or write clear documents that teach them, or teach other people who teach them, or something like that.

(Unfunded, but could use 10k-50k to get Tor people and documents in the right places to teach people.)

References

[1] Adam Back, Ulf Möller, and Anton Stiglic. Traffic analysis attacks and trade-offs in anonymity providing systems. In Ira S. Moskowitz, editor, *Information Hiding (IH 2001)*, pages 245–257. Springer-Verlag, LNCS 2137, 2001.

[2] Kevin Bauer, Damon McCoy, Dirk Grunwald, Tadayoshi Kohno, and Douglas Sicker. Low-resource routing attacks against tor. In *Proceedings of the Workshop on Privacy in the Electronic Society (WPES 2007)*, Washington, DC, USA, October 2007.

[3] George Danezis and Paul Syverson. Bridging and fingerprinting: Epistemic attacks on route selection. In Nikita Borisov and Ian Goldberg, editors, *Proceedings of the Eighth International Symposium on Privacy Enhancing Technologies (PETS 2008)*, pages 133–150, Leuven, Belgium, July 2008. Springer.

[4] Peter C. Johnson, Apu Kapadia, Patrick P. Tsang, and Sean W. Smith. Nymble: Anonymous IP-address blocking. In *Privacy Enhancing Technologies (PET 2007)*. Springer-Verlag, LNCS 4776, 2007.

[5] Jon McLachlan and Nicholas Hopper. Don't clog the queue: Circuit clogging and mitigation in P2P anonymity schemes. In *Proceedings of Financial Cryptography (FC '08)*, January 2008.

[6] Steven J. Murdoch and George Danezis. Low-cost traffic analysis of Tor. In *IEEE Symposium on Security and Privacy*. IEEE CS, May 2005.

[7] Steven J. Murdoch and Robert N. M. Watson. Metrics for security and performance in low-latency anonymity networks. In Nikita Borisov and Ian Goldberg, editors, *Proceedings of the Eighth International Symposium on Privacy Enhancing Technologies (PETS 2008)*, pages 115–132, Leuven, Belgium, July 2008. Springer.

[8] Robin Snader and Nikita Borisov. A tune-up for Tor: Improving security and performance in the Tor network. In *Proceedings of the Network and Distributed Security Symposium - NDSS '08*. Internet Society, February 2008.

Tor Development Roadmap:
Three years to a more stable and scalable circumvention network

Roger Dingledine

1 Introduction

This document is a brainstorming draft to describe Tor development items that should happen on a three-year timeframe to move Tor forward at being a useful circumvention tool.

There are two goals to this project. First, we want to make sure to continue adapting Tor to changing censorship environments so it can continue to be a useful circumvention tool during the contract period — we want our partners in the proposal to be able to deploy it to users on the ground and have it work. Second, we want to tackle some of the long-term issues that have been holding Tor back from being useful to a broad set of people — issues that will require some investment now, but will ultimately pay off in creating a more sustainable and automated circumvention network. I tried to include the right mix between these near-term-useful items and the items that will need several years of design and analysis before they can become useful.

I've labelled each item "High Priority", meaning we should try to fit it into the project somehow because it is an important piece to the above two goals; "Medium Priority", meaning it would still be very useful to solving the goals; "Low Priority", meaning it would be useful to do but probably won't fit into the budget we have; or "Optional", meaning it was useful to write down and we should keep it mind as we plan the rest of the development work. Items without budget estimates are probably best considered Optional at this point.

The proposed projects fall into four categories. The first (largest) category is performance and robustness improvements: load balancing so we can use the available Tor relays in a way that keeps all the traffic moving quickly, making Tor relays work on Windows without crashing, encouraging more users to become relays or bridges, some preliminary scalability work, and work to continue reducing the overhead of directory information. The second category is more circumvention features, focusing on the critical gaps in the current design. The third category is client safety, that is, steps to make sure that educated and prepared users can possibly be secure while using Tor. The last category is then client usability: how to make it easier for more ordinary users to still have safety while using Tor, and how to make it more convenient for them to set it up.

2 Improve performance through better congestion control and better packet loss tolerance: $165k of High priority tasks

2.1 Improve Path Selection and Load Balancing

There are various methods for attempting to choose paths or relays intelligently and to attempt to balance the load of the Tor network. Google Summer of Code student Johannes Renner extended the TorFlow library in 2007 to help evaluate performance and anonymity implications of some of these methods. We need to decide if the anonymity implications are acceptable, and if so integrate his work into the actual Tor client. Additionally, some research needs to be done on integrating load balancing feedback with path selection. Johannes did some initial work here, and Nikita Borisov of University of Illinois published a paper at NDSS 2008 in this direction. Most recently, Steven Murdoch will publish a paper at PETS 2008 in Leuven refuting the load balancing aspect of Nikita's design.

We shouldn't expect immediate results on this, because it first requires more measurements and more analysis. We could make some great progress on this in the next year or two though, and since this is one of the critical next steps in making Tor scale better, it would be great to work more on it.

There are several steps here. First we need to examine all these proposed algorithms more closely, design new ones as necessary, and simulate the effects of the one(s) that seem most promising. One example here is looking at 2-hop paths vs 3-hop paths, and comparing the usability, performance, and average network load benefits against the potential for decreased anonymity. Another example is looking at choosing the middle hop of the path based on latencies, or geographical or network locality, to again trade off performance for anonymity. The goal is to identify strategies where we can make a small compromise in anonymity for a large performance gain. Part of the challenge here is to understand what sort of anonymity metrics we should use, so they both reflect reality and are practical to compute. Good load balancing improvements could double or more the average throughput of Tor users — this is because the variances that Tor clients are seeing right now are so high, perhaps because a small number of nodes are extremely overloaded. **($75k, High priority.)**

At the same time we should build an automated infrastructure to measure and track performance (both bandwidth and latency) in the network over time, so we can observe trends and see what effects our modified algorithms are producing in the network once we deploy them. **($20k design/setup and $20k operation, High priority.)**

Another design component we need to consider while we're working on these load balancing algorithms is susceptibility to attack. That is, right now Tor relays self-advertise whatever bandwidth they want. This has led to a variety of attack papers where an attacker signs up an allegedly high-resource node and attracts a lot more traffic than he otherwise could have handled. So while we design load balancing schemes above, we should aim for ones where the weight for each node is measured rather than just declared – this could be accomplished for example via community consensus or via a threshold of authorities.

The final step would involve implementing and deploying new better load balancing algorithms that improve (or at least don't hinder) both performance and anonymity, perhaps taking into account relay bandwidth, latency between relays, countries or continents of relays, etc. **($50k, High priority.)**

2.2 Tor over UDP, UDP over Tor

Moving to using UDP transport in Tor will provide huge advantages to performance, since user connections will do end-to-end congestion control and we should be able to fit many more connections onto a Tor network with a given capacity, and since we will tolerate dropped packets without slowing down every stream over the connection that dropped the packet. Moving to UDP transport in Tor will also provide scalability advantages, because each Tor relay doesn't need to hold open a TCP socket to each other Tor relay – meaning we can increase the network capacity, and thus again increase performance. Further, more relays (specifically, more diversity of relays) leads to better anonymity for users.

Another advantage of moving to UDP transport is that we Tor would now be able to handle connections for UDP-based applications like Skype (VoIP in general), OpenVPN, DNS, etc.

We've been working with Ian Goldberg at Waterloo. He was the Chief Scientist at Zero-Knowledge Systems, a company that deployed a UDP-based anonymity system called the Freedom network that was quite like Tor. He has several grad students who want to work on exactly this problem. We are also working with Camilo Viecco at Indiana University, a grad student working on this for his PhD thesis.

However, the academic goals in both cases are currently just to write a paper and move on – they aren't actually going to do a full design, or even assess whether the full design will improve things or how much.

The first step would be to rederive how the Freedom network worked, combine that with ideas from the above two research groups, and produce a design and specification for how to pass Tor traffic over UDP. This step involves adding sequence numbers and MACs to each packet as it traverses the Tor network, handling and retrying dropped packets during the circuit-level crypto handshake, etc. The second step would be to analyze the design with respect to Tor's current security properties, including perfect forward secrecy from the current circuit handshake, not partitioning traffic across multiple connections, etc. Step three is to figure out a migration plan that allows us to move to the new design within a year or so, and doesn't harm user

security or performance too much during migration. (For example, some designs we've seen involve a "flag day" where all users and servers stop using one network and start using a different one.) Then we iterate steps one, two, and three until we get a realistic design that still has adequate security properties. **(Design and papers and review. $75k, Medium priority.)**

However, there's still a big gap: since we're transporting TCP and UDP packets end-to-end and just writing them onto the network on each side, the details of the TCP stack used on the client side becomes relevant. Operating systems like Windows, OS X, and Linux choose sequence numbers, source ports, and other connection properties in a predictable way, meaning the exit relay, the destination site, or somebody in between can observe the traffic and discover that two connections are coming from the same user. This attack probably works across different exit nodes, and probably works across time (e.g. users with a given timestamp skew will probably retain it later too). See https://wiki.torproject.org/noreply/TheOnionRouter/TorFAQ#PhysicalFingerprint

So step four is to find, adapt, and/or write a user-space TCP stack that can rewrite and normalize TCP and UDP packets and streams so they no longer contain these identifying properties. This is a huge task. **(User-space TCP stack. $200k-500k?, Low priority.)**

At this point, we will have a convincing design for how to migrate to UDP for connections between Tor relays, and for Tor clients that are in a position to use UDP. But clients in censored areas may still not be able to use our UDP design for their first hop, because it will have an unusual network footprint. These users would still benefit from most of the above advantanges (improved performance inside the Tor network, improved scalability and thus improved anonymity), but they wouldn't get what is perhaps the most important benefit for them, which is tolerating high packet loss to their first hop.

So step five is to reverse engineer Skype, or pick a different popular UDP-based app that is allowed through most firewalls, figure out what security properties it's missing (for example, I bet its crypto handshake doesn't provide perfect forward secrecy), and then either figure out how to achieve our security properties while looking like Skype traffic, or decide to have a second inferior handshake that would provide less security to these users and also partition them from the rest of the Tor user base. Then we iterate steps one-three above until our new protocol seems like the right one to deploy. **(Design and papers and review. $75k, Low priority.)**

Step six is then to implement, deploy, and manage the migration. **(Implement and deploy. $100k, Low priority.)**

2.3 More robust connection protocol for the first hop

The above UDP design is probably more involved than we want to deal with for this project. The best subset of it to focus on is dealing with flaky connections from the Tor client to the Tor network. If these links drop many packets, then Tor will appear slow even if the rest of the circuit through the Tor network is fast. Therefore we could implement an alternate transport mechanism just for this first hop.

TCP is designed for reliable connections: it assumes that packet loss happens because of network congestion. In reality, flaky connections can occur because of static on the local wires or any number of other problems. Alternate transport protocols such as the Airhook protocol are designed for wireless connections where packet loss is not simply a function of traffic congestion.

To deploy this approach, we would need to teach both Tor clients and also Tor relays and/or Tor bridges about the new protocol that we choose. We would also need to advertise support in the server descriptor, along with details such as which version they support.

($25k for investigation and analysis, $50k for deployment and integration, $75k if we want the look-like-Skype feature from above. Medium priority.)

3 Improve performance through more capacity: $370k of High priority tasks

Better performance comes from increased network capacity, and better security comes from increased network diversity.

3.1 Relay stability on Windows

Tor relays still don't work well or reliably on Windows XP or Windows Vista, because we don't use the Windows-native "overlapped IO" approach. Christian King made a good start at teaching libevent about overlapped IO during Google Summer of Code 2007, and the next steps are to A) finish that work, B) teach Tor to do OpenSSL calls on buffers rather than interacting directly with the network, and C) teach Tor to use the new libevent buffers approach. **($45k to finish the above steps, $45k to make it actually work smoothly. High priority.)**

3.2 Tor clients that find themselves reachable and reliable should automatically become a bridge or relay.

We've made a lot of progress towards letting an ordinary Tor client also serve as a Tor relay, and we will continue to make progress as we move forward. This item would be first to finish off and integrate all the required items that we are currently working on or plan to get to soon: upnp with Vidalia integration, clients with very wrong clocks, etc. There are several more topics that need investigation still:

1) Better algorithms for giving priority to local traffic. Proposal 111 made a lot of progress at separating local traffic from relayed traffic, so Tor users can rate limit the relayed traffic at a stricter level. But since we want to pass both traffic classes over the same TCP connection, we can't keep them entirely separate. The current compromise is that we treat all bytes to/from a given connectin as local traffic if any of the bytes within the past N seconds were local bytes. But a) we could use some more intelligent heuristics, and b) this leaks information to an active attacker about when local traffic was sent/received. **(Medium priority.)**

2) First a bridge, then a public relay? Once enough of the items in this section are done, I want all clients to start out automatically detecting their reachability and opting to be bridge relays. Then if they realize they have enough consistency and bandwidth, they should automatically upgrade to being non-exit relays. **($30k, but see item 3 next. High priority.)**

3) Understand the risks from letting the attacker send traffic through your relay while you're also initiating your own anonymized traffic. Three different research papers [1, 3, 4] describe ways to identify the nodes in a circuit by running traffic through candidate nodes and looking for dips in the traffic while the circuit is active. These clogging attacks are not that scary in the Tor context so long as relays are never clients too. But if we're trying to encourage more clients to turn on relay functionality too (whether as bridge relays or as normal relays), then we need to understand this threat better and learn how to mitigate it.

One promising research direction is to investigate the RelayBandwidthRate feature that lets Tor rate limit relayed traffic differently from local traffic. Since the attacker's "clogging" traffic is not in the same bandwidth class as the traffic initiated by the user, it may be harder to detect interference. Or it may not be.

We aren't really comfortable setting users up en masse as bridges or relays until we understand these issues more.

($100k, High priority.)

3.3 Incentives design

Roger has been working with researchers at Rice University to simulate and analyze a new design where the directory authorities assign gold stars to well-behaving relays, and then all the relays give priority to traffic from gold-starred relays. The great feature of the design is that not only does it provide the (explicit) incentive to run a relay, but it also aims to grow the overall capacity of the network, so even non-relays will benefit.

However, the current incentives design we invented has a serious flaw, which is that the set of gold-starred relays is known to the adversary, and over time he can narrow down which gold-star users are always the ones online when a certain activity (e.g. posting to a blog) happens. We need to revamp the design so the set of high-priority users and the set of currently online relays is less clearly related.

($50k-100k, Medium priority.)

3.4 Continue research on how to splinter the network as it grows so we can maintain a good balance of both anonymity and scalability.

This topic is probably the hardest open research problem in the field right now. We need to enumerate and analyze the various solutions we've come up with already, and work on new better solutions.

Step one is to specify the details for the simple version: partition the networkstatus documents as they get too large, and have clients fetch and use only a single partition, and have mirrors only cache descriptors from within their partition. Analyze the scalability, performance, and anonymity properties therein. **($25k for initial analysis, another $25k to have a design ready to implement when needed. High priority.)**

Step two is to research variants of this design that "blend" multiple partitions together. [more detail here] **($100k, Medium priority.)**

3.5 Clients download less directory info. Especially useful for clients on modems.

See "piece one" in https://www.torproject.org/svn/trunk/doc/spec/proposals/ideas/xxx-grand-scaling-plan.txt

The challenge here is that many of the design decisions for this topic impact other scalability and partitioning decisions down the road: that is, what we do here will decide what options we have for all the other designs. So we need to think very carefully

($50k for analysis of the various designs; then $50k to figure out a transition plan and deploy, if we conclude it's a good idea. High priority.)

4 Suitability for circumvention: $230k of High priority tasks

4.1 Distributed bridge authorities

Make our 'single bridge authority' design into a 'redundant bridge authorities' design, so bridge publish/lookup can't be knocked over or broken into by attacking a single location.

4.2 Normalize our network fingerprint even more

Play the TLS handshake arms race as needed. We also need to investigate how well our "get people to offer their ORPort on 443" strategy is working. Another research item to tackle is whether our fixed cell size of 512 bytes makes us stand out on the wire, and if there are any light padding approaches that can blur the pattern. [other items pending once Nick writes them down]

($25k for periodic adjustments as Smartfilter and Websense do their thing. High priority.)

4.3 More bridge address distribution strategies

Deploy more bridge address distribution strategies, based on a broader set of technologies like SMS, radio, WoW, etc. Many of these approaches will require more manual ongoing attention than our first few approaches.

More strategies are worth working on, but I think we should focus first on making everything smooth for the ones we've got.

4.4 Guard nodes for the bridges

Decide whether bridge users need to choose a second "layer" of entry guards, so it's harder for an ordinary Tor middle server to enumerate bridge relays just by seeing who connects. Start solving this problem somehow, for example by making bridge users do the above.

I think this is going to be necessary in the near term, since if it turns out to be a real attack, it is a very practical one. **($10k for analysis, $40k for deployment. High priority.)**

4.5 Tracking bridge reachability

Better and more automated measurement tools for whether bridges are actually up, and actually reachable from inside target countries.

"Actually up" is quite straightforward: we already do simple reachability testing from the bridge authority. Tracking reachability from inside target countries will be a statistical game based on how many geoip details we can collect from the bridges themselves. See the "better user metrics" item below.

4.6 Email auto-responder

Email auto-responder so for example gmail users can fetch the Tor software via email. Social network distribution techniques. Continue beating on this problem.

($5k design and setup, $10k operation. High priority.)

4.7 Research: scanning-resistance

4.8 Research: hiding whether the user is reading or publishing?

4.9 Research: how many bridges do you need to know to maintain reachability?

We need to track the churn of bridges over time and then analyze how many bridges are smart to know, or how often it is smart to learn new bridges, in order to stay connected. If a few bridges are likely to last a long time, we can focus on other problems. On the other hand, if even a half dozen bridges are likely to all vanish soon, we need to work both on encouraging bridge stability and on better algorithms for learning about new bridges.

Then we need to watch for trends and changes over time to see if our job is getting easier or harder.

($5k to assess what and how to store data, $15k for operation and analysis. High priority.)

4.10 Better user metrics and measurements

The proposal wants us to demonstrate that we'll be able to measure progress. There are a number of ways to do this, but most of them have anonymity implications so they must be approached carefully.

Step one is to automate the collection of current network traffic statistics: volume of data over time, number of available relays each day, total advertised capacity for the day, etc. We've been collecting some of this data already so we can have some baselines, but we need to collect it in a more reliable fashion (i.e. pay somebody to be sure to do it), and we need to redesign and rewrite our scripts for processing it into useful graphs and figures since we're talking dozens of gigabytes of input data. **($5k redesign; $20k operation. High priority.)**

Step two is to clean up and maintain the "exitlist" service we are aiming to offer, so it's easy for other sites to check if a given IP address was a Tor exit relay at a given point. This will let websites measure how many of their users are coming from Tor. We will also need to provide some glue, and maybe training, to let people convert apache logs into useful information about the fraction (and variety?) of Tor users. To be most effective, we will want to answer not just "is this IP address a Tor exit relay now?", but also "was it an exit relay in the recent past?" **($10k implementation, $10k operation. High priority.)**

Step three is to implement the "geoip lookup" designs that Nick is designing in Q2 08.. These should give us a better estimate of the overall set of current Tor users; and as we migrate to the directory guards design, it should still be able to give us some rough numbers. **($10k deployment. High priority.)**

Step four is to try to handle the skew in our user geoip stats from dynamic IP addresses. Currently we're forced to just measure a rolling 24 hour window, since most users in Germany and China get a new IP address daily. Thus our metrics leave out most of the people who use Tor only infrequently – which could be a big part of our target population. Can we find or generate a database of Internet address blocks that are associated with frequent cycling? If so, we could discard the sightings of those users after 24 hours,

while accumulating the other sightings over a longer time period. **($5k research, $10k integration and operation. High priority.)**

Step five is to investigate whether we can safely switch the current website logs over to doing a GeoIP lookup before discarding the IP address. Being able to track downloads or page views over time is not a perfect metric, because it doesn't include downloads from other sites (e.g. Debian repositories) or downloads shared among friends, but it is still a metric. Do we need to randomize or otherwise lock down the logs to make this increased data gathering acceptably safe? **($5k investigation and operation. High priority.)**

Step six is to look into whether entry guards and/or directory mirrors should collect per-country byte information in addition to their current GeoIP stat collection. If we publish it in per-day chunks, rather than per-15-minute chunks, it seems less dangerous. But we still need to carefully consider edge cases where there aren't enough users from a given location on a given day to provide cover. Currently, we can estimate the number of bytes users from a given country use by looking at the "total volume of Tor traffic" from step one and the "percentage of Tor users from that country' from step three. But some countries have better net connections than others; this approach would allow us to collect more accurate usage stats. **($20k research, $5k deployment and operation. High priority?)**

Step seven is to do some analysis about how to detect country-wide blocking events based on trends in the above data. For example, if the website sees a sharp drop-off in hits from a given country, perhaps the site is blocked by that country's firewall or ISPs. Similarly, a sharp drop-off in reported GeoIP stats from bridges or directory guards could show that the country is starting to block access to bridges and/or block Tor's network signature. The challenge here is that when there are very few data points, it's hard to learn something statistically significant from losing a few. Is there a way to aggregate the overall data points in a way that makes it easier to notice changes? **($20k research. High priority.)**

Step eight is to consider whether we can learn about the countries of Tor's users by looking at exit destinations. For example, if half the websites visited by Tor are Chinese-language, based on Hong Kong, etc, then we have learned something. This also presents a metric that can be tracked over time to find trends. We could learn this by integrating our geoip module into exit destinations and tracking and reporting those in a similar way to our current client statistics; we would want to investigate the anonymity properties for this item even more carefully, and it may turn out to be a really bad idea. **($20k research, Medium priority.)**

5 Client safety: $325k of High priority tasks

5.1 Automatic update

Tor currently has no mechanism for updating clients in the event of security vulnerabilities and changes to blocking mechanisms. To be effective in the arms race, we will need one.

Step one is to work on auto-update-of-Tor features inside Vidalia:

- looking at the majority-signed networkstatus consensus to decide when to update and to what version (Tor already lists what versions are considered safe, in each networkstatus document),
- doing the update either via Tor or via the directory mirror update protocol (proposal 127) when possible, for additional privacy,
- checking package signatures,
- giving the user an interface for these updates, including letting her opt to migrate from one major Tor version to the next.

This should work on both Windows and OS X. Ideally we would adapt some third-party lib to do parts of this for us – but we haven't found any good free-software security-oriented auto-update lib out there yet. Then we need to work on auto-update for Vidalia itself too, as well as for other supporting applications like

Polipo. We also need to produce better plans around security issues like signing key rotation so we can be sure to provide safe and secure service over time. **($50k. High priority.)**

Then we should take a step back and revise our design into a thin client that's separate from Vidalia. It should know how to fetch new versions of the components it wants and check their signatures appropriately. We could imagine this approach as a tiny bundle stub that bootstraps itself into a full Tor bundle; it would also replace our current stopgap workaround for modem users who can't fetch the entire Tor Browser Bundle without losing their connection. We would still want to ship the full bundle too for folks who want to get the whole set and carry it around with them, though. **($20k. High priority.)**

Then we need to look at how Firefox's automatic update scheme works with our automatic update scheme in the context of the Tor Browser Bundle. What about other FF extensions inside the bundle? Which app(s) should be the one managing the automatic update? Does Firefox do anything smarter than check the SSL certificate of the site that it updates from? **($30k. High priority.)**

5.2 Vidalia integration for PlaintextPorts warnings and other status events

Vidalia should walk the user through recognizing that connections to plaintext ports (e.g. 110, 143) probably indicate a bad move on the user's part.

More generally, there are a variety of status events that Tor currently sends to Vidalia (e.g. for reachability testing as a server), but Vidalia has no way to present them usefully to the user. We should come up with a generalized way to interact with the user when errors or warnings occur.

Look into ways for Vidalia and Torbutton to communicate. For example, some link seems necessary for the "new identity" button to function properly when both Firefox and Polipo are determined to do keepalive on their current connections.

($50k. High priority.)

5.3 Node/Network scanning

We've written a prototype node scanner called 'SoaT' to scan the Tor network for malfunctioning and malicious nodes; we presented it at Black Hat and Defcon 2007. Currently, it scans for malicious content injection at the exit relays by checking MD5 sums of documents. It would be nice to produce a more flexible fingerprint of a page, perhaps based on the javascript/object/image content only. Additionally, it would be nice to integrate scan results into the Tor directory consensus, so clients can use the information in their routing decisions to avoid malicious or failing nodes. Passive scanning and reliability reporting from clients and nodes to the directory servers is also a possibility, but some of this may need experimentation and research. http://fscked.org/transient/SecuringTheTorNetwork.pdf

There are several components here. The big first steps are A) a more automated scanning mechanism, B) coming up with plausible tests that are hard to distinguish from "normal" fetches, and C) integrating the results into the directory authorities so Tor clients get quick feedback about problems we discover.

The first cut at this tool would focus on making it easy to configure and run on a variety of platforms, giving it a good suite of plausible-looking tests, and feeding the results into a directory authority. **($40k. High priority.)**

Then we should make the tool more automated and easier to run for long periods unattended, including ways for users to submit tests directly to the database ("I'm having problems with the following page or website, please add it to the test suite") and other ways to add the client reliability reporting described above. **($35k. High priority.)**

5.4 Torbutton development

We need to stabilize and then continue the "Torbutton-dev" work so we can give people a robust Firefox extension that not only lets them toggle Tor on/off, but also protects them from many application-level threats on the web.

We've made a huge amount of progress here: https://torbutton.torproject.org/dev/design/

But many more issues remain — in particular, while we've been focusing lately on making sure that Torbutton *can* provide safe browsing, it probably is at odds with usability and useful browsing in many ways. We will need to work with users to find the right balance between safety and usefulness.

Further, Torbutton will need constant attention to maintain the current features as Firefox's internals change. We have filed several dozen Firefox bugs, and are working with Mozilla to resolve them. Alas, there will continue to be more.

($50k further work, $50k maintenance. High priority.)

6 Client usability: $220k of High priority tasks

6.1 Transparently intercepting connections

Currently, Tor clients need to actively configure their applications to use Tor as a SOCKS proxy. This step results in many confused users, as well as (probably) many users who are using Tor in an unsafe configuration. Even for web browsing, many of the challenges that Torbutton attempts to solve come from ways that websites can trick users into bypassing their proxy settings. And simply disabling all of these avenues results in usability problems, e.g., preventing users from watching Flash videos at Youtube.

Some tools like Xerobank VM and JanusVM use a virtual machine (generally VMWare running a Linux OS) to run the Tor client and web proxy, and they use the VM's ability to intercept outgoing connections so they can redirect them into Tor.

It might be that the VM approach is necessary, in which case we should work on adapting QEMU so our bundles are not subject to VMWare's restrictive redistribution licenses. Or it might be that we can just reuse drivers like the ones OpenVPN uses to transparently intercept the outgoing connections and pass them into the Tor client.

We need to research the options and implications, and document the problems that can come up. For example, trying to pass outgoing email connections into Tor will not work very well, since few Tor relays have an exit policy that will allow outgoing email. We may end up needing an administrative interface for which addresses and ports should be sent into Tor; but we'd like to find a more usable solution than that.

($50k for research and analysis. $50k for initial deployment. High priority.)

6.2 Tor Browser Bundle

Lock down the Tor Browser Bundle, make it more robust, etc. Deal with deployment issues, updating features and maintenance for new versions of the component software, translations, etc. **($20k development, $30k maintenance. High priority.)**

6.3 LiveCD

We need a nice bootable LiveCD containing a minimal OS and a few applications configured to use it correctly. The Anonym.OS project has demonstrated that this is quite feasible, and there is a nice student in Sweden who is currently trying to maintain it for free.

It needs more documentation, more analysis about whether its configuration choices are the right ones, and more thought for what applications to include so it has enough that it's useful — and what apps to *exclude* to keep its complexity from dragging it down.

Most appealingly, Anonym.OS has recently added a feature allowing it to be booted from inside Windows as a standalone system that provides a set of self-contained correctly-configured applications chosen for their good security. This approach could provide the best combination of the transparently intercepted connections item above with the Tor Browser Bundle item. The tradeoff is download size. **($25k analysis, $50k maintenance. Medium priority.)**

6.4 Playing nicely with websites

Right now Wikipedia and some other services block posts from Tor users, due to a few abusers. Tools like Nymble[2] allows these services to recognize the abusers later without ever needing to learn who or where they are – thus allowing the non-abusers to start using the service like usual again.

The first step here is to write down a clear explanation of why anonymity is not at odds with open access to sites like Wikipedia and Slashdot. Roger gave a talk at Wikimania 2006 to explain to the Mediawiki developers about other options they have besides blacklisting IP addresses[1]. Once we have this essay ready, we can start to educate other people in the community about the fact that practical options do exist. The Wikipedia community is diverse though, and can be stubborn, so this will not be accomplished quickly. At the same time we need to work with the legitimate Tor users (e.g. the ones who would like to edit Wikipedia through Tor) and help them to adopt some interim solutions, if any. **($15k advocacy, $15k operations. High priority.)**

The next step is to help deploy a system like Nymble, which is an infrastructure for letting websites blacklist abusive users without needing to (or being able to) unveil their location or identity. They are apparently working on an implementation right now, but it will definitely need help with deployment, usability, and sustainability. **(Medium priority.)**

6.5 IPv6 for clients and destinations

Support clients that have only IPv6 addresses (we're told that a lot of OLPC recipients will be using solely IPv6), and support exit destinations that have only IPv6 addresses.

6.6 Let users specify their exit country

As the network grows, and censorship gets more varied and widespread, exiting from a censored country becomes more of a hassle.

The first steps are a) changing the Vidalia interface to let you communicate which country you want to exit from, b) turning countries into sets of Tor servers that we're pretty sure are in those countries, and c) making sure that the Tor client does in fact do the right thing with the set of preferred servers (the basic features have been in for a while, but nobody uses them for much and they likely have some problems). **($20k. High priority.)**

The next steps are to tackle making the Tor client scale well when you specify really large lists of server IDs either in the "use these" or the "don't use these" side, work on more complex interface approaches ("use these two countries but not this one"), specify other constraints the user has in mind ("please only pick servers with certain uptime, certain bandwidth, certain operating system, etc"), look into the anonymity issues with choosing a smaller set of options at each step, look into the security questions of using geoip data vs whois data for country codes, let us build a more efficient interface (that doesn't require interacting with the GUI) by specifying country properties in the url (like www.google.com.cn.exit) without actually broadcasting this url to the destination website in the Host: http header, and figure out various other issues that come up as we start deploying a solution. **($60k. Medium Priority.)**

6.7 Tutorials on safe usage

General tutorials on what common applications are Tor-friendly, and how to configure things safely.

6.8 Outreach and training the trainers

We should train the trainers so they better understand the technology, issues, and tradeoffs and can then do a better job of training the users. **($20k for travel and preparation. High Priority.)**

[1] `http://freehaven.net/~arma/slides-wiki-tor.pdf`

References

[1] Adam Back, Ulf Möller, and Anton Stiglic. Traffic analysis attacks and trade-offs in anonymity providing systems. In Ira S. Moskowitz, editor, *Information Hiding (IH 2001)*, pages 245–257. Springer-Verlag, LNCS 2137, 2001.

[2] Peter C. Johnson, Apu Kapadia, Patrick P. Tsang, and Sean W. Smith. Nymble: Anonymous IP-address blocking. In *Privacy Enhancing Technologies (PET 2007)*. Springer-Verlag, LNCS 4776, 2007.

[3] Jon McLachlan and Nicholas Hopper. Don't clog the queue: Circuit clogging and mitigation in P2P anonymity schemes. In *Proceedings of Financial Cryptography (FC '08)*, January 2008.

[4] Steven J. Murdoch and George Danezis. Low-cost traffic analysis of Tor. In *IEEE Symposium on Security and Privacy*. IEEE CS, May 2005.

Tor Development Roadmap: Wishlist for Nov 2006–Dec 2007

Roger Dingledine Nick Mathewson Shava Nerad

November 11, 2006

1 Introduction

Tor (the software) and Tor (the overall software/network/support/document suite) are now experiencing all the crises of success. Over the next year, we're probably going to grow more in terms of users, developers, and funding than before. This gives us the opportunity to perform long-neglected maintenance tasks.

2 Code and design infrastructure

2.1 Protocol revision

To maintain backward compatibility, we've postponed major protocol changes and redesigns for a long time. Because of this, there are a number of sensible revisions we've been putting off until we could deploy several of them at once. To do each of these, we first need to discuss design alternatives with other cryptographers and outside collaborators to make sure that our choices are secure.

First of all, our protocol needs better **versioning support** so that we can make backward-incompatible changes to our core protocol. There are difficult anonymity issues here, since many naive designs would make it easy to tell clients apart (and then track them) based on their supported versions.

With protocol versioning support would come the ability to **future-proof our ciphersuites**. For example, not only our OR protocol, but also our directory protocol, is pretty firmly tied to the SHA-1 hash function, which though not yet known to be insecure for our purposes, has begun to show its age. We should remove assumptions throughout our design based on the assumption that public keys, secret keys, or digests will remain any particular size indefinitely.

Our OR **authentication protocol**, though provably secure[4], relies more on particular aspects of RSA and our implementation thereof than we had initially believed. To future-proof against changes, we should replace it with a less delicate approach.

(For all the above: 2 person-months to specify, spread over several months with time for interaction with external participants. One person-month to implement. Start specifying in early 2007.)

We might design a **stream migration** feature so that streams tunneled over Tor could be more resilient to dropped connections and changed IPs. **(Not in 2007.)**

A new protocol could support **multiple cell sizes**. Right now, all data passes through the Tor network divided into 512-byte cells. This is efficient for high-bandwidth protocols, but inefficient for protocols like SSH or AIM that send information in small chunks. Of course, we need to investigate the extent to which multiple sizes could make it easier for an adversary to fingerprint a traffic pattern. **(Not in 2007.)**

As a part of our design, we should investigate possible **cipher modes** other than counter mode. For example, a mode with built-in integrity checking, error propagation, and random access could simplify our protocol significantly. Sadly, many of these are patented and unavailable for us. **(Not in 2007.)**

2.2 Scalability

2.2.1 Improved directory efficiency

Right now, clients download a statement of the **network status** made by each directory authority. We could reduce network bandwidth significantly by having the authorities jointly sign a statement reflecting their vote on the current network status. This would save clients up to 160K per hour, and make their view of the network more uniform. Of course, we'd need to make sure the voting process was secure and resilient to failures in the network. **(Must do; specify in 2006. 2 weeks to specify, 3-4 weeks to implement.)**

We should **shorten router descriptors**, since the current format includes a great deal of information that's only of interest to the directory authorities, and not of interest to clients. We can do this by having each router upload a short-form and a long-form signed descriptor, and having clients download only the short form. Even a naive version of this would save about 40% of the bandwidth currently spent by clients downloading descriptors. **(Must do; specify in 2006. 3-4 weeks.)**

We should **have routers upload their descriptors even less often**, so that clients do not need to download replacements every 18 hours whether any information has changed or not. (As of Tor 0.1.2.3-alpha, clients tolerate routers that don't upload often, but routers still upload at least every 18 hours to support older clients.) **(Must do, but not until 0.1.1.x is deprecated in mid 2007. 1 week.)**

2.2.2 Non-clique topology

Our current network design achieves a certain amount of its anonymity by making clients act like each other through the simple expedient of making sure that

all clients know all servers, and that any server can talk to any other server. But as the number of servers increases to serve an ever-greater number of clients, these assumptions become impractical.

At worst, if these scalability issues become troubling before a solution is found, we can design and build a solution to **split the network into multiple slices** until a better solution comes along. This is not ideal, since rather than looking like all other users from a point of view of path selection, users would "only" look like 200,000–300,000 other users. **(Not unless needed.)**

We are in the process of designing **improved schemes for network scalability**. Some approaches focus on limiting what an adversary can know about what a user knows; others focus on reducing the extent to which an adversary can exploit this knowledge. These are currently in their infancy, and will probably not be needed in 2007, but they must be designed in 2007 if they are to be deployed in 2008. **(Design in 2007; unknown difficulty. Write a paper.)**

2.2.3 Relay incentives

To support more users on the network, we need to get more servers. So far, we've relied on volunteerism to attract server operators, and so far it's served us well. But in the long run, we need to **design incentives for users to run servers** and relay traffic for others. Most obviously, we could try to build the network so that servers offered improved service for other servers, but we would need to do so without weakening anonymity and making it obvious which connections originate from users running servers. We have some preliminary designs [1, 2], but need to perform some more research to make sure they would be safe and effective. **(Write a draft paper; 2 person-months.)**

2.3 Portability

Our **Windows implementation**, though much improved, continues to lag behind Unix and Mac OS X, especially when running as a server. We hope to merge promising patches from Mike Chiussi to address this point, and bring Windows performance on par with other platforms. **(Do in 2007; 1.5 months to integrate not counting Mike's work.)**

We should have **better support for portable devices**, including modes of operation that require less RAM, and that write to disk less frequently (to avoid wearing out flash RAM). **(Optional; 2 weeks.)**

We should **stop using socketpair on Windows**; instead, we can use in-memory structures to communicate between cpuworkers and the main thread, and between connections. **(Optional; 1 week.)**

2.4 Performance: resource usage

We've been working on **using less RAM**, especially on servers. This has paid off a lot for directory caches in the 0.1.2, which in some cases are using 90% less memory than they used to require. But we can do better, especially in the area

around our buffer management algorithms, by using an approach more like the BSD and Linux kernels use instead of our current ring buffer approach. (For OR connections, we can just use queues of cell-sized chunks produced with a specialized allocator.) This could potentially save around 25 to 50% of the memory currently allocated for network buffers, and make Tor a more attractive proposition for restricted-memory environments like old computers, mobile devices, and the like. **(Do in 2007; 2-3 weeks plus one week measurement.)**

We should improve our **bandwidth limiting**. The current system has been crucial in making users willing to run servers: nobody is willing to run a server if it might use an unbounded amount of bandwidth, especially if they are charged for their usage. We can make our system better by letting users configure bandwidth limits independently for their own traffic and traffic relayed for others; and by adding write limits for users running directory servers. **(Do in 2006; 2-3 weeks.)**

On many hosts, sockets are still in short supply, and will be until we can migrate our protocol to UDP. We can **use fewer sockets** by making our self-to-self connections happen internally to the code rather than involving the operating system's socket implementation. **(Optional; 1 week.)**

2.5 Performance: network usage

We know too little about how well our current path selection algorithms actually spread traffic around the network in practice. We should **research the efficacy of our traffic allocation** and either assure ourselves that it is close enough to optimal as to need no improvement (unlikely) or **identify ways to improve network usage**, and get more users' traffic delivered faster. Performing this research will require careful thought about anonymity implications.

We should also **examine the efficacy of our congestion control algorithm**, and see whether we can improve client performance in the presence of a congested network through dynamic 'sendme' window sizes or other means. This will have anonymity implications too if we aren't careful.

(For both of the above: research, design and write a measurement tool in 2007: 1 month. See if we can interest a graduate student.)

We should work on making Tor's cell-based protocol perform better on networks with low bandwidth and high packet loss. **(Do in 2007 if we're funded to do it; 4-6 weeks.)**

2.6 Performance scenario: one Tor client, many users

We should **improve Tor's performance when a single Tor handles many clients**. Many organizations want to manage a single Tor client on their firewall for many users, rather than having each user install a separate Tor client. We haven't optimized for this scenario, and it is likely that there are some code paths in the current implementation that become inefficient when a single Tor is servicing hundreds or thousands of client connections. (Additionally, it is likely

that such clients have interesting anonymity requirements the we should investigate.) We should profile Tor under appropriate loads, identify bottlenecks, and fix them. **(Do in 2007 if we're funded to do it; 4-8 weeks.)**

2.7 Tor servers on asymmetric bandwidth

Tor should work better on servers that have asymmetric connections like cable or DSL. Because Tor has separate TCP connections between each hop, if the incoming bytes are arriving just fine and the outgoing bytes are all getting dropped on the floor, the TCP push-back mechanisms don't really transmit this information back to the incoming streams. **(Do in 2007 since related to bandwidth limiting. 3-4 weeks.)**

2.8 Running Tor as both client and server

Many performance tradeoffs and balances that might need more attention. We first need to track and fix whatever bottlenecks emerge; but we also need to invent good algorithms for prioritizing the client's traffic without starving the server's traffic too much. **(No idea; try profiling and improving things in 2007.)**

2.9 Protocol redesign for UDP

Tor has relayed only TCP traffic since its first versions, and has used TLS-over-TCP to do so. This approach has proved reliable and flexible, but in the long term we will need to allow UDP traffic on the network, and switch some or all of the network to using a UDP transport. **Supporting UDP traffic** will make Tor more suitable for protocols that require UDP, such as many VOIP protocols. **Using a UDP transport** could greatly reduce resource limitations on servers, and make the network far less interruptible by lossy connections. Either of these protocol changes would require a great deal of design work, however. We hope to be able to enlist the aid of a few talented graduate students to assist with the initial design and specification, but the actual implementation will require significant testing of different reliable transport approaches. **(Maybe do a design in 2007 if we find an interested academic. Ian or Ben L might be good partners here.)**

3 Blocking resistance

3.1 Design for blocking resistance

We have written a design document explaining our general approach to blocking resistance. We should workshop it with other experts in the field to get their ideas about how we can improve Tor's efficacy as an anti-censorship tool.

3.2 Implementation: client-side and bridges-side

Our anticensorship design calls for some nodes to act as "bridges" that are outside a national firewall, and others inside the firewall to act as pure clients. This part of the design is quite clear-cut; we're probably ready to begin implementing it. To **implement bridges**, we need to have servers publish themselves as limited-availability relays to a special bridge authority if they judge they'd make good servers. We will also need to help provide documentation for port forwarding, and an easy configuration tool for running as a bridge.

To **implement clients**, we need to provide a flexible interface to learn about bridges and to act on knowledge of bridges. We also need to teach them how to know to use bridges as their first hop, and how to fetch directory information from both classes of directory authority.

Clients also need to **use the encrypted directory variant** added in Tor 0.1.2.3-alpha. This will let them retrieve directory information over Tor once they've got their initial bridges. We may want to get the rest of the Tor user base to begin using this encrypted directory variant too, to provide cover.

Bridges will want to be able to **listen on multiple addresses and ports** if they can, to give the adversary more ports to block.

3.3 Research: anonymity implications from becoming a bridge

3.4 Implementation: bridge authority

The design here is also reasonably clear-cut: we need to run some directory authorities with a slightly modified protocol that doesn't leak the entire list of bridges. Thus users can learn up-to-date information for bridges they already know about, but they can't learn about arbitrary new bridges.

3.5 Normalizing the Tor protocol on the wire

Additionally, we should **resist content-based filters**. Though an adversary can't see what users are saying, some aspects of our protocol are easy to fingerprint *as* Tor. We should correct this where possible.

Look like Firefox; or look like nothing? Future research: investigate timing similarities with other protocols.

3.6 Access control for bridges

Design/impl: password-protecting bridges, in light of above. And/or more general access control.

3.7 Research: scanning-resistance

3.8 Research/Design/Impl: how users discover bridges

Our design anticipates an arms race between discovery methods and censors. We need to begin the infrastructure on our side quickly, preferably in a flexible language like Python, so we can adapt quickly to censorship.

phase one: personal bridges phase two: families of personal bridges phase three: more structured social network phase four: bag of tricks Research: phase five...

Integration with Psiphon, etc?

3.9 Document best practices for users

Document best practices for various activities common among blocked users (e.g. WordPress use).

3.10 Research: how to know if a bridge has been blocked?

3.11 GeoIP maintenance, and "private" user statistics

How to know if the whole idea is working?

3.12 Research: hiding whether the user is reading or publishing?

3.13 Research: how many bridges do you need to know to maintain reachability?

3.14 Resisting censorship of the Tor website, docs, and mirrors

We should take some effort to consider **initial distribution of Tor and related information** in countries where the Tor website and mirrors are censored. (Right now, most countries that block access to Tor block only the main website and leave mirrors and the network itself untouched.) Falling back on word-of-mouth is always a good last resort, but we should also take steps to make sure it's relatively easy for users to get ahold of a copy.

4 Security

4.1 Security research projects

We should investigate approaches with some promise to help Tor resist end-to-end traffic correlation attacks. It's an open research question whether (and to what extent) **mixed-latency** networks, **low-volume long-distance padding**, or other approaches can resist these attacks, which are currently some of the

most effective against careful Tor users. We should research these questions and perform simulations to identify opportunities for strengthening our design without dropping performance to unacceptable levels. **(Start doing this in 2007; write a paper. 8-16 weeks.)**

We've got some preliminary results suggesting that **a topology-aware routing algorithm** [3] could reduce Tor users' vulnerability against local or ISP-level adversaries, by ensuring that they are never in a position to watch both ends of a connection. We need to examine the effects of this approach in more detail and consider side-effects on anonymity against other kinds of adversaries. If the approach still looks promising, we should investigate ways for clients to implement it (or an approximation of it) without having to download routing tables for the whole Internet. **(Not in 2007 unless a graduate student wants to do it.)**

We should research the efficacy of **website fingerprinting** attacks, wherein an adversary tries to match the distinctive traffic and timing pattern of the resources constituting a given website to the traffic pattern of a user's client. These attacks work great in simulations, but in practice we hear they don't work nearly as well. We should get some actual numbers to investigate the issue, and figure out what's going on. If we resist these attacks, or can improve our design to resist them, we should. **(Possibly part of end-to-end correlation paper. Otherwise, not in 2007 unless a graduate student is interested.)**

4.2 Implementation security

Right now, each Tor node stores its keys unencrypted. We should **encrypt more Tor keys** so that Tor authorities can require a startup password. We should look into adding intermediary medium-term "signing keys" between identity keys and onion keys, so that a password could be required to replace a signing key, but not to start Tor. This would improve Tor's long-term security, especially in its directory authority infrastructure. **(Design this as a part of the revised "v2.1" directory protocol; implement it in 2007. 3-4 weeks.)**

We should also **mark RAM that holds key material as non-swappable** so that there is no risk of recovering key material from a hard disk compromise. This would require submitting patches upstream to OpenSSL, where support for marking memory as sensitive is currently in a very preliminary state. **(Nice to do, but not in immediate Tor scope.)**

There are numerous tools for identifying trouble spots in code (such as Coverity or even VS2005's code analysis tool) and we should convince somebody to run some of them against the Tor codebase. Ideally, we could figure out a way to get our code checked periodically rather than just once. **(Almost no time once we talk somebody into it.)**

We should try **protocol fuzzing** to identify errors in our implementation. **(Not in 2007 unless we find a grad student or undergraduate who wants to try.)**

Our guard nodes help prevent an attacker from being able to become a chosen client's entry point by having each client choose a few favorite entry points as "guards" and stick to them. We should implement a **directory guards** feature to keep adversaries from enumerating Tor users by acting as a directory cache. **(Do in 2007; 2 weeks.)**

4.3 Detect corrupt exits and other servers

With the success of our network, we've attracted servers in many locations, operated by many kinds of people. Unfortunately, some of these locations have compromised or defective networks, and some of these people are untrustworthy or incompetent. Our current design relies on authority administrators to identify bad nodes and mark them as nonfunctioning. We should **automate the process of identifying malfunctioning nodes** as follows:

We should create a generic **feedback mechanism for add-on tools** like Mike Perry's "Snakes on a Tor" to report failing nodes to authorities. **(Do in 2006; 1-2 weeks.)**

We should write tools to **detect more kinds of innocent node failure**, such as nodes whose network providers intercept SSL, nodes whose network providers censor popular websites, and so on. We should also try to detect **routers that snoop traffic**; we could do this by launching connections to throwaway accounts, and seeing which accounts get used. **(Do in 2007; ask Mike Perry if he's interested. 4-6 weeks.)**

We should add **an efficient way for authorities to mark a set of servers as probably collaborating** though not necessarily otherwise dishonest. This happens when an administrator starts multiple routers, but doesn't mark them as belonging to the same family. **(Do during v2.1 directory protocol redesign; 1-2 weeks to implement.)**

To avoid attacks where an adversary claims good performance in order to attract traffic, we should **have authorities measure node performance** (including stability and bandwidth) themselves, and not simply believe what they're told. Measuring stability can be done by tracking MTBF. Measuring bandwidth can be tricky, since it's hard to distinguish between a server with low capacity, and a high-capacity server with most of its capacity in use. **(Do "Stable" in 2007; 2-3 weeks. "Fast" will be harder; do it if we can interest a grad student.)**

Operating a directory authority should be easier. We rely on authority operators to keep the network running well, but right now their job involves too much busywork and administrative overhead. A better interface for them to use could free their time to work on exception cases rather than on adding named nodes to the network. **(Do in 2007; 4-5 weeks.)**

4.4 Protocol security

In addition to other protocol changes discussed above, we should add **hooks for denial-of-service resistance**; we have some preliminary designs, but we

shouldn't postpone them until we really need them. If somebody tries a DDoS attack against the Tor network, we won't want to wait for all the servers and clients to upgrade to a new version. **(Research project; do this in 2007 if funded.)**

5 Development infrastructure

5.1 Build farm

We've begun to deploy a cross-platform distributed build farm of hosts that build and test the Tor source every time it changes in our development repository.

We need to **get more participants**, so that we can test a larger variety of platforms. (Previously, we've only found out when our code had broken on obscure platforms when somebody got around to building it.)

We need also to **add our dependencies** to the build farm, so that we can ensure that libraries we need (especially libevent) do not stop working on any important platform between one release and the next.

(This is ongoing as more buildbots arrive.)

5.2 Improved testing harness

Currently, our **unit tests** cover only about 20% of the code base. This is uncomfortably low; we should write more and switch to a more flexible testing framework. **(Ongoing basis, time permitting.)**

We should also write flexible **automated single-host deployment tests** so we can more easily verify that the current codebase works with the network. **(Worthwhile in 2007; would save lots of time. 2-4 weeks.)**

We should build automated **stress testing** frameworks so we can see which realistic loads cause Tor to perform badly, and regularly profile Tor against these loads. This would give us *in vitro* performance values to supplement our deployment experience. **(Worthwhile in 2007; 2-6 weeks.)**

We should improve our memory profiling code. **(...)**

5.3 Centralized build system

We currently rely on a separate packager to maintain the packaging system and to build Tor on each platform for which we distribute binaries. Separate package maintainers is sensible, but separate package builders has meant long turnaround times between source releases and package releases. We should create the necessary infrastructure for us to produce binaries for all major packages within an hour or so of source release. **(We should brainstorm this at least in 2007.)**

5.4 Improved metrics

We need a way to **measure the network's health, capacity, and degree of utilization**. Our current means for doing this are ad hoc and not completely accurate

We need better ways to **tell which countries are users are coming from, and how many there are**. A good perspective of the network helps us allocate resources and identify trouble spots, but our current approaches will work less and less well as we make it harder for adversaries to enumerate users. We'll probably want to shift to a smarter, statistical approach rather than our current "count and extrapolate" method.

(All of this in 2007 if funded; 4-8 weeks)

5.5 Controller library

We've done lots of design and development on our controller interface, which allows UI applications and other tools to interact with Tor. We could encourage the development of more such tools by releasing a **general-purpose controller library**, ideally with API support for several popular programming languages. **(2006 or 2007; 1-2 weeks.)**

6 User experience

6.1 Get blocked less, get blocked less broadly

Right now, some services block connections from the Tor network because they don't have a better way to keep vandals from abusing them than blocking IP addresses associated with vandalism. Our approach so far has been to educate them about better solutions that currently exist, but we should also **create better solutions for limiting vandalism by anonymous users** like credential and blind-signature based implementations, and encourage their use. Other promising starting points including writing a patch and explanation for Wikipedia, and helping Freenode to document, maintain, and expand its current Tor-friendly position. **(Do a writeup here in 2007; 1-2 weeks.)**

Those who do block Tor users also block overbroadly, sometimes blacklisting operators of Tor servers that do not permit exit to their services. We could obviate innocent reasons for doing so by designing a **narrowly-targeted Tor RBL service** so that those who wanted to overblock Tor could no longer plead incompetence. **(Possibly in 2007 if we decide it's a good idea; 3 weeks.)**

6.2 All-in-one bundle

We need a well-tested, well-documented bundle of Tor and supporting applications configured to use it correctly. We have an initial implementation well

under way, but it will need additional work in identifying requisite Firefox extensions, identifying security threats, improving user experience, and so on. This will need significantly more work before it's ready for a general public release.

6.3 LiveCD Tor

We need a nice bootable livecd containing a minimal OS and a few applications configured to use it correctly. The Anonym.OS project demonstrated that this is quite feasible, but their project is not currently maintained.

6.4 A Tor client in a VM

a.k.a JanusVM [.....]
which is quite related to the firewall-level deployment section below. JanusVM is a Linux kernel running in VMWare. It gets an IP address from the network, and serves as a DHCP server for its host Windows machine. It intercepts all outgoing traffic and redirects it into Privoxy, Tor, etc. This Linux-in-Windows approach may help us with scalability in the short term, and it may also be a good long-term solution rather than accepting all security risks in Windows.

6.5 Firewall-level deployment

Another useful deployment mode for some users is using **Tor in a firewall configuration**, and directing all their traffic through Tor. This can be a little tricky to set up currently, but it's an effective way to make sure no traffic leaves the host un-anonymized. To achieve this, we need to **improve and port our new TransPort** feature which allows Tor to be used without SOCKS support; to **add an anonymizing DNS proxy** feature to Tor; and to **construct a recommended set of firewall configurations** to redirect traffic to Tor.

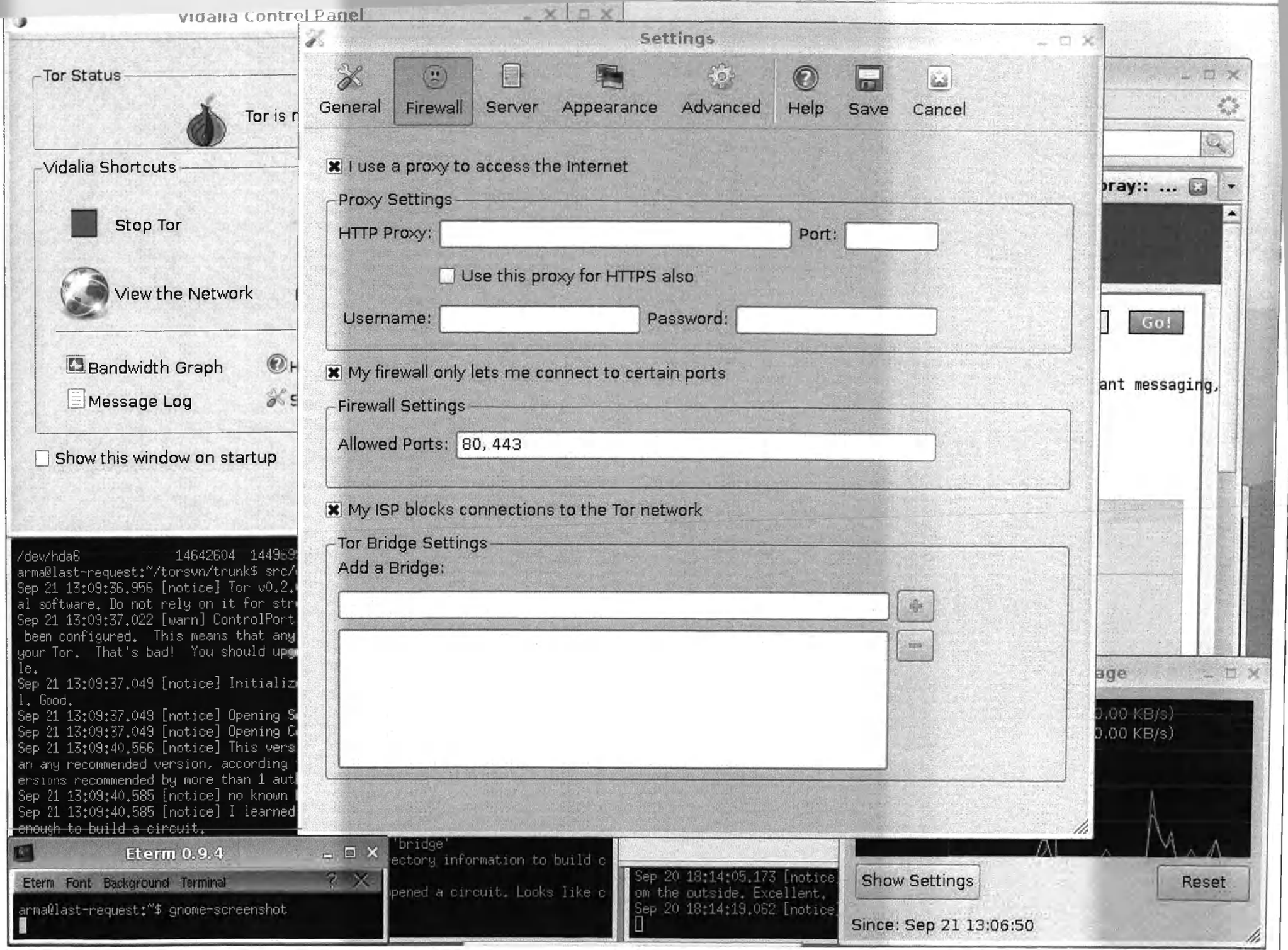
This is an area where **deployment via a livecd**, or an installation targeted at specialized home routing hardware, could be useful.

6.6 Assess software and configurations for anonymity risks

Right now, users and packagers are more or less on their own when selecting Firefox extensions. We should **assemble a recommended list of browser extensions** through experiment, and include this in the application bundles we distribute.

We should also describe **best practices for using Tor with each class of application**. For example, Ethan Zuckerman has written a detailed tutorial on how to use Tor, Firefox, GMail, and Wordpress to blog with improved safety. There are many other cases on the Internet where anonymity would be helpful, and there are a lot of ways to screw up using Tor.

The Foxtor and Torbutton extensions serve similar purposes; we should pick a favorite, and merge in the useful features of the other.


Vidalia Control Panel
Tor Status
Tor is n
Vidalia Shortcuts
Stop Tor
View the Network
Bandwidth Graph
Message Log
Show this window on startup
Settings
General
Firewall
Server
Appearance
Advanced
Help
Save
Cancel
I use a proxy to access the Internet
Proxy Settings
HTTP Proxy:
Port:
Use this proxy for HTTPS also
Username:
Password:
My firewall only lets me connect to certain ports
Firewall Settings
Allowed Ports: 80, 443
My ISP blocks connections to the Tor network
Tor Bridge Settings
Add a Bridge:
Eterm 0.9.4
Eterm Font Background Terminal
arma@last-request:~$ gnome-screenshot
Show Settings
Reset
Since: Sep 21 13:06:50
Go!

SECTION C

DESCRIPTION/SPECIFICATIONS/STATEMENT OF WORK

C.1 BACKGROUND

The Broadcasting Board of Governors (BBG) oversees the mission and operation of several overseas broadcasting entities of the United States Government (USG). The International Broadcasting Bureau (IBB) oversees the daily operations of several USG broadcasters, including the Voice of America (VOA), and is responsible for all contractual and fiscal matters pertaining to broadcast operations. The IBB's Internet anti-censorship program seeks to ensure Internet users in target countries are able to access USG broadcasters' web sites to access their news and other programming, using a variety of tools to counter foreign government-sponsored Internet censorship controls.

This Statement of Work defines those duties the Contractor shall perform to enable the IBB to meet its goals of using Tor as a tool to further its Internet anti-censorship efforts.

C.2 TECHNICAL REQUIREMENTS

C.2.1 The Contractor shall identify, design and develop enhancements to the existing Tor software to increase its suitability as a tool for Internet users in countries with government-sponsored Internet censorship to circumvent censorship controls.

C.2.2 The Contractor shall submit system architecture and technical design documentation for Tor enhancements specifically related to anti-censorship improvements in C.2.1 to the Authorized Representative of the Contracting Officer (AR/CO) for review and approval before implementation.

C.2.3 The Contractor shall continue development of Tor network scalability, with the goal of supporting 2 million or more concurrent end users.

C.2.4 The Contractor shall work with IBB staff and other IBB contractors to identify tasks in support of this program that might be developed collaboratively with Contractor. Tasks involving areas such as documentation, bug fixes, software testing, and any area where specific knowledge of foreign government-sponsored Internet censorship may be especially appropriate for this purpose.

C.2.5 The Contractor shall communicate tasks identified for delegation to IBB in C.2.4 to the AR/CO and negotiate time frames for their completion. The Contractor shall monitor and coordinate work performed by IBB staff on delegated tasks and integrate it into Tor software releases as appropriate.

C.2.6 The Contractor shall promote active growth of the Tor server network and advocacy of Tor products to increase the performance, stability, and usability of Tor, with a focus on the end user experience for users in countries with government-sponsored Internet censorship.

C.3 ADMINISTRATIVE REQUIREMENTS

C.3.1 The Contractor shall provide a Monthly Status Report within ten (10) business days of the end of the month to the AR/CO detailing work performed during the previous month. This report shall describe the work performed for specific requirements of this contract. The report shall also include any other relevant information on Tor changes that might have indirect impacts on contracted work.

C.3.2 The Contractor shall be available for a telephone conference call with the AR/CO, other IBB staff and representatives at a mutually agreeable time on a periodic basis averaging no more than 2 calls per month of one hour's duration. This requirement is in addition to any other required communication by telephone or email with the AR/CO for execution of this contract.

C.4 ADDITIONAL TERMS

C.4.1 All software developed under the terms of this contract must be distributed under an open source software license, such as the "BSD License" or other commonly accepted open source software license as mutually agreed upon by the Contractor and the AR/CO.

C.4.2 All documentation written under the terms of this contract must be distributed under an open source documentation license, such as the "FreeBSD Documentation License" or other commonly accepted open source documentation license as mutually agreed upon by the Contractor and the AR/CO.

COSTS

Cost of 8-month contract per terms above $_____

COST for 12 MONTH EXTENSION

Cost of additional 12-month contract extension per terms above $_____

under way, but it will need additional work in identifying requisite Firefox extensions, identifying security threats, improving user experience, and so on. This will need significantly more work before it's ready for a general public release.

6.3 LiveCD Tor

We need a nice bootable livecd containing a minimal OS and a few applications configured to use it correctly. The Anonym.OS project demonstrated that this is quite feasible, but their project is not currently maintained.

6.4 A Tor client in a VM

a.k.a JanusVM [.....]
which is quite related to the firewall-level deployment section below. JanusVM is a Linux kernel running in VMWare. It gets an IP address from the network, and serves as a DHCP server for its host Windows machine. It intercepts all outgoing traffic and redirects it into Privoxy, Tor, etc. This Linux-in-Windows approach may help us with scalability in the short term, and it may also be a good long-term solution rather than accepting all security risks in Windows.

6.5 Firewall-level deployment

Another useful deployment mode for some users is using **Tor in a firewall configuration**, and directing all their traffic through Tor. This can be a little tricky to set up currently, but it's an effective way to make sure no traffic leaves the host un-anonymized. To achieve this, we need to **improve and port our new TransPort** feature which allows Tor to be used without SOCKS support; to **add an anonymizing DNS proxy** feature to Tor; and to **construct a recommended set of firewall configurations** to redirect traffic to Tor.

This is an area where **deployment via a livecd**, or an installation targeted at specialized home routing hardware, could be useful.

6.6 Assess software and configurations for anonymity risks

Right now, users and packagers are more or less on their own when selecting Firefox extensions. We should **assemble a recommended list of browser extensions** through experiment, and include this in the application bundles we distribute.

We should also describe **best practices for using Tor with each class of application**. For example, Ethan Zuckerman has written a detailed tutorial on how to use Tor, Firefox, GMail, and Wordpress to blog with improved safety. There are many other cases on the Internet where anonymity would be helpful, and there are a lot of ways to screw up using Tor.

The Foxtor and Torbutton extensions serve similar purposes; we should pick a favorite, and merge in the useful features of the other.

6.7 Localization

Right now, most of our user-facing code is internationalized. We need to internationalize the last few hold-outs (like the Tor expert installer), and get more translations for the parts that are already internationalized.

Also, we should look into a **unified translator's solution**. Currently, since different tools have been internationalized using the framework-appropriate method, different tools require translators to localize them via different interfaces. Inasmuch as possible, we should make translators only need to use a single tool to translate the whole Tor suite.

7 Support

It would be nice to set up some **user support infrastructure** and **contributor support infrastructure**, especially focusing on server operators and on coordinating volunteers.

This includes intuitive and easy ticket systems for bug reports and feature suggestions (not just mailing lists with a half dozen people and no clear roles for who answers what), but it also includes a more personalized and efficient framework for interaction so we keep the attention and interest of the contributors, and so we make them feel helpful and wanted.

8 Documentation

8.1 Unified documentation scheme

We need to **inventory our documentation.** Our documentation so far has been mostly produced on an *ad hoc* basis, in response to particular needs and requests. We should figure out what documentation we have, which of it (if any) should get priority, and whether we can't put it all into a single format.

We could **unify the docs** into a single book-like thing. This will also help us identify what sections of the "book" are missing.

8.2 Missing technical documentation

We should **revise our design paper** to reflect the new decisions and research we've made since it was published in 2004. This will help other researchers evaluate and suggest improvements to Tor's current design.

Other projects sometimes implement the client side of our protocol. We encourage this, but we should write **a document about how to avoid excessive resource use**, so we don't need to worry that they will do so without regard to the effect of their choices on server resources.

8.3 Missing user documentation

Our documentation falls into two broad categories: some is 'discoursive' and explains in detail why users should take certain actions, and other documentation is 'comprehensive' and describes all of Tor's features. Right now, we have no document that is both deep, readable, and thorough. We should correct this by identifying missing spots in our design.

References

[1] Roger Dingledine and Nick Mathewson. Tor incentives design brainstorms. `http://tor.eff.org/svn/trunk/doc/incentives.txt`.

[2] Roger Dingledine, Nick Mathewson, and Paul Syverson. Challenges in deploying low-latency anonymity, 2005. Manuscript.

[3] Nick Feamster and Roger Dingledine. Location diversity in anonymity networks. In *Proceedings of the Workshop on Privacy in the Electronic Society (WPES 2004)*, Washington, DC, USA, October 2004. `http://freehaven.net/doc/routing-zones/routing-zones.ps`.

[4] Ian Goldberg. On the security of the tor authentication protocol. In *Proceedings of the Sixth Workshop on Privacy Enhancing Technologies (PET 2006)*, Cambridge, UK, June 2006. Springer. `http://www.cypherpunks.ca/~iang/pubs/torsec.pdf`.

SECTION C

DESCRIPTION/SPECIFICATIONS/STATEMENT OF WORK

C.1 BACKGROUND

The Broadcasting Board of Governors (BBG) oversees the mission and operation of several overseas broadcasting entities of the United States Government (USG). The International Broadcasting Bureau (IBB) oversees the daily operations of several USG broadcasters, including the Voice of America (VOA), and is responsible for all contractual and fiscal matters pertaining to broadcast operations. The IBB's Internet anti-censorship program seeks to ensure Internet users in target countries are able to access USG broadcasters' web sites to access their news and other programming, using a variety of tools to counter foreign government-sponsored Internet censorship controls.

This Statement of Work defines those duties the Contractor shall perform to enable the IBB to meet its goals of using Tor as a tool to further its Internet anti-censorship efforts.

C.2 TECHNICAL REQUIREMENTS

C.2.1 The Contractor shall continue design and development of enhancements to the existing Tor software to increase its suitability as a tool for Internet users in countries with government-sponsored Internet censorship to circumvent censorship controls, based on the existing research and documentation performed during the previous contract period (e.g. as described in the paper "Design of a blocking-resistant anonymity system").

C.2.2 The Contractor shall submit system architecture and technical design documentation for Tor enhancements specifically related to anti-censorship improvements in C.2.1 to the Authorized Representative of the Contracting Officer (AR/CO) for review and approval before implementation. Significant changes to the design that are discovered during implementation must be documented and reviewed by the AR/CO as soon as the Contractor becomes aware of the need for these revisions.

C.2.3 The Contractor shall develop and implement the bridge relay mechanism as designed during the previous contract period to allow individual Tor users to easily reconfigure their Tor client to automatically relay traffic from users in countries with government-imposed Internet censorship so as to circumvent that censorship.

C.2.4 The Contractor shall develop and implement the bridge directory authority mechanism as designed during the previous contract period to allow Tor clients configured as bridge relays (as described in C.2.3) to communicate their existence to the bridge directory authority, and to allow users in countries with government-imposed Internet censorship to discover addresses of available bridge relays so that they may access the Tor network.

C.2.5 The Contractor shall design and develop revisions to the Tor network protocols to hide the network signature of Tor traffic so it cannot be identified Tor traffic and trivially blocked by government-sponsored Internet censors.

C.2.6 The Contractor shall develop and implement enhancements to Tor's cell-based protocol to improve performance on substandard network connections including those with low bandwidth and/or high latency and/or high packet loss.

C.2.7 The Contractor shall continue development of Tor network scalability, with the goal of supporting 2 million or more concurrent end users. This requirement is only a goal for system scalability and is not a requirement on number of actual concurrent users of the Tor network.

C.2.8 The Contractor shall work with IBB staff and other IBB contractors to identify tasks in support of this program that might be developed collaboratively with Contractor. Tasks involving areas such as documentation, bug fixes, software testing, and any area where specific knowledge of foreign government-sponsored Internet censorship may be especially appropriate for this purpose.

C.2.9 The Contractor shall communicate tasks identified for delegation to IBB in C.2.8 to the AR/CO and negotiate time frames for their completion. The Contractor shall monitor and coordinate work performed by IBB staff on delegated tasks and integrate it into Tor software releases as appropriate.

C.2.10 The Contractor shall promote active growth of the Tor server network and advocacy of Tor products to increase the performance, stability, and usability of Tor, with a focus on the end user experience for users in countries with government-sponsored Internet censorship.

C.2.11 The Contractor shall improve the ease of use of Tor for end users by continuing research and development of one or both of the following products: (1) all-in-one software bundle containing Tor and supporting applications, as well as an easy-to-use installer for Microsoft Windows operating systems, as well as option to install and run from a Universal Serial Bus (USB) flash device; (2) bootable CD-ROM image ("LiveCD") which contains a minimal operating system, Tor, and supporting applications. Both would have all appropriate applications pre-configured to use Tor out of the box with only minimal additional configuration required by the end user. If Contractor determines it is not feasible to develop both products, Contractor will provide detailed written technical analysis and explanation to the AR/CO. The Contractor shall make an initial public release of at least one of these products during the term of this contract.

C.3 ADMINISTRATIVE REQUIREMENTS

C.3.1 The Contractor shall provide a Monthly Status Report within ten (10) business days of the end of the month to the AR/CO detailing work performed during the previous month. This report shall describe the work performed for specific requirements of this contract. The report shall also include any other relevant information on Tor changes that might have indirect impacts on contracted work.

C.3.2 The Contractor shall be available for a telephone conference call with the AR/CO, other IBB staff and representatives at a mutually agreeable time on a periodic basis averaging no more than 2 calls per month of one hour's duration. This requirement is in addition to any other required communication by telephone or email with the AR/CO for execution of this contract.

C.4 ADDITIONAL TERMS

C.4.1 All software and accompanying documentation developed under the terms of this contract must be distributed under an open source software license, such as the "BSD License" or other commonly accepted open source software license as mutually agreed upon by the Contractor and the AR/CO.

COSTS

Cost of 12-month contract per terms above $_____

COST for 12 MONTH EXTENSION

Cost of additional 12-month contract extension per terms above $_____

SECTION C

DESCRIPTION/SPECIFICATIONS/STATEMENT OF WORK

C.2 **TECHNICAL REQUIREMENTS**

C.2.12 The Contractor shall continue to develop and implement improvements to the bridge relay and bridge directory authority mechanisms to improve the usability, performance and reliability of the Tor network by users in countries with government-imposed Internet censorship.

C.2.13 The Contractor shall research and document additional options for the scalability of the Tor network beyond 2 million concurrent users, including analysis of splitting the network into multiple segments, switching to datagram-based protocols, and improving the load balancing within the network.

C.2.14 The Contractor shall continue research into the option of providing incentives for Tor users to run Tor relay servers. If further research indicates that this should be pursued, the Contractor shall develop a project plan and timeline for this work. If further research indicates this option should be abandoned, the Contractor shall document and explain in writing the reasoning behind this decision.

C.2.15 The Contractor shall develop a more reliable download mechanism for the Tor browser bundle for users on slow and/or unreliable network connections, by means of a split download of multiple smaller files, implementation of a lightweight download manager, reduction in the software bundle file size, or other method as chosen by the Contractor.

C.2.16 The Contractor shall test the Tor browser bundle on multiple computer systems and analyze these systems afterwards for any changes to the system that may have been made inadvertently by use of the Tor browser bundle. The Contractor shall document any such changes found and develop a plan to reduce the footprint of Tor browser bundle use.

C.2.17 The Contractor shall develop and implement a web-based portal to manage the translations of text into multiple languages for the user interface text of software of Torbutton and Vidalia and other software that may in the future be included in the Tor browser bundle. The web site must allow non-technical users the ability to contribute translations by providing text to be translated in English, as well as any needed context on the use of the text, and allowing users to enter the translation into their language from their web browser.

COSTS

Cost of 12-month extension with additional terms above $_____

SECTION C

DESCRIPTION/SPECIFICATIONS/STATEMENT OF WORK

C.2 TECHNICAL REQUIREMENTS

C.2.18 The Contractor shall implement methods to reduce Tor directory overhead for Tor servers during bootstrapping and maintenance to better support users on low bandwidth connections, by means of Tor proposal 158 ("microdescriptors") or other methods as determined by the Contractor.

C.2.19 The Contractor shall develop and implement changes to the Tor software to reduce the impact of high data volume circuits on the performance of low data volume circuits by dynamically prioritizing writing of data for low volume circuits to the network, thus squeezing the bandwidth of high volume circuits slightly.

C.2.20 The Contractor shall enhance the existing Tor Weather service to provide better support for Tor relay operators on the status and functioning of their Tor server, with flexible levels of notification and notification timeframes which can be customized by each user. Additionally, the Contractor shall notify Tor relay operators of the availability of the Tor Weather service and advocate for its use with more prominent information, documentation, and links to the subscription service in the Tor web pages and software.

C.2.21 The Contractor shall develop and implement a new method to balance traffic over the available bandwidth as provided by the Tor relays, to overcome the problem in the current traffic balancing algorithm which causes fast Tor relays to end up with less load than slow relays. The goal of this revised traffic balancing algorithm should be to reduce latency on Tor circuits as much as possible.

COSTS

Cost of 12-month extension with additional terms above $_____

The Tor Project
56 Waterhouse Street #1
Somerville, MA 02144 USA
http://tor.eff.org/
+ (b) (6)

International Broadcasting Bureau
330 Independence Avenue, S.W.
Washington, DC 20237
ATTN: Ken Berman, Kelly DeYoe

Dear Ken and Kelly,

For our twelve-month contract under our proposed statement of work:

Work proposed for Sections **[C.2.3]** (bridge relay) and **[C.2.4]** (bridge directory authority) includes development and software release with these features. The software release and deployment is the deliverable.
price for this, which is the main part of this year's contract, is **$220.000.00**

Work proposed for section **[C.2.5]** (protocol signature hiding) includes development and software release for these features. The software release and deployment is the deliverable.
price of **$30,000.00**

Work proposed for section **[C.2.6]** (low bandwidth performance improvements) includes development and software release for these features. The design document is the deliverable.
price of **$10,000.00**

Work proposed for section **[C.2.11]** (end-user bundle and/or LiveCD) includes development and software release for these features. The software release and deployment is the deliverable.
price of **$30,000.00** for R&D
price of **$10,000.00** for implementation, test, documentation and release.

Many thanks!

[digitized signature here]

Shava Nerad
Executive Director
http://tor.eff.org/
(b) (6)
(b) (6)
(b) (6) (cell)

The Tor Project
56 Waterhouse Street #1
Somerville, MA 02144 USA
http://tor.eff.org/
+[redacted]

International Broadcasting Bureau
330 Independence Avenue, S.W.
Washington, DC 20237
ATTN: Ken Berman, Kelly DeYoe

Sent as fax, hard copy to follow

Dear Ken and Kelly,

Here are our estimates of work for this year's contract [insert reference number if any here]:

The bridge relay mechanism (C.2.3) and bridge directory authority mechanism (C.2.4) comprise the bulk of the work for the contract, a software release with these components is the main deliverable, estimated cost $220,000.

The protocol hiding (C.2.5) component is estimated cost of $30,000.

The low bandwidth performance improvements design (C.2.6) document is estimated at cost of $10,000.

The Tor end-user bundle and/or LiveCD (C.2.11) is estimated at $30,000 for research and development, and $10,000 for implementation, testing, documentation and release.

Many thanks!

[sig]

Shava Nerad
Executive Director
http://tor.eff.org/
[redacted] (cell)

CW adv_bw P_guard P_middle P_exit Nickname Link
Exit Guard CC AS_num AS_name
1.6592% 0.7573% 0.0000% 0.0000% 4.9777% lumumba
https://atlas.torproject.org/#details/24B1F63F7DF9F85D711864811CC401BE5EB5FB9A Exit nl AS43350
NFOrce Entertainment BV
1.2849% 0.7844% 0.9002% 0.8999% 2.0545% TorLand1
https://atlas.torproject.org/#details/D223399907113A1F216AAA64997BC1D4CFA8E1AC Exit Guard gb AS29302
Hosting Services Inc
1.1028% 0.5260% 0.0000% 0.0000% 3.3083% bouazizi
https://atlas.torproject.org/#details/67FD1D03F922975269F94EC7E4FD38C6D0E5E900 Exit de AS13722
Default Route, Inc.
1.0927% 0.6875% 0.7655% 0.7653% 1.7472% rainbowwarrior
https://atlas.torproject.org/#details/DB8C6D8E0D51A42BDDA81A9B8A735B41B2CF95D1 Exit Guard nl AS43350
NFOrce Entertainment BV
1.0724% 0.7961% 0.0000% 0.0000% 3.2173% politkovskaja2
https://atlas.torproject.org/#details/B93DCC053D7F0472BB17A4514E06FE76D9FB714B Exit nl AS43350
NFOrce Entertainment BV
1.0056% 1.5983% 0.7045% 0.7044% 1.6080% bolobolo3
https://atlas.torproject.org/#details/4F8D80A0F768A2A29856A8F26B05D35DEAA39850 Exit Guard us AS8100
IPTelligent LLC
0.9854% 0.5738% 0.0000% 0.0000% 2.9563% Unnamed
https://atlas.torproject.org/#details/505BD69565964F7B20D51A3FC4A4825BD93CA444 Exit se AS47155
ViaEuropa Sweden
0.9379% 0.6852% 0.6570% 0.6569% 1.4997% chomsky
https://atlas.torproject.org/#details/253DFF1838A2B7782BE7735F74E50090D46CA1BC Exit Guard nl AS43350
NFOrce Entertainment BV
0.8245% 0.5192% 0.0000% 0.0000% 2.4737% Unnamed
https://atlas.torproject.org/#details/AE5A97FA3591F133D8D039232CF0005088190C91 Exit se AS47155
ViaEuropa Sweden
0.7861% 0.5047% 0.5507% 0.5506% 1.2570% assk
https://atlas.torproject.org/#details/8543536F43E4DFD33BFE89204C315515D4DE8B01 Exit Guard se AS51815
Teknikbyran i Sverige AB
0.7578% 0.5547% 0.0000% 0.0000% 2.2733% Unnamed
https://atlas.torproject.org/#details/2624AE0466BD02AFAF3F263D4361D79ABE0E7E05 Exit se AS47155
ViaEuropa Sweden
0.7578% 0.4433% 0.5309% 0.5308% 1.2117% assk2
https://atlas.torproject.org/#details/1A7A34FD161EEF2320728E79FB56391660329955 Exit Guard se AS51815
Teknikbyran i Sverige AB
0.7446% 0.3493% 0.5217% 0.5215% 1.1907% ahmiaTORproxy01
https://atlas.torproject.org/#details/35A9322E265EA3F07E76520D28E0C3BDD68C8F82 Exit Guard fr AS16276
OVH Systems
0.7375% 0.6109% 0.0000% 0.0000% 2.2126% politkovskaja
https://atlas.torproject.org/#details/7DCB5313B9541DD29C94BFDE0ADF91DC91D2CFE9 Exit nl AS43350
NFOrce Entertainment BV
0.5858% 0.7312% 0.4104% 0.4103% 0.9367% manning1
https://atlas.torproject.org/#details/073F27934762FF8BA956FFCE136AAC1CCF45EA13 Exit Guard us AS29761
OC3 Networks & Web Solutions, LLC
0.5757% 0.8604% 0.0000% 0.0000% 1.7270% manning2
https://atlas.torproject.org/#details/D0236B1908B3CC686DB0A361F4931073A25793F1 Exit us AS29761 OC3
Networks & Web Solutions, LLC
0.5666% 0.5592% 0.3969% 0.3968% 0.9059% bolobolo1
https://atlas.torproject.org/#details/9F7A37446BC034B4FDB27CAE2C6CAAB83A40A361 Exit Guard us AS8100
IPTelligent LLC

0.5150% 0.3562% 0.3608% 0.3607% 0.8234% ahmiaTORproxy02
https://atlas.torproject.org/#details/69A8ACED13F9CE359FD8B4FEEA69B66DC8DDF298 Exit Guard fr AS16276 OVH Systems
0.5089% 0.3146% 0.3565% 0.3564% 0.8137% gurgle
https://atlas.torproject.org/#details/948CDA1CE63D2165567B81706CD8C0E9F8934A47 Exit Guard ca AS12093 University of Waterloo
0.5038% 0.7413% 0.3530% 0.3529% 0.8056% manning3
https://atlas.torproject.org/#details/80F870DD215A0C56005266A71C46F92F39F1973B Exit Guard us AS29761 OC3 Networks & Web Solutions, LLC
0.4957% 0.3573% 0.0000% 0.0000% 1.4872% TerrorSquad
https://atlas.torproject.org/#details/8BAEB37A2E5F4A3A9FBB8F90D8901D714C52678B Exit us AS23367 Genesis Adaptive, INC.
0.4856% 0.3744% 0.3402% 0.3401% 0.7765% Amunet5
https://atlas.torproject.org/#details/FE71DDAA8299E9B2998B1C403F362DF507A7F88B Exit Guard us AS22219 Applied Operations, LLC
0.4694% 0.3765% 0.3289% 0.3288% 0.7506% Amunet6
https://atlas.torproject.org/#details/FB6243F6C5EF7436CFFAED108D57A4863D66045B Exit Guard us AS22219 Applied Operations, LLC
0.4634% 0.5181% 0.0000% 0.0000% 1.3901% raskin
https://atlas.torproject.org/#details/4186509C707E96B77B51A76F8294D7E22FF52C61 Exit de AS13722 Default Route, Inc.
0.4492% 0.3448% 0.3147% 0.3146% 0.7183% Amunet3
https://atlas.torproject.org/#details/9D3BFD006D5C65E156DA15E248810017A24B449E Exit Guard us AS22219 Applied Operations, LLC
0.4482% 0.5601% 0.0000% 0.0000% 1.3446% qwertyoruiop1
https://atlas.torproject.org/#details/3F6529905DC70EED873BFFC7172A889E131AAA85 Exit nl AS29073 Ecatel Network
0.4239% 0.3640% 0.2970% 0.2969% 0.6778% Amunet4
https://atlas.torproject.org/#details/FD42AC42239218F8BFE9EB34FE16B5A6B3537832 Exit Guard us AS22219 Applied Operations, LLC
0.4209% 0.3703% 0.2949% 0.2948% 0.6730% Amunet1
https://atlas.torproject.org/#details/21B8466BC4FEF2DCB2FCC8710A4FEA23E108D8B5 Exit Guard us AS22219 Applied Operations, LLC
0.4189% 0.5400% 0.0000% 0.0000% 1.2566% qwertyoruiop2
https://atlas.torproject.org/#details/4A6F047595008A194FB0AE916A24554D556658D7 Exit nl AS29073 Ecatel Network
0.4148% 0.2967% 0.2906% 0.2905% 0.6633% BostonUCompSci
https://atlas.torproject.org/#details/9D4D995AA745A3CAA6276AFAD505D3E4097AA075 Exit Guard us AS111 Boston University
0.4138% 0.3426% 0.2899% 0.2898% 0.6617% Amunet7
https://atlas.torproject.org/#details/5877487A8989EDE4594C4F9E15EC185A80B52CD1 Exit Guard us AS22219 Applied Operations, LLC
0.4057% 0.9931% 0.2842% 0.2842% 0.6487% wau
https://atlas.torproject.org/#details/0ECBAB33DD27A6DA5C1141B39F839F931F92334C Exit Guard ro AS39743 Voxility SRL
0.3956% 0.3614% 0.2771% 0.2771% 0.6325% Amunet2
https://atlas.torproject.org/#details/DB4C1871B146C057CC92D9AE7DF623E99D5133D9 Exit Guard us AS22219 Applied Operations, LLC
0.3511% 0.3230% 0.0000% 0.0000% 1.0532% bolobolo2
https://atlas.torproject.org/#details/C1E2CF4BB774A030FF5408FF35CC637ACE24D439 Exit us AS8100 IPTelligent LLC
0.3146% 0.7384% 0.2204% 0.2204% 0.5031% gorz
https://atlas.torproject.org/#details/F3D4C7479F8789758A77FF61D2D8929311568394 Exit Guard ro AS39743 Voxility SRL

0.2408% 0.3238% 0.0000% 0.0000% 0.7224% anonnode20
https://atlas.torproject.org/#details/6BD3E034D42AB112A8ECE5B95FB904CFC7BFDCAD Exit ua AS44820
Denis Pavlovich Semenyuk
0.2185% 0.1701% 0.1531% 0.1531% 0.3494% SilentT
https://atlas.torproject.org/#details/12C1478D06E76B4A9D49301EC79276D3A7DE8332 Exit Guard us AS16276
OVH Systems
0.2074% 0.5380% 0.1453% 0.1453% 0.3316% sofia
https://atlas.torproject.org/#details/43691853EA556C21A77E006886A5DC579855F527 Exit Guard ro AS39743
Voxility SRL
0.1528% 0.1900% 0.1070% 0.1070% 0.2443% wagtail
https://atlas.torproject.org/#details/131B60B9AFE6AEA60042132D648798534ABEA07E Exit Guard ch AS13030
Init7 Global Backbone
24.1952% 21.0362% 9.8514% 9.8491% 52.8855% (total in selection)

The Tor Project
56 Waterhouse Street #1
Somerville, MA 02144 USA
http://tor.eff.org/
+ (b) (6)

From: Shava Nerad, Tor Development Director
To: Kelly DeYoe, program officer, IBB
RE: contract BBGCON1807S6441
Date: July 10, 2007

This report documents progress in June 2007 on contract BBGCON1807S6441 between IBB and The Tor Project.

C.2.1 The Contractor shall continue design, development and implementation of enhancements to the existing Tor software to increase its suitability as a tool for Internet users in countries with government-sponsored Internet censorship to circumvent censorship controls, based on the existing research and documentation performed during the previous contract period (e.g. as described in the paper "Design of a blocking-resistant anonymity system").

Additional enhancements have been made to the Tor website Chinese translation.

Tor clients can now make encrypted directory connections to bridges. They can also list bridges in their configuration file by IP address and port -- and they can list them with or without knowing the corresponding identity key fingerprint. Not needing to know the identity fingerprint means users in blocked countries can communicate bridge addresses more easily, e.g. by instant message, writing them on business cards, etc. (More work remains on making it work smoothly when a bridge fails or disappears, though -- right now the user is just cut off from the network, and we need to make the user automatically try reconnecting without overloading the bridge with retries.)

We have also started a new research discussion on whether to use two Tor servers in each path or the default of three. Using three "hops" provides a strong level of anonymity, but it also slows down the connection, which is particularly noticeable on slow or unstable network connections. Using two hops would provide a faster connection by sacrificing some security. We are beginning to explore the research questions to learn how much speed we could gain, and how much security we might be sacrificing.

We are working on a new development version of Torbutton, which is the recommended Firefox extension to let users easily configure their Firefox browser to use Tor. This new development version will do many more Firefox configuration aspects automatically, which will allow ordinary people to use Tor and still be safe. In particular, many of our warnings and instructions on how to be safe while using Tor are written in English, and not all users can read and understand them.

C.2.2 *The Contractor shall submit system architecture and technical design documentation for Tor enhancements specifically related to anti-censorship improvements in C.2.1 to the Authorized Representative of the Contracting Officer (AR/CO) for review and approval before development and implementation. Significant changes to the design that are discovered during development must be documented and reviewed by the AR/CO as soon as the Contractor becomes aware of the need for these revisions.*

No revisions this month.

C.2.3 *The Contractor shall develop and implement the bridge relay mechanism, as designed during the previous contract period, to allow individual Tor users to easily reconfigure their Tor client to automatically relay traffic from users in countries with government-imposed Internet censorship so as to circumvent that censorship.*

We released two new development versions of Tor this month: Tor 0.2.0.1-alpha and 0.2.0.2-alpha. These development releases add many of the features described in the May report: new features for people running Tor as both a client and a server (check out the new RelayBandwidth config options); let Tor run as a DNS proxy; and many others. See http://archives.seul.org/or/talk/Jun-2007/msg00026.html **for the full details.**
[This item also applied to C.2.11]

We have implemented the first step for simple bridges: users can put a few lines in their configuration file and they now act as bridges, meaning they publish their descriptors to the bridge directory authority if specified (and if no authority is specified, they do not publish); they answer directory queries that are sent as encrypted directory requests; and they let clients route traffic through them to the rest of the Tor network.

C.2.4 *The Contractor shall develop and implement the bridge directory authority mechanism, as designed during the previous contract period, to allow Tor clients configured as bridge relays (as described in C.2.3) to communicate their existence to the bridge directory authority, and to provide a subset of addresses of available bridge relays to Tor users in countries with government-imposed Internet censorship so that they may access the Tor network.*

We have implemented the first step for simple bridge directory

authorities: we are running a separate experimental bridge directory authority, and bridges can publish their descriptors to it.

C.2.5 *The Contractor shall design and develop revisions to the Tor network protocols to hide the network signature of Tor traffic so it is difficult for government-sponsored Internet censors to identify Tor traffic and trivially block it.*

We have been learning more about the current filtering regimes. It appears that connections to the Tor network are now blocked in Sudan, some parts of Saudi Arabia, and some parts of UAE. We believe the two main technologies behind the blocking are a filtering tool called Smartfilter, and a hardware tool called WatchGuard. We are hoping to learn more about these tools so we can understand where we stand in the arms race.

C.2.6 *The Contractor shall design enhancements to Tor's cell-based protocol to improve performance on substandard network connections including those with low bandwidth and/or high latency and/or high packet loss.*

We continued development towards the "consensus voting" design, which will allow Tor users to learn a unified view of the Tor network while downloading fewer directory-related bytes.

C.2.7 *The Contractor shall continue development of enhancements to improve the scalability of the Tor network toward the goal of supporting 2 million or more concurrent end users. This requirement is only a goal for system scalability and is not a requirement on number of actual concurrent users of the Tor network.*

We continued design work on an incentives design that will encourage more ordinary Tor users to want to relay traffic. We have put this design on the back burner for now though due to time constraints.

Two new papers presented by Tor researchers at the PET workshop this month propose new algorithms for building circuits in Tor that provide similar security properties but scale better and/or use less computation and fewer steps. They are still in the research phase, so much work remains before we can deploy them.

C.2.8 *The Contractor shall work with IBB staff and other IBB contractors to identify tasks in support of this program that might be developed collaboratively with Contractor. Tasks in areas such as documentation, bug fixes, software testing, and any other areas involving specific knowledge of foreign government-sponsored Internet censorship may be especially appropriate for this purpose.*

No reports for this month.

C.2.9 *The Contractor shall communicate tasks identified for delegation to IBB in C.2.8 to the AR/CO and negotiate time frames for their completion. The Contractor shall monitor and coordinate work performed by IBB staff on delegated tasks and integrate it into Tor software releases as appropriate.*

No reports for this month.

C.2.10 *The Contractor shall promote active growth of the Tor server network and advocacy of Tor products to increase the performance, stability, and usability of Tor.*

Most of the Tor research and development community attended the Privacy Enhancing Technologies (PET) workshop in Ottawa, which is the premiere venue for anonymity researchers around the world. Many other researchers are interested in using Tor are a platform for their own research, and the beginnings of several good research papers were discussed that focus on how to scale the Tor network and/or make it more useful and usable.

We continued to collaborate with Hal Roberts (a researcher from the Harvard Berkman Center working with Open Net Initiative and funded by Hivos and the Oak Foundation) on his upcoming report summarizing circumvention tools. This report looks at performance in countries like China, as well as the level of documentation, openness, and community support for each tool.

We finished a draft for an invited piece on Tor to be published in an upcoming issue of ACM Security & Privacy magazine.

We continued to work with a European NGO to finalize a contract related to blocking-resistance R&D. If all goes well, this contract will let us add a third developer -- we are working with Dr. Steven Murdoch, from the University of Cambridge, to bring him on board once we have the contract in place.

Shava presented a talk at Amnesty International's Irrepressible conference via telelink on June 6th. The talk was very well received (in fact, it received international fan mail, as well as kudos from Amnesty staff). Transcript is available here:
http://www.gather.com/viewArticle.jsp?articleId=281474977022186

C. *The Contractor shall improve the ease of use of Tor for end users by continuing research and development on one or both of the following products: (1) all-in-one software bundle containing Tor and supporting applications, as well as an easy-to-use installer for Microsoft Windows operating systems, as well as option to install and run from a Universal Serial Bus (USB) flash device; (2) bootable CD-ROM image ("LiveCD") which contains a minimal operating system, Tor, and supporting applications. Both would have all appropriate*

applications pre-configured to use Tor out of the box with only minimal additional configuration required by the end user. If Contractor determines it is not feasible to develop both products, Contractor will provide detailed written technical analysis and explanation to the AR/CO. The Contractor shall make an initial public release of at least one implementation of one of these products during the term of this contract.

We have an early draft of a "Best Practices" document for Tor LiveCD configurations. This looks at both the set of applications that are necessary and/or useful on a LiveCD, but also specific recommended configurations for each application. In particular, the document aims to unify the configurations from two different Tor LiveCDs under development: the first is "Incognito" being developed by Pat Double, and the second is "Rockate" being developed by Benjamin Schieder. The next steps are to refine the document and to get more community discussion and feedback.

Additional news:

Due to slow recovery from an illness which, among other things, caused Shava to take a bad hit to her liver, Shava is now Development Director of Tor, and Roger is taking up executive director duties, with support from the board.

In the works for the next month or so:

- **new funding is coming in from a nonprofit in France interested in security for citizen journalists**
- **This may allow us to hire new technical staff as well as possibly a volunteer coordinator**

The Tor Project
122 Scott Circle
Dedham, MA 02026 USA
https://www.torproject.org/

From: Roger Dingledine, Tor Project Leader
To: Kelly DeYoe, program officer, IBB
RE: contract BBGCON1807S6441
Date: May 10, 2008

This report documents progress in April 2008 on contract BBGCON1807S6441 between IBB and The Tor Project.

C.2.0 New package releases and related software.

Tor 0.2.0.24-rc (released Apr 22) adds dizum (run by Alex de Joode) as the new sixth v3 directory authority, makes relays with dynamic IP addresses and no DirPort notice more quickly when their IP address changes, fixes a few rare crashes and memory leaks, and fixes a few other miscellaneous bugs. Tor 0.2.0.25-rc (released Apr 23) makes Tor work again on OS X and certain BSDs.
http://archives.seul.org/or/talk/May-2008/msg00014.html

Torbutton 1.1.18 (released Apr 17) fixes many usability and interoperability items, in an attempt to make the new Torbutton not so obnoxious in its zeal to protect the user. It also includes new translations for French, Russian, Farsi, Italian, and Spanish.

We hired Jacob Appelbaum as a full-time contractor in mid April. He will be working on a translation portal, auto update for Tor on Windows and OS X, an email autoresponder for sending Tor clients to users who can't reach our website, and other projects down the road.

We will be hiring Matt Edman as a part-time employee at the beginning of May. He will be working on Vidalia maintenance, bugfixes, and new features --- for example, providing a GUI interface for the above auto update feature, letting users change their preferred language in Vidalia without requiring an application restart, and providing a better GUI for showing Tor's start-up progress.

We worked on a funding proposal to the State Dept's DRL grant in cooperation with Internews and Psiphon. We'll hear about that one... sometime.

We have been awarded two grants by NLNet (http://www.nlnet.nl), a Dutch NGO that emphasizes free-software development and is focusing this year on privacy software. One grant is

to work harder on lowering the overhead of directory requests, especially during bootstrap, and should directly improve the experience for Tor users on modems or cell phones; it will allow us to bring Peter Palfrader on half-time from mid-May to January to accelerate our scalability work. The other grant is to work on making hidden service rendezvous and interaction faster, with the goal of making it easier to set up and advertise a hidden service even for short periods of time; it will allow us to bring Karsten Loesing on quarter-time from mid-May to January so we can work harder in this direction.

The additions of Jacob, Matt, Peter, and Karsten will move Tor from 3 FTE developers to 5 FTE developers.

We gave $5k to the research group of Ian Goldberg, a professor at Waterloo in Canada, to fund his graduate student to work on a UDP design for Tor. Our funding was matched 4x by MITACS, a Canadian research organization similar to NSF.

And that's not all! Google is funding seven students to work on Tor projects over the summer as part of the "Google Summer of Code":
https://blog.torproject.org/blog/congrats-2008-google-summer-code-students%21

C.2.1 The Contractor shall continue design, development and implementation of enhancements to the existing Tor software to increase its suitability as a tool for Internet users in countries with government-sponsored Internet censorship to circumvent censorship controls, based on the existing research and documentation performed during the previous contract period (e.g. as described in the paper "Design of a blocking-resistant anonymity system").

We continued enhancements to the Chinese and Russian Tor website translations.

We did a complete overhaul of the https://check.torproject.org/ page. Now it accepts a language choice, e.g. https://check.torproject.org/?lang=fa-IR

Available languages are German, English, Spanish, Italian, Farsi, Japanese, Polish, Portugese, Russian, and Chinese. The Tor Browser Bundle automatically uses the appropriate language as its home page, based on which language of the Browser Bundle was downloaded.

We started on a documentation page to explain to users what bridges are, how they can decide whether they need one, and how to configure their Tor client to use them:
https://www.torproject.org/bridges.html

C.2.2 The Contractor shall submit system architecture and technical design documentation for Tor enhancements specifically related to anti-censorship improvements in C.2.1 to the Authorized Representative of the Contracting Officer (AR/CO) for review and approval before development and implementation. Significant changes to the design that are discovered during development must be documented and reviewed by the AR/CO as soon as the Contractor becomes aware of the need for these revisions.

We've started working on a design proposal for letting the v3 directory authorities produce a consensus networkstatus even when they disagree about who is a valid authority. As we add more

v3 authorities, it becomes more and more of a hassle to coordinate getting a majority of authorities to upgrade immediately.
https://www.torproject.org/svn/trunk/doc/spec/proposals/134-robust-voting.txt

We've also started working on a design proposal for making it easier to set up a private or testing Tor network. With the advent of the v3 directory protocol, it currently takes up to 30 minutes before a test network will produce a useful networkstatus consensus. Also, there are a dozen different config options that need to be set correctly for a Tor network running on a single IP address to not trigger various security defenses. This approach should let more people set up their own Tor networks, either for testing or because they can't reach the main Tor network.
https://www.torproject.org/svn/trunk/doc/spec/proposals/135-private-tor-networks.txt

We have the beginnings of a grand plan for how to successfully scale the Tor network to orders of magnitude more relays than we have currently. Much more work and thinking remain.
https://www.torproject.org/svn/trunk/doc/spec/proposals/ideas/xxx-grand-scaling-plan.txt

We also did a retrospective on currently open but not finished design proposals, so we don't have as many "open" proposals in the pipeline but not getting attention:
http://archives.seul.org/or/dev/Apr-2008/msg00009.html

C.2.5 The Contractor shall design and develop revisions to the Tor network protocols to hide the network signature of Tor traffic so it is difficult for government-sponsored Internet censors to identify Tor traffic and trivially block it.

As far as we know, nobody's put any effort into blocking our current protocol as it stands, since it no longer says "TOR" in the TLS certificates or "/tor/" in the directory fetch requests.

The next two steps in the arms race will make it harder for an attacker to catch up:

1) Spoof Firefox's ciphersuites in our TLS handshake. That is, extend or adapt OpenSSL internals so that the list of advertised ciphersuites from Tor matches the list that Firefox advertises. This will require advertising ciphers that OpenSSL doesn't actually support, failing safely if those ciphers are actually selected.

2) Spoof Firefox's extensions list in our TLS handshake. Turn on extensions in OpenSSL to match those advertised in Firefox. If any don't exist (we currently think they all do), then find a way to make OpenSSL advertise them without actually supporting them.

We hope to get a first cut at these deployed in June.

C.2.10 The Contractor shall promote active growth of the Tor server network and advocacy of Tor products to increase the performance, stability, and usability of Tor.

Roger and Nick talked to Apu Kapadia at Dartmouth about his plans to open-source Nymble, which is their web-based scheme to let services like Wikipedia blacklist Tor users without needing to (or being able to) learn their location/identity. We're going to continue encouraging them discuss Nymble on or-talk / or-dev, and hopefully sometime in 2008 we will have a first version

ready for testing:
http://www.cs.dartmouth.edu/~nymble/

Roger also talked to Robert Guerra about his DRL proposal as head of a new group at Freedom House. We concluded that we weren't in a position to give him an official letter of endorsement, but that we would be happy to work together if either of us get funded. I asked him to keep me in mind if he has any trainings where I could be useful, since putting me in front of users has been a good move in the past for both me and the users.

Along those lines, Roger also talked to Ethan Zuckerman about the Berkman Center's proposal to DRL. They are hoping to get some funding to do more thorough and periodic analyses of the available circumvention tools; they have Hal Roberts on board, the fellow who did the earlier report that the earlier funders then quashed. Ethan explained that they will continue to emphasize open-source and open-design as critical criteria, so Tor will likely be in good shape going forward if they end up being the ones to do the analyses.

Roger talked to Valer Mischenko at NLNet about some of his plans to make a Privacy CD. Pointed him to Tactical Tech's NGO-in-a-Box project. Valer is the director for NLNet, so it seems smart to keep him happy.

Roger collected a new set of stats for GeoIP-based breakdown of Tor clients. It looks like the overall Tor population has grown by 50% in the past four months, with a particular increase in Germany (our #1 country by user base). We pondered a little bit how to get a more accurate and comprehensive answer; we're hoping to finish a design proposal draft in this direction in May.

Roger went to Beansec, which is a monthly gathering of security professionals in the Boston area, and met a nice fellow from SiteAdvisor, who independently discovered Tor last week and had been thinking of using it to audit websites in a way that the sites don't realize they're being audited. I gave him my card but haven't followed up with him yet.

We added several more research papers that we'd like to see written to the https://www.torproject.org/volunteer#Research page. In May we'll add a few more and then start pointing academic professors at the new list.

Kevin Bauer and Damon McCoy have an upcoming PETS paper on measuring Tor users and usage. We looked through it to give suggestions on how to make their measurements more accurate and their conclusions more useful.

Roger visited Gari Clifford's group at the MIT Media Lab. They're working on citizen journalism in e.g. Bolivia, and want to get something like Tor working for cell phones. I'll meet with them again at the end of May, and see what they've come up with.

C.2.11 *The Contractor shall improve the ease of use of Tor for end users by continuing research and development on one or both of the following products: (1) all-in-one software bundle containing Tor and supporting applications, as well as an easy-to-use installer for Microsoft Windows operating systems, as well as option to install and run from a Universal Serial Bus (USB) flash device; (2) bootable CD-ROM image ("LiveCD") which contains a minimal operating system, Tor, and supporting*

applications. Both would have all appropriate applications pre-configured to use Tor out of the box with only minimal additional configuration required by the end user. If Contractor determines it is not feasible to develop both products, Contractor will provide detailed written technical analysis and explanation to the AR/CO. The Contractor shall make an initial public release of at least one implementation of one of these products during the term of this contract.

The development version of Vidalia now has GUI boxes to configure an http proxy that Vidalia should launch when it starts. (The Tor Browser Bundle already uses these config options internally to launch Polipo when it starts.) The next steps are to make sure that Polipo (our preferred new http proxy) is stable enough on Windows, and then start shipping some new standard bundles with Polipo rather than Privoxy.

We cleaned up the Torbutton install in the OS X bundles so it installs Torbutton for the local user, rather than global. Hopefully this will make OS X users happier.

C.2.12 *The Contractor shall continue to develop and implement improvements to the bridge relay and bridge directory authority mechanisms to improve the usability, performance and reliability of the Tor network by users in countries with government-imposed Internet censorship.*

No work on this item this month.

C.2.13 *The Contractor shall research and document additional options for the scalability of the Tor* network *beyond 2 million concurrent users, including analysis of splitting the network into multiple segments, switching to datagram-based protocols, and improving the load balancing within the network.*

We removed the Tor relay "lefkada" as a v3 directory authority, since it has been down for several months; and set up the Tor relay "dizum" (run by Alex de Joode) as the replacement sixth v3 directory authority.

From the Tor 0.2.0.24-rc ChangeLog:
"Detect address changes more quickly on non-directory mirror relays. Bugfix on 0.2.0.18-alpha; fixes bug 652."

We started work on a patch for OpenSSL that will make it keep less buffer space around. Currently fast Tor relays use (waste) as much as 100M of memory in OpenSSL's buffers.

We made a lot of progress on the 0.2.1.x development tree at reducing our memory overhead. The first 0.2.1.x alpha release will come out in May or June. (It depends when 0.2.0.x finally stabilizes.)

We're making progress on integrating a UPnP library into Vidalia. This feature will allow users who want to set up a Tor relay but don't want to muck with manual port forwarding on their router/firewall to just click a button and have Vidalia interact with their router/firewall automatically. This approach won't work in all cases, but it should work in at least some. We hope to land the first version of this in May.

Steven Murdoch and Robert Watson worked towards a final version of their PETS 2008 paper called "Metrics for Security and Performance in Low-Latency Anonymity Systems." The final version will be available in May at:
http://www.cl.cam.ac.uk/~sjm217/papers/pets08metrics.pdf

C.2.14 The *Contractor shall continue research into the option of providing incentives for Tor users to run Tor relay servers. If further research indicates that this should be pursued, the Contractor shall develop a project plan and timeline for this work. If further research indicates this option should be abandoned, the Contractor shall document and explain in writing the reasoning behind this decision.*
Mike Perry found a major flaw in our earlier "gold star" incentives design: by passing the priority of the client along the entire circuit, we let the exit node correlate the times of certain actions with whether certain relays are on-line at those times. Over time, an attacker can learn which relays are often online when target actions happen. One approach to address this would be to give out e-cash digital coins for good service, and then these coins can be used later even when the relay isn't online. Many issues remain before this alternate design can be considered better, though.

C.2.15 *The Contractor shall develop a more reliable download mechanism for the Tor browser bundle for users on slow and/or unreliable network connections, by means of a split download of multiple smaller files, implementation of a lightweight download manager, reduction in the software bundle file size, or other method as chosen by the Contractor.*

So far there appear to be no free-software zip splitters that work on Windows and produce self-contained exe files for automatically reconstructing the file. Rather than using a closed-source shareware application (as it seems a shame to put a trust gap in our build process when we don't need to), the current plan is to write some instructions for users to fetch the 7zip program, and then fetch a set of blocks, and run a batch file to reconstruct them. We're in the process of trying to learn how large the blocks can be -- preliminary guess is 2MB.

We also started exploring whether we can mail the entire Tor Browser Bundle exe as a gmail attachment. The answer appears to be yes, but we need to zip it first so gmail doesn't complain about an executable attachment. In May we're hoping to set up an email autoresponder to see if the users consider this approach practical also.

C.2.16 The *Contractor shall test the Tor browser bundle on multiple computer systems and analyze these systems afterwards for any changes to the system that may have been made inadvertently by use of the Tor browser bundle. The Contractor shall document any such changes found and develop a plan to reduce the footprint of Tor browser bundle use.*

No work on this item yet. We're planning to get to it in June.

C.2.17 *The Contractor shall develop or adapt existing open source software to implement a web-based portal to manage the translations of text into multiple languages for the user interface text of* software *of Torbutton and Vidalia and other software that may in the future be included in the Tor browser bundle. The web site must allow non-technical users the ability to contribute translations by providing text to be translated in English, as well as any needed context on the use of the text, and allowing users to enter the translation into their language from their web browser.*

We have a first draft of a translation portal up here:
https://www.torproject.org/translation-portal

The Vidalia GUI now has (manual) translation instructions:
http://trac.vidalia-project.net/wiki/Translations

We've registered the Vidalia project on "LaunchPad", which is a web-based translation site that is compatible with Vidalia's string format:
https://translations.launchpad.net/vidalia/trunk/+pots/vidalia
We're currently working to try to upload our current translations into the LaunchPad interface.

We've registered the Torbutton project on "BabelZilla", which is a web-based translation site designed specifically for Firefox extensions. We've uploaded the current translation strings:
http://www.babelzilla.org/index.php?option=com_wts&Itemid=88&extension=3510&type=lang

Lastly, we've begun developer-oriented documentation for how to manage and maintain these various translation web-interfaces:
https://www.torproject.org/svn/trunk/doc/translations.txt

From: Andrew Lewman, Executive Director
To: Kelly DeYoe, program officer, BBG
RE: contract BBGCON1807S6441
Date: March 10, 2009

This report documents progress in February 2009 on contract BBGCON1807S6441 between BBG and The Tor Project.

C. New releases, new hires, new funding

On February 8, we released versions 0.2.0.34-stable and 0.2.1.12-alpha.

Tor 0.2.0.34 features several more security-related fixes. You should upgrade, especially if you run an exit relay (remote crash) or a directory authority (remote infinite loop), or you're on an older (pre-XP) or not-recently-patched Windows (remote exploit).

This release marks end-of-life for Tor 0.1.2.x. Those Tor versions have many known flaws, and nobody should be using them. You should upgrade. If you're using a Linux or BSD and its packages are obsolete, stop using those packages and upgrade anyway.

o Security fixes:
- Fix an infinite-loop bug on handling corrupt votes under certain circumstances. Bugfix on 0.2.0.8-alpha.
- Fix a temporary DoS vulnerability that could be performed by a directory mirror. Bugfix on 0.2.0.9-alpha; reported by lark.
- Avoid a potential crash on exit nodes when processing malformed input. Remote DoS opportunity. Bugfix on 0.2.0.33.
- Do not accept incomplete ipv4 addresses (like 192.168.0) as valid. Spec conformance issue. Bugfix on Tor 0.0.2pre27.

o Minor bugfixes:
- Fix compilation on systems where time_t is a 64-bit integer. Patch from Matthias Drochner.
- Don't consider expiring already-closed client connections. Fixes bug 893. Bugfix on 0.0.2pre20.

Changes in version 0.2.1.12-alpha - 2009-02-08
o Security fixes:
- Fix an infinite-loop bug on handling corrupt votes under certain circumstances. Bugfix on 0.2.0.8-alpha.

- Fix a temporary DoS vulnerability that could be performed by a directory mirror. Bugfix on 0.2.0.9-alpha; reported by lark.
- Avoid a potential crash on exit nodes when processing malformed input. Remote DoS opportunity. Bugfix on 0.2.1.7-alpha.

o Minor bugfixes:
- Let controllers actually ask for the "clients_seen" event for getting usage summaries on bridge relays. Bugfix on 0.2.1.10-alpha; reported by Matt Edman.
- Fix a compile warning on OSX Panther. Fixes bug 913; bugfix against 0.2.1.11-alpha.
- Fix a bug in address parsing that was preventing bridges or hidden service targets from being at IPv6 addresses.
- Solve a bug that kept hardware crypto acceleration from getting enabled when accounting was turned on. Fixes bug 907. Bugfix on 0.0.9pre6.
- Remove a bash-ism from configure.in to build properly on non-Linux platforms. Bugfix on 0.2.1.1-alpha.
- Fix code so authorities _actually_ send back X-Descriptor-Not-New headers. Bugfix on 0.2.0.10-alpha.
- Don't consider expiring already-closed client connections. Fixes bug 893. Bugfix on 0.0.2pre20.
- Fix another interesting corner-case of bug 891 spotted by rovv: Previously, if two hosts had different amounts of clock drift, and one of them created a new connection with just the wrong timing, the other might decide to deprecate the new connection erroneously. Bugfix on 0.1.1.13-alpha.
- Resolve a very rare crash bug that could occur when the user forced a nameserver reconfiguration during the middle of a nameserver probe. Fixes bug 526. Bugfix on 0.1.2.1-alpha.
- Support changing value of ServerDNSRandomizeCase during SIGHUP. Bugfix on 0.2.1.7-alpha.
- If we're using bridges and our network goes away, be more willing to forgive our bridges and try again when we get an application request. Bugfix on 0.2.0.x.

o Minor features:
- Support platforms where time_t is 64 bits long. (Congratulations, NetBSD!) Patch from Matthias Drochner.
- Add a 'getinfo status/clients-seen' controller command, in case controllers want to hear clients_seen events but connect late.

o Build changes:

- Disable GCC's strict alias optimization by default, to avoid the likelihood of its introducing subtle bugs whenever our code violates the letter of C99's alias rules.

C.2.1. Design, develop, and implement enhancements that make Tor a better tool for users in censored countries.

In Tor 0.2.1.12-alpha, if we're using bridges and our network goes away, be more willing to forgive our bridges and try again when we get an application request. Bugfix on 0.2.0.x.

Continued to develop research and coding items for improving Tor's performance using a number of techniques. We're focusing on six main reasons for slow performance: congestion control, tcp backoff, wrong window sizes at start, lack of priority for circuit control cells, and user load from peer to peer bulk data transfers.

We've implemented KDE Marble as an alternate visualization of the world into Vidalia. The first phase is to get a better 3-D globe for clients. The next phase is to enable "click to exit" so users can choose their country of preference for exit nodes.

C.2.2. Architecture and technical design docs for Tor enhancements related to blocking-resistance.

More thoughts written on the possible next steps in an arms race with censors. Document forthcoming.

C.2.5. Hide Tor's network signature.

we sent on Feb 20 a mail: "Tor blocking resistance: likely attacks and defenses"

C. Grow the Tor network and user base. Outreach.

Andrew and Roger attended an Open Society Institute Forum on, "The Future of Freedom and Control in the Internet Age", http://www.soros.org/initiatives/fellowship/events/freedom_20090210. Rebecca MacKinnon and Evgeny Morozov both mentioned Tor and its positive uses multiple times during the talk and subsequent Q&A. We had discussions with the Yahoo Fellow at Georgetown, Susan from Human Rights in China, and a number of OSI people.

Andrew attended Mobile Activism 4 Change barcamp on February 21. This generated some citizen media press about security, privacy, and anonymity in reference to the mobile activist world. You can read more at http://barcamp.org/MobileTechForSocialChangeNewYork. Many organizations who had not previously heard of Tor now know of Tor and its uses. Specific follow up meetings with Students for a Free Tibet, Development Alternatives, and Digital Democracy are in progress.

Jacob attended the InfoActivism camp, http://www.informationactivism.org/, in Bangalore, India. He

gave 20 presentations, trainings, and lectures on Tor.

Produced a guide to Tor and circumvention with the Center for Human Rights and Democracy in Saudi Arabia.

Worked with Global Voices to update their guide to anonymous blogging with Tor and Wordpress. We recommend the Tor Browser Bundle by default, and provide clearer instructions and more pictures to assist users in getting configured quickly and securely.

There was a talk at BlackHat from Xinwen Fu. Our official response and thoughts on the topic are available at https://blog.torproject.org/blog/one-cell-enough

From Feb 6-9, Roger, Nick, Wendy, and Andrew attended ShmooconV, http://shmoocon.org/, in February. Discussed Tor present and futures with many of the attendees. Talked to Bob Stratton, now of Symantec, at length about Tor and the 3-year development roadmap.

End of Feb, Steven and Roger went to Financial Crypto 2009. We talked more with economics and "economics of information security" professors and researchers to get a better intuition about how to balance usability and load on the network. Steven also did a lightning talk on the "TLS footprint" arms race question: should we wait to fix known flaws, to slow down the arms race, or should we fix everything asap to discourage the censors from even trying?

Feb 17, Roger did a guest lecture on Tor in Drexel's senior-level computer security class.

In Feb we also met with the Freedom House people, to help them understand how Tor works and to try to get in on the trainings they're hoping to organize.

In late February, had a 60 minute discussion about Tor with Michael Roffman, a contractor to the State Department. Discussed why Tor over others, why anonymity matters, and how we can help the DRL group achieve their goals.

Jillian C. York continued her blogging for Tor at KnightPulse with "From Tunisia to Japan: Anonymity Everywhere", http://www.knightpulse.org/blog/09/02/25/tunisia-japan-anonymity-everywhere

C.2.11. Preconfigured privacy (circumvention) bundles for USB or LiveCD.

On February 18, we released Tor Browser Bundle 1.1.9 with an updated Tor version to 0.2.1.12-alpha, Vidalia updated to 0.1.11, and Firefox 3.0.6. Andrew has taken over building the bundle to reduce the time between tor releases and bundles which include it.

Updated the Incognito LiveCD TODO list to provide some more direction and tasks for the near future, http://archives.seul.org/or/cvs/Feb-2009/msg00056.html

We continued development and enhancement of TorVM with software updates to libevent, openwrt, vidalia, openvpn, tor, and win pcap. Enhanced the self-extraction and build scripts for easier creation by less technical users.

C. Bridge relay and bridge authority work.

In Tor 0.2.1.12-alpha, if we're using bridges and our network goes away, be more willing to forgive our bridges and try again when we get an application request. Bugfix on 0.2.0.x.

C.2.13. Scalability, load balancing, directory overhead, efficiency.

We wrote up a summary of directory overhead work here:
https://blog.torproject.org/blog/overhead-directory-info%3A-past%2C-present%2C-future

Csaba Kiraly has been doing research on how to reduce the overall
load on the Tor network, while also reducing latency for clients:
http://archives.seul.org/or/dev/Feb-2009/msg00000.html

We have our exit scanner up and working. Roger sent Kelly et al mail about it on Feb 20.

C.2.14. Incentives work.

No changes.

C.2.15. More reliable (e.g. split) download mechanism.

Updated our get-tor email auto-responder to include more languages, added in the English version of the tor browser bundle, tested gmail download and resuming interrupted downloads, and fleshed out the design for easier localization of the message text and commands.

C.2.16. Footprints from Tor Browser Bundle.

No changes.

C.2.17. Translation work, ultimately a browser-based approach.

We had a combined 113 commits across Polish, Chinese, Italian, German, Spanish, Russian, Argentinian, Brazilian Portuguese, and Romanian languages. 41 of these commits were through our translation portal.

The Tor Project
122 Scott Circle
Dedham, MA 02026 USA
https://www.torproject.org/

From: Roger Dingledine, Tor Project Leader
To: Kelly DeYoe, program officer, BBG
RE: contract BBGCON1807S6441
Date: September 10, 2008

This report documents progress in July 2008 on contract BBGCON1807S6441 between BBG and The Tor Project.

C.2.0 New package releases and related software.

Vidalia 0.1.7 (released August 2) fixes a bug that caused Vidalia to not recognize Tor's version correctly in Tor 0.2.0.x, adds an "nsh2po" tool that helps Pootle translate the Vidalia bundle installer strings, adds "TZ=UTC" to the BrowserExecutable's environment variables when launched via Vidalia, and updates the Czech, French, and German translations.
http://trac.vidalia-project.net/browser/vidalia/tags/vidalia-0.1.7/CHANGELOG

Incognito 2008.1 (released August 2) is a Gentoo-based Tor LiveCD. This new release adds a "walkthrough" which will launch on startup; adds language support for Arabic, Green, Hebrew, Russian, and Swedish; improves the support for Chinese and Japanese fonts; adds support for VMWare and partial support for VirtualBox; switches to Tor 0.2.0.30 and Torbutton 1.2.0; and adds some new privacy-supporting software and removes some applications that are too likely to leak private information.
https://svn.torproject.org/svn/incognito/trunk/ChangeLog

Tor 0.2.1.3-alpha (released August 3) implements most of the pieces to prevent infinite-length circuit attacks (see proposal 110); fixes a bug that might cause exit relays to corrupt streams they send back; allows address patterns (e.g. 255.128.0.0/16) to appear in ExcludeNodes and ExcludeExitNodes config options; and fixes a big pile of bugs.
http://archives.seul.org/or/talk/Aug-2008/msg00039.html

Tor 0.2.1.4-alpha (released August 4) fixes a pair of crash bugs in 0.2.1.3-alpha.
http://archives.seul.org/or/talk/Aug-2008/msg00039.html

Tor Browser Bundle 1.1.2 (released August 9) updates Vidalia to version 0.1.6, updates Firefox to 2.0.0.16, updates Tor to 0.2.1.4-alpha, updates Torbutton to 1.2.0, and disables the TZ=UTC environment variable trick since Vidalia 0.1.7 now handles that for us.

https://svn.torproject.org/svn/torbrowser/trunk/README

Vidalia 0.1.8 (released August 17) makes the bandwidth graph window look better for languages like Farsi, includes ssleay32.dll in the Windows packages so Vidalia won't crash when it finds an incompatible version of ssleay32.dll in the user's $PATH, makes "escape" and "return" shortcuts for the settings window, and fixes a variety of other bugs.
http://trac.vidalia-project.net/browser/vidalia/tags/vidalia-0.1.8/CHANGELOG

Tor 0.2.0.30 (released July 15, announced August 21) switches to a more efficient directory distribution design, adds features to make connections to the Tor network harder to block, allows Tor to act as a DNS proxy, adds separate rate limiting for relayed traffic to make it easier for clients to become relays, fixes a variety of potential anonymity problems, and includes the usual huge pile of other features and bug fixes.
http://archives.seul.org/or/announce/Aug-2008/msg00000.html

Tor Browser Bundle 1.1.3 (released August 22) fixes a bug in the 0.1.2 release that messed up translations in the homepage, adds "small=1" to the homepage URL so it doesn't show the huge green onion by default, and updates Vidalia to 0.1.8.
https://svn.torproject.org/svn/torbrowser/trunk/README

Tor 0.2.1.5-alpha (released August 31) moves us closer to handling IPv6 destinations, puts in a lot of the infrastructure for adding authorization to hidden services, lays the groundwork for having clients read their load balancing information out of the networkstatus consensus rather than the individual router descriptors, addresses two potential anonymity issues, and fixes a variety of smaller issues.
http://archives.seul.org/or/talk/Sep-2008/msg00072.html

C.2.1 The Contractor shall continue design, development and implementation of enhancements to the existing Tor software to increase its suitability as a tool for Internet users in countries with government-sponsored Internet censorship to circumvent censorship controls, based on the existing research and documentation performed during the previous contract period (e.g. as described in the paper "Design of a blocking-resistant anonymity system").

We continued enhancements to the Chinese and Russian Tor website translations. We cleaned up the Farsi website translations. Jacob worked with Laurent from Internews to get the Farsi web page translations displaying right-to-left correctly. The tables on the download page still need some work to display well right-to-left, though.

The Tor 0.2.1.3-alpha and 0.2.1.4-alpha releases include more fixes for hidden service performance and robustness, have slightly improved bootstrap status event behavior, and start hunting down a horrible bug that looks like it could leak private information:
https://bugs.torproject.org/flyspray/index.php?do=details&id=779

Now that the Tor 0.2.0.30 release has been declared stable, ordinary users will finally get bridge features, the new harder-to-block network protocol, and other features by default.

C.2.2 The Contractor shall submit system architecture and technical design documentation for Tor

enhancements specifically related to anti-censorship improvements in C.2.1 to the Authorized Representative of the Contracting Officer (AR/CO) for review and approval before development and implementation. Significant changes to the design that are discovered during development must be documented and reviewed by the AR/CO as soon as the Contractor becomes aware of the need for these revisions.

We're working on a draft for a new "automatic software update" protocol, code-named Glider, that incorporates the previous proposals 153 and 154 but is easier to extend to other packages, and is easier to implement and maintain on the server side. We hope to have this new draft out as an actual proposal document, along with some early prototypes of the server side, in September.
https://svn.torproject.org/svn/updater/trunk/specs/glider-spec.txt
Part of the ongoing development question is how to write the client side of this auto update engine in a convenient and easy language like Python, yet have it still be extremely compact on the client side -- since Windows doesn't include Python by default, shipping a Python interpreter with the auto updater could add 10MB to the package size.

Roger sent the list of "research directions we should look at" to or-dev, so more people could look at it:
http://archives.seul.org/or/dev/Aug-2008/msg00031.html
We are working these items into a more comprehensive research and development roadmap; stay tuned.

C.2.5 The Contractor shall design and develop revisions to the Tor network protocols to hide the network signature of Tor traffic so it is difficult for government-sponsored Internet censors to identify Tor traffic and trivially block it.

Nobody has blocked the new signature, as far as we know.

We're thinking about how to choose the right strategy for addressing known potential mechanisms for blocking Tor traffic. Should we fix as many as possible and deploy the fixes? Should we wait until an attacker blocks some of them, and then react? Should we design and implement fixes, but not deploy until forced? The consensus so far seems to be that we should think hard and try to predict what will get blocked next, and design fixes for those cases, but not deploy any fixes until forced.

C.2.10 The Contractor shall promote active growth of the Tor server network and advocacy of Tor products to increase the performance, stability, and usability of Tor.

We answered a lot of press organizations about Tor and the Olympics this month. Our main goal was to explain to technical people how bridges work, what they're for, and explain that in most countries right now Tor works just fine out of the box, so bridges are the backup plan for later down the arms race. The CCC (and others) succeeded in making some good press articles, e.g.
http://www.rsf.org/article.php3?id_article=27991
http://www.guardian.co.uk/technology/2008/aug/07/censorship.hacking
http://www.guardian.co.uk/commentisfree/2008/aug/05/china.censorship

Roger attended Black Hat and Defcon. His Defcon talk was:

"Attacks/Vulnerabilities on Tor: past, present, future"
Slides are at http://freehaven.net/~arma/slides-dc08.pdf
He had a packed room of 500+ people. Lucky Green summarized his take-away from the talk as "we would love to work with you if you find any problems with Tor, and we have a good track record of working well with the community." That sounds like what we were aiming for. We're still waiting for the video to come out so we can link to it from the documentation page.

We also talked a lot with the Mozilla people about privacy-impacting bugs in Firefox. We have a list now:
https://www.torproject.org/torbutton/design/#FirefoxBugs
and should start looking for good Firefox developers to fix them and funding to incent them to do so.

Roger talked to Nate Evans, Christian Grothoff's student, about his attack on Tor and the mini-Defcon talk where he presented it. We explained our progress on implementing proposal 110, and answered other related questions here:
http://archives.seul.org/or/talk/Aug-2008/msg00148.html

We started the paperwork to work with Paul Syverson on his "Foundations of Traffic Security" project at NRL. More on this project in future months once it ramps up.

Roger talked to John Bashinski, our contact at Cisco who used to be a ZKS person, about a conference they're organizing in San Diego in October for law enforcement. They want him to be on an anonymity panel there. Perhaps this will be another avenue for getting more LEO educated about Tor. The overall topic of the conference appears to be 'virtual worlds', so it's a better venue than if it's a "criminals who use privacy tools" conference.

We answered Angelos D. Keromytis (one of the authors on the "approximating a global passive adversary" Tor attack paper) with comments on the paper and suggestions for future directions. TODO: schedule with Angelos to visit NYC at some point, maybe do a Tor talk, and then answer more questions they might have. Keeping these people liking Tor -- and aware that Tor wants to have a hand in the announcement of their new attacks -- will be instrumental in dealing with whatever fallout arises. While I'm there I should meet with Steve Bellovin's lab about their incentives-in-Tor paper, and maybe also do a guest lecture for the NYU law professor ("Ira Rubinstein") I've been ignoring.

We worked with "anonym" to help him submit his Incognito specification as part of the NLnet "User Safety" contest:
http://nlnet.nl/contest/2008/index.html
We should hear in not too long how it fared.

Before Black Hat, Roger attended a 'spook training'. It turned out to be useful mostly for the other speakers he got to hang out with, rather than the ability to teach clueful spooks more about Tor.

Roger talked at length to Robert Stratton, who was once at In-q-tel trying to fund the Freedom network, and is now hanging around DC hooking up people who have ideas to people who have

money. We suggested to him that we'd love to team up and have him write an NSF funding proposal for us -- there's lots to be done and not enough grantwriters on the Tor team, and he seems to know what he's talking about. We should be aware that he will encourage us to commercialize in ways that we're not currently comfortable doing; this is both good and bad. :)

We produced some hypothetical budgets for 2008 / 2009 / 2010 / 2011 based on various permutations of funding coming through. Under our current budget projection, we add Andrew, Mike Perry, and Martin Peck full-time starting in January, keep Karsten on half-time after January, and keep Matt and Peter on quarter-time after January. Then the next funding that arrives (e.g. from NRL, Google, NLnet -- *something* else will arrive :) would be used to bump Karsten up to full-time, and the rest of the funding that appears after that would be used to make sure we can keep paying people in 2010.

We put up our mid-August NLnet reports:
https://www.torproject.org/projects/hidserv#Aug08
https://www.torproject.org/projects/lowbandwidth#Aug08

I sent a follow-up mail to Jim Hughes at Sun, about his potential hardware and funding for "start-ups" like Tor. Andrew picked up the thread after that, and has been continuing with it. Alas, it looks like the free hardware angle is unlikely to work out there.

Coordinated the Google Summer of Code (GSoC) wrapup. We had seven students, and we ultimately decided to pass only four of them. On the positive side, several of the four look like they're going to become long-term contributing community members, and we might even be able to usefully fund one of them (Sebastian) to help Karsten working on Tor metrics and stats in 2009. I filled out our GSoC survey, and made sure everybody else filled out theirs, so we would stay in good standing with Google.
TODO: I should send a follow-up "how GSoC went for Tor" mail to Leslie, talking about both our great results and our lessons learned.

Jacob spent a long week of hacking in Mar Del Plata, Argentina, for DebConf 8 (the yearly Debian Conference). Lots of Tor advocacy. Another box of Tor stickers applied to many many laptops. Lots of people were interested in Tor and many many people installed Tor on both laptops and servers. This advocacy resulted in at least two new high bandwidth nodes that he helped the administrators configure. The first is in Japan. The second is our first major high bandwidth node in New Zealand.

Coverity (coverity.com) is now scanning Tor. It found a bunch of minor memory leaks, a few false positives, and some other miscellaneous bugs. Nick fixed almost all of the bugs in a quick afternoon, excepting some testing code that has some resource leaks. Jacob is going to work on getting other Tor related projects into Coverity.

Mike Perry has been working lately on publicity for moving more high-profile websites to use SSL correctly. Last year at Defcon he reported a bug in how many sites (including GMail) handle their cookies: he basically described an easy way for anybody in Starbucks to steal your GMail cookie and log into your gmail account, even if you are always very careful to only use "https" when logging in to your gmail account. The attack works because cookies *can* be set with an

"only present this cookie on an SSL connection" flag when they're created, but no sites actually set this flag because they are concerned about usability. This attack is easy to perform as a Tor exit relay too. This year, Mike presented an actual tool that performs this attack on a local wireless network in an automated way. Some high-profile sites are slowly moving to use more secure login approaches.

Matt Edman finished running the "Vidalia logo design contest". The contest resulted in 76 entries. There were a lot of ridiculous submissions (Vidalia ninjas?!), but there were also a few good ones. He is tending towards this entry as his choice for the new Vidalia logo:
http://www.worth1000.com/view.asp?entry=479229

Andrew finished the first draft of his "Tor Certification" plan. The idea would be to let commercial organizations use our mark, if their designs pass a security and openness evaluation. First, the trademark office seems to think we need to license our mark to some commercial organizations, to show that we're serious about using it; second, we really would like some organized mechanism for deciding whether a given company is using our name in an accurate way. We expect this discussion to continue for quite a while before anything concrete comes of it.

The DRL grant actually happened! The project will be called "iFree". We're just a subcontractor, and we haven't signed anything with the actual contractor yet. Roger spent much of the end of August working with Eric and Chris to help teach them about our role in the project, and help craft a roadmap and funding timeframe that a) fit with their nascent plans, b) are feasible on our side, and c) are actually in line with where Tor needs to go to be an effective circumvention tool. We also introduced Andrew into the mix, so they can get used to the idea that Tor administration is multiple people. Andrew and Roger will be visiting DC sometime in September for a project-wide meeting. We'll drop by BBG while we're there.

We started the process of coming up with a comprehensive development roadmap, with deliverables and milestones and lead developers, for our piece of the iFree project. Along with that comes a funding spreadsheet with timings, budget numbers, etc. It turns out DRL involves a lot of "pass-down" bureaucracy requirements like workplace initiatives, comprehensive and expensive yearly financial audits, etc, and Andrew tried to factor these indirect costs into the budget too. More on all this in the September status report.

C.2.11 *The Contractor shall improve the ease of use of Tor for end users by continuing research and development on one or both of the following products: (1) all-in-one software bundle containing Tor and supporting applications, as well as an easy-to-use installer for Microsoft Windows operating systems, as well as option to install and run from a Universal Serial Bus (USB) flash device; (2) bootable CD-ROM image ("LiveCD") which contains a minimal operating system, Tor, and supporting applications. Both would have all appropriate applications pre-configured to use Tor out of the box with only minimal additional configuration required by the end user. If Contractor determines it is not feasible to develop both products, Contractor will provide detailed written technical analysis and explanation to the AR/CO. The Contractor shall make an initial public release of at least one implementation of one of these products during the term of this contract.*

Incognito 2008.1 (released August 2) is a Gentoo-based Tor LiveCD. This new release adds a "walkthrough" which will launch on startup; adds language support for Arabic, Green, Hebrew,

Russian, and Swedish; improves the support for Chinese and Japanese fonts; adds support for VMWare and partial support for VirtualBox; switches to Tor 0.2.0.30 and Torbutton 1.2.0; and adds some new privacy-supporting software and removes some applications that are too likely to leak private information.
https://svn.torproject.org/svn/incognito/trunk/ChangeLog

Incognito now comes with much more thorough documentation about which software packages are included, and how they are configured:
http://www.browseanonymouslyanywhere.com/incognito/uploadfiles/docs.html

Incognito's next step is to work on a "hardened" option that uses a more secure kernel and other applications. The goal is to keep the same usability but be even less vulnerable to application-level and kernel-level attacks that could be used to gain access to the system and then try to unveil the user.

Tor Browser Bundle 1.1.2 (released August 9) updates Vidalia to release 0.1.6, updates Firefox to 2.0.0.16, updates Tor to 0.2.1.4-alpha, updates Torbutton to 1.2.0, and disables the TZ=UTC environment variable trick since Vidalia 0.1.7 now handles that for us.
https://svn.torproject.org/svn/torbrowser/trunk/README

Tor Browser Bundle 1.1.3 (released August 22) fixes a bug in the 0.1.2 release that messed up translations in the homepage, adds "small=1" to the homepage URL so it doesn't show the huge green onion by default, and updates Vidalia to 0.1.8.
https://svn.torproject.org/svn/torbrowser/trunk/README

We're working on a new branch of Vidalia that can be used in Tor Browser Bundle, for launching Firefox directly without needing the extra installer scripts called "Firefox Portable". If we get this working, then we can hopefully make progress on running multiple Firefoxes at once (one used for Tor launched by TBB, and one used for non-Tor).
http://trac.vidalia-project.net/browser/vidalia/branches/alt-launcher

The German CCC organization put together a version of the Tor Browser Bundle called the "Freedom Stick" for use in teaching the media about the Chinese firewall and the Olympics:
http://chinesewall.ccc.de/freedomstick-en.html

Work by Martin Peck and Kyle Williams on the Tor VM project continues. We're only paying them quarter-time currently, so we expect it will take a while to land before we get our first prototype, but we hope to see that in September or October. Current design document is under development at https://svn.torproject.org/svn/torvm/trunk/doc/design.html

C.2.12 *The Contractor shall continue to develop and implement improvements to the bridge relay and bridge directory authority mechanisms to improve the usability, performance and reliability of the Tor network by users in countries with government-imposed Internet censorship.*

We mostly spent August making sure that the 0.2.0.30 release was ready and stable for ordinary use. We are hoping that with the new stable bundles, we will get more people signing up to be bridge relays, thus improving the robustness of the bridge system as a whole.

One of the next steps is to investigate what metrics we can use for measuring growth of the set of bridge relays. Down the road we should produce graphs over time, so we can see trends; and we should produce graphs over time of which countries are using bridges most actively. We hope to bring Karsten Loesing on in 2009 to give this topic more attention.

C.2.13 *The Contractor shall research and document additional options for the scalability of the Tor* network *beyond 2 million concurrent users, including analysis of splitting the network into multiple segments, switching to datagram-based protocols, and improving the load balancing within the network.*

Joel Reardon, Ian Goldberg's student at Waterloo, has finished the first draft of his thesis "Improving Tor using a TCP-over-DTLS tunnel":
http://www.cs.uwaterloo.ca/~jreardon/thesis.pdf
We funded this research (along with 4x matching funding from MITACS in Canada) in the hopes that it would move us close enough to being able to switch to a UDP design that we can put it on the Tor development roadmap at some point. Many large challenges remain, but this is also promising work in that it shows that we can expect very serious performance improvements if we go this route.

From the Tor 0.2.1.5-alpha ChangeLog:
"More progress toward proposal 141: Network status consensus documents and votes now contain bandwidth information for each router and a summary of that router's exit policy. Eventually this will be used by clients so that they do not have to download every known descriptor before building circuits."

We're worked on getting "Tor Weather" back up and working:
https://weather.torproject.org/
Weather is a service to let relay operators get notified when their relay is unreachable for an extended period of time. It's still in its early experimental stages, but it's already proved useful to its early testers. It's also using SSL as its base URL now. There is an intermittent failure that isn't always crashing in the same way; Jacob is tracking it down.

Jacob has also been working on a Tor network map, to visualize where our relays are. Using all of the known descriptors, it maps each node with some GeoIP code and plot it onto a map. You can interact with the data to see the IP address of each node, the node name and the city/country information if we could find it. Sadly, it *will* lock your browser up for one or two minutes, as there's a lot of data to parse:
http://freehaven.net/~ioerror/maps/v3-tormap.html

C.2.14 The *Contractor shall continue research into the option of providing incentives for Tor users to run Tor relay servers. If further research indicates that this should be pursued, the Contractor shall develop a project plan and timeline for this work. If further research indicates this option should be abandoned, the Contractor shall document and explain in writing the reasoning behind this decision.*

No progress this month.

C.2.15 *The Contractor shall develop a more reliable download mechanism for the Tor browser bundle for users on slow and/or unreliable network connections, by means of a split download of multiple smaller files, implementation of a lightweight download manager, reduction in the software bundle file size, or other method as chosen by the Contractor.*

The "gettor" email auto-responder is up and mostly working. We still need to do translations for it and other usability features.
https://www.torproject.org/finding-tor

C.2.16 The *Contractor shall test the Tor browser bundle on multiple computer systems and analyze these systems afterwards for any changes to the system that may have been made inadvertently by use of the Tor browser bundle. The Contractor shall document any such changes found and develop a plan to reduce the footprint of Tor browser bundle use.*

No changes.

We've started to think about moving the Tor Browser Bundle from Firefox 2 to Firefox 3. This will mean we should measure new traces. We'll do it once Torbutton is known to be more stable on Firefox 3.

C.2.17 *The Contractor shall develop or adapt existing open source software to implement a web-based portal to manage the translations of text into multiple languages for the user interface text of* software *of Torbutton and Vidalia and other software that may in the future be included in the Tor browser bundle. The web site must allow non-technical users the ability to contribute translations by providing text to be translated in English, as well as any needed context on the use of the text, and allowing users to enter the translation into their language from their web browser.*

We have our translation server up and online:
https://translation.torproject.org/
https://www.torproject.org/translation-portal
Users continued to submit updated translations for many different languages.

The Tor Project
122 Scott Circle
Dedham, MA 02026 USA
https://www.torproject.org/

From: Roger Dingledine, Tor Project Leader
To: Kelly DeYoe, program officer, BBG
RE: contract BBGCON1807S6441
Date: January 10, 2008

This report documents progress in December 2008 on contract BBGCON1807S6441 between BBG and The Tor Project.

C.2.0 New package releases and related software.

Tor 0.2.1.8-alpha (released December 8) fixes some crash bugs in earlier alpha releases, builds better on unusual platforms like Solaris and old OS X, and fixes a variety of other issues.
http://archives.seul.org/or/talk/Dec-2008/msg00129.html

Tor Browser Bundle 1.1.6 (released December 2) and 1.1.7 (released December 12) update Tor to 0.2.1.8-alpha, include a new version of Firefox, and attempt to wrestle with the "AllowMultipleInstances=false" design that could allow us to run Tor Browser Bundle alongside a normal Firefox.
https://svn.torproject.org/svn/torbrowser/trunk/README

Tor 0.2.1.9-alpha (released December 25) fixes many more bugs, some of them security-related.
http://archives.seul.org/or/talk/Jan-2009/msg00029.html

C.2.1 The Contractor shall continue design, development and implementation of enhancements to the existing Tor software to increase its suitability as a tool for Internet users in countries with government-sponsored Internet censorship to circumvent censorship controls, based on the existing research and documentation performed during the previous contract period (e.g. as described in the paper "Design of a blocking-resistant anonymity system").

We continued enhancements to the Chinese and Russian Tor website translations. Our Farsi translation from this summer is slowly becoming obsolete; we should solve that at some point.

Security fixes in the Tor 0.2.1.8-alpha release:
- When the client is choosing entry guards, now it selects at most one guard from a given relay family. Otherwise we could end up with all of our entry points into the network run by the same operator. Suggested by Camilo Viecco. Fix on 0.1.1.11-alpha.

- The "ClientDNSRejectInternalAddresses" config option wasn't being consistently obeyed: if an exit relay refuses a stream because its exit policy doesn't allow it, we would remember what IP address the relay said the destination address resolves to, even if it's an internal IP address. Bugfix on 0.2.0.7-alpha; patch by rovv.
- The "User" and "Group" config options did not clear the supplementary group entries for the Tor process. The "User" option is now more robust, and we now set the groups to the specified user's primary group. The "Group" option is now ignored. For more detailed logging on credential switching, set CREDENTIAL_LOG_LEVEL in common/compat.c to LOG_NOTICE or higher. Patch by Jacob Appelbaum and Steven Murdoch. Bugfix on 0.0.2pre14. Fixes bug 848.

Talked to H D Moore about his "decloak" browser anonymity test server:
http://decloak.net/
It would be great if it had more attacks based on things that Torbutton worked hard to defend against, so we can use it as a regression test. Also, it would be great to get his framework into svn somewhere (either ours or metasploit's) so it can start getting more open.

C.2.2 The Contractor shall submit system architecture and technical design documentation for Tor enhancements specifically related to anti-censorship improvements in C.2.1 to the Authorized Representative of the Contracting Officer (AR/CO) for review and approval before development and implementation. Significant changes to the design that are discovered during development must be documented and reviewed by the AR/CO as soon as the Contractor becomes aware of the need for these revisions.

Proposal 157 ("Make certificate downloads specific") fixes a problem in how clients identify which v3 directory authority certificates they want to download. Previously, clients asked either for "the latest certificate by that authority's identity key" (which might mean they get an old one), or "the latest certificate by a given signing key" (which might mean they get a forged certificate that contains the right signing key signed with the wrong identity key). Now clients specify both the signing key and the identity key they want.
https://svn.torproject.org/svn/tor/trunk/doc/spec/proposals/157-specific-cert-download.txt

Mailed or-dev about the new flaw Peter Eckersley found in our defense to Christian Grothoff's 'infinite length circuit' attack.
http://archives.seul.org/or/dev/Dec-2008/msg00001.html
Currently there are no known fixes for Peter's version of the attack, and it's not looking good that we'll come up with any. Oops.

We finally made our 3-year development roadmap public:
https://blog.torproject.org/blog/our-three-year-development-roadmap-published

C.2.5 The Contractor shall design and develop revisions to the Tor network protocols to hide the network signature of Tor traffic so it is difficult for government-sponsored Internet censors to identify Tor traffic and trivially block it.

Nobody has blocked the new signature, as far as we know.

We have built a plan for how to address potential ways for people to block Tor based on its

network signature. We are aiming to have an internal list of known potential vulnerabilities by early 2009, along with suggested paths to addressing each. Then we can react to actual blocking as it occurs, and periodically update our list of potential flaws and intended solutions as we get more intuition.

Steven has started to build a "Tor session decoder", to understand the detail of how the data is packaged up, in order to improve Tor's traffic analysis and censorship resistance, as well as performance:
http://archives.seul.org/or/dev/Nov-2008/msg00006.html
He has started to uncover some interesting anomalies; hopefully down the road we can figure out the details and address them.

C.2.10 The Contractor shall promote active growth of the Tor server network and advocacy of Tor products to increase the performance, stability, and usability of Tor.

Jillian York continued blogging for us about the good uses of Tor:
http://www.knightpulse.org/blog/tor

"Syria: Using Tor for Censorship Resistance", Dec 1
http://www.knightpulse.org/blog/08/12/01/syria-using-tor-censorship-resistance

"Australia Addresses Internet Circumvention", Dec 19
http://www.knightpulse.org/blog/08/12/19/australia-addresses-internet-circumvention

Howcast produced a quick video for the masses on how to circumvent censorship. We were technical consultants for this video. It's tough to talk about Tor, when the first question you're trying to answer is "What is a proxy? And why do I care?" Howcast did a great job for a high-level overview of circumvention technologies in four minutes.
https://blog.torproject.org/blog/how-circumvent-internet-proxy-howcast

We finally updated the footer on each website page to specify a copyright license. We chose "cc-by", the Creative Commons license that provides the most flexibility to people wanting to reuse our materials.
http://creativecommons.org/licenses/by/3.0/us/

Continued brainstorming with Jake and Andrew about metrics we might use for a "node manager" position as part of the Sesawe project. We want to take some money so we can pay Jake to focus more on relay advocacy and support; but they want to know exactly how successful we will be before they'll give us the money. Work on this item continues.

Continued talking to Aaron Swartz and Virgil Griffith about their tor2web design, and various design questions like "can we configure our Firefox to only proxy .onion urls?"
http://tor2web.org/
Also tried to stress the importance of good use cases and good user stories about why hidden services are important.

Andrew talked to James Cason, the former US Ambassador to Paraguay, about Tor in general.

He's greatly concerned about human rights in Cuba and wanted to talk to a technical person about options. Turns out the biggest problem is lack of uncontrolled internet access. And by controlled, he means it's video cameras in the 1 cybercafe on the island, and using stolen internet access credentials on registered phone lines for dial-up access. After he figures this out, he'll be interested in hearing more about how to get tor working on the island, especially with mobile phones.

Wendy was a panelist at a conference organized by Paul Ohm and others at Colorado U at the beginning of December on law, wiretapping, and research-oriented data collection: "The Law and Ethics of Network Monitoring":
http://www.silicon-flatirons.org/events.php?id=544

Dec 4, had dinner with Jake, Adam Shostack, and hikari in Seattle. Adam asked why we were focusing so much on a "free toolchain" build for TBB and Tor VM and the like. I explained about transparency and repeatability. This flaw in mingw that Steven and Jake found, where it never builds the same executable twice, is really a rough flaw for us, since nobody can actually verify our binaries on Windows.

Dec 5, Roger did a Tor talk for U Washington, wherein he also talked to Justin Cappos (see C.2.15)

Dec 5, had dinner with Jake, Dan Kaminsky, Stuart Schechter, and some other security-and-usability people visiting Microsoft from CMU -- including Serge Egelman, Lorrie Cranor's student at CMU who is finally putting together a user study of Tor exit traffic, now that he's sorted out all the legal issues with the CMU counsel. We'll see how that goes.

Dec 12, Roger did a Tor talk for U Penn. Talked at length with Micah Sherr:
http://chopsticksandlox.com/htmlsite/
who is one of Matt Blaze's grad students and is working on topologies of anonymity systems for his PhD thesis. More on that later I hope.

Talked to "Sebastian Schmidt" about a new commercial anonymity company called Cloakfish
http://www.cloakfish.com/
that was violating our license. The basic idea of the company is to give you Tor modified to use two-hop paths, plus a GUI frontend, and then they sell you subsets of the Tor network -- the more you pay, the more relays you learn about. It got interesting when the Cloakfish people said that Tor approved of their plan, and then some guy named "Roger D" showed up on the Cloakfish forums saying this was a brilliant scheme and everybody should love it.

Roger, Karsten, Sebastian, Steven, Jake, Mike, Peter, Wendy, Frank, Christian, and others attended the 25C3 conference in Berlin, Dec 27-30.
Roger gave a talk there, similar to the DC08 talk but focusing entirely on 'present' and 'future': "Security and anonymity vulnerabilities in Tor: past, present, and future"
http://freehaven.net/~arma/slides-25c3.pdf

Talked to Fabian Keil, a privoxy maintainer and Tor volunteer, who wanted to know what Privoxy had done wrong that we were moving away from them. He explained that they're adding

in keepalive stuff. I told him that Polipo does HTTP 1.1 better and pipelines better. I think Privoxy is trying to catch up. We'll see.

Karsten met with Bernhard Fischer, the OnionCat developer, at 25C3. They talked about various technical things related to hidden services and their performance. Karsten clarified a few things, such as the fact that rendezvous circuits are not re-established after five minutes. "I hope that I have convinced him that establishing multiple circuits to hidden services only to pick the fastest one is not a good solution to improve performance."

There was a workshop after my talk on Germany and data retention. Sebastian Hahn was really great at representing Tor there, particularly because it was right after my talk so I missed half of it, and because it was mostly in German. I tried to add the points that a) I really still do want to do Tor talks for German law enforcement (I got a few leads), and b) the major German Tor relay busts were in 2006-2007, not 2008, and maybe we're finally making progress.

Talked to the Wikileaks people (Daniel and Julian) about their use of Tor hidden services, and how we can make things better for them. It turns out they use the hidden service entirely as a way to keep users from screwing up -- either it works and they know they're safe or it fails, but either way they don't reveal what they're trying to leak locally. So I'd like to add a new "secure service" feature that's just like a hidden service but it only makes one hop from the server side rather than three. A more radical design would be for the "intro point" to be the service itself, so it really would be like an exit enclave.

Jake was among the presenters at 25C3 on a talk about how they had managed to forge a root SSL certificate. In short, this meant that they could pretend to be any https site on the Internet, and no browser would complain. Nick wrote up a response explaining how it works and how it can affect Tor users:
"The MD5 certificate collision attack, and what it means for Tor"
http://blog.torproject.org/blog/md5-certificate-collision-attack%2C-and-what-it-means-tor

C.2.11 *The Contractor shall improve the ease of use of Tor for end users by continuing research and development on one or both of the following products: (1) all-in-one software bundle containing Tor and supporting applications, as well as an easy-to-use installer for Microsoft Windows operating systems, as well as option to install and run from a Universal Serial Bus (USB) flash device; (2) bootable CD-ROM image ("LiveCD") which contains a minimal operating system, Tor, and supporting applications. Both would have all appropriate applications pre-configured to use Tor out of the box with only minimal additional configuration required by the end user. If Contractor determines it is not feasible to develop both products, Contractor will provide detailed written technical analysis and explanation to the AR/CO. The Contractor shall make an initial public release of at least one implementation of one of these products during the term of this contract.*

We went through and updated our step-by-step build instructions for the Tor Browser Bundle, and got Jacob able to build the packages reliably too. Now we can free up Steven for more development work.

Started the conversation with Foebud about their Privacy Dongle:
http://www.privacydongle.de/

It's basically a next-generation Torpark that works on Windows, OSX, Linux. They built their own Firefox extension called TorWaechter which can actually start and stop the Tor process from inside the Chrome. Neat-o. (TODO: need to follow up with them to make sure they send Mike Perry details on TorWaechter.) On the other hand, they didn't know about the new Torbutton, and otherwise had a kind of screwy design. Overall, they handled it very well and really wanted to do the right thing. We should keep working with them to help get it right.

Kyle Williams has been working on a new hardware Tor gadget:
http://hackaday.com/2008/12/21/tor-hardware-privacy-adapter/

Started chatting with the Sesawe folks about a Linux-oriented Tor Browser Bundle, now that China is mandating its own Linux flavor in its Internet cafes. Not much has come of this. We should put it on the roadmap for the next few years.

Wikileaks wants to make a Tor Browser Bundle derivative that ships with Wikileaks as its default homepage, and ships with only one SSL cert (the correct one). This would be neat, but there are some pitfalls: mere possession of TBB doesn't indicate intent nearly as much as possession of the Wikileaks browser does.

Wikileaks would also like a tool integrated which will upload files in chunks, and retry after failures. The problem they have is that Tor connections are a bit flaky and if an upload of a large document fails you have to go all the way back to the start. Indymedia were looking for such a thing a few years ago, for a similar reason (except they weren't using Tor, just flaky modem connections). As far as we know, they didn't find one. It looks like someone would have to spend the time to write one.

Rop Gonggrijp (famous Dutch hacker and anti-voting-computer superstar) wants to make a TBB derivative tailored for Dutch government officials, and send one out to each person along with the note "Elections are coming up. Anything you feel you should share with your country?" I suspect nothing will come of this unless we help him make it happen, but wouldn't that be neat.

C.2.12 *The Contractor shall continue to develop and implement improvements to the bridge relay and bridge directory authority mechanisms to improve the usability, performance and reliability of the Tor network by users in countries with government-imposed Internet censorship.*

Karsten began to crunch the numbers on all our historical bridge relay information, to look for trends, and to start being able to display the database more graphically. More news on this in January.

C.2.13 *The Contractor shall research and document additional options for the scalability of the Tor* network *beyond 2 million concurrent users, including analysis of splitting the network into multiple segments, switching to datagram-based protocols, and improving the load balancing within the network.*

New feature from the Tor 0.2.1.8-alpha ChangeLog:
- New DirPortFrontPage option that takes an html file and publishes it as "/" on the DirPort. Now relay operators can provide a disclaimer without needing to set up a separate webserver.

There's a sample disclaimer in contrib/tor-exit-notice.html.

Performance scalability fixes from the Tor 0.2.1.9-alpha ChangeLog:
- Clip the MaxCircuitDirtiness config option to a minimum of 10 seconds. Warn the user if lower values are given in the configuration. Bugfix on 0.1.0.1-rc. Patch by Sebastian.
- Clip the CircuitBuildTimeout to a minimum of 30 seconds. Warn the user if lower values are given in the configuration. Bugfix on 0.1.1.17-rc. Patch by Sebastian.

Relay stability fixes from the Tor 0.2.1.9-alpha ChangeLog:
- Fix a logic error that would automatically reject all but the first configured DNS server. Bugfix on 0.2.1.5-alpha. Possible fix for part of bug 813/868. Bug spotted by coderman.
- When we can't initialize DNS because the network is down, do not automatically stop Tor from starting. Instead, retry failed dns_init() every 10 minutes, and change the exit policy to reject *:* until one succeeds. Fixes bug 691.

Karsten discovered a bug where some directory authorities would take many minutes to send out a network status, because they were rate limiting too low. The short-term fix is to get those authorities to set
"MaxAdvertisedBandwidth 10 KB"
in their torrc, so they don't spend as much of their bandwidth relaying ordinary Tor traffic.
https://bugs.torproject.org/flyspray/index.php?do=details&id=847
We need to consider longer-term solutions too, where clients actually recover more gracefully from this situation.

Sent my geoip-counting patches to Karsten so he can begin playing around with trying to estimate the number of Tor users in each country. Wrote some very simple notes on where I think we're undercounting and where I think we're overcounting with the patch. More on that in January.

C.2.14 The *Contractor shall continue research into the option of providing incentives for Tor users to run Tor relay servers. If further research indicates that this should be pursued, the Contractor shall develop a project plan and timeline for this work. If further research indicates this option should be abandoned, the Contractor shall document and explain in writing the reasoning behind this decision.*

Talked to Tom Heydt-Benjamin (of IBM Zurich / ETH) about zero-knowledge proof approaches for anonymous credentials, in the context of the incentive designs. Wouldn't it be nice if we could issue credentials that users could use to prove to each relay in the circuit that they should get priority, without revealing other info? I think the level of crypto overhead is too high though to make this practical, alas.

My next step, sometime in January, is to put up a blog entry on our incentives tech report, and let or-talk and others know about it.

C.2.15 *The Contractor shall develop a more reliable download mechanism for the Tor browser bundle for users on slow and/or unreliable network connections, by means of a split download of multiple smaller files, implementation of a lightweight download manager, reduction in the software bundle file size, or other method as chosen by the Contractor.*

We continued work on Thandy (our secure updater) this month.

Thandy itself is working smoothly at this point -- it can contact the central repository, check all the keys, look in the registry and compare the currently installed version to the new choices, fetch the right packages, check all the signatures, and launch the install.

We also now have a branch of Vidalia that has the GUI components for our updater in and working. It launches the updater to check for updates periodically, and there's a "check now" button. It does the update via Tor if Tor is up and running, and via direct connection otherwise.

We had hoped to be able to get away with patching our current .nsi Windows installer, but it turns out that "nsi silent (non-GUI) install" and "Vista" are not compatible concepts: Vista only likes MSI-based silent installs, due to that whole permissions thing that Vista gets so excited about.

So we now have a shiny new wxs-based msi installer for Tor on Windows:
https://svn.torproject.org/svn/tor/trunk/contrib/tor.wxs.in
with buildbot-style output here:
https://data.peertech.org/torbld

The new installer has been tested for install, upgrade, repair and removal. But that's just Tor, and our recommended download bundle contains four components: Tor, Vidalia (the GUI), Torbutton (our Firefox extension), and either Privoxy or Polipo (an http proxy configured to use Tor -- we're migrating from Privoxy to Polipo).

So, the next step is to work on MSI installer files for the other three, plus a meta-msi file for the bundle. We're aiming to have a first go of that at the beginning of January. That way we can give a simpler demo of "download this bundle, then it will automatically notice that it should upgrade Tor, and it will fetch the new package and upgrade."

In other news, I had a long chat with Justin Cappos in early December. Justin did his PhD thesis on security of package managers, and is now a post-doc at UW working on (among other things) auto-update frameworks. He was really excited about our design, and wants to incorporate it into various academic grant proposals he's writing. So that's some good academic validation at least. See the beginning of the thread here:
http://archives.seul.org/or/dev/Dec-2008/msg00010.html

C.2.16 The *Contractor shall test the Tor browser bundle on multiple computer systems and analyze these systems afterwards for any changes to the system that may have been made inadvertently by use of the Tor browser bundle. The Contractor shall document any such changes found and develop a plan to reduce the footprint of Tor browser bundle use.*

No changes.

C.2.17 *The Contractor shall develop or adapt existing open source software to implement a web-based portal to manage the translations of text into multiple languages for the user interface text of* software

of Torbutton and Vidalia and other software that may in the future be included in the Tor browser bundle. The web site must allow non-technical users the ability to contribute translations by providing text to be translated in English, as well as any needed context on the use of the text, and allowing users to enter the translation into their language from their web browser.

We have our translation server up and online:
https://translation.torproject.org/
https://www.torproject.org/translation-portal

The Tor Project
122 Scott Circle
Dedham, MA 02026 USA
https://www.torproject.org/

From: Roger Dingledine, Tor Project Leader
To: Kelly DeYoe, program officer, BBG
RE: contract BBGCON1807S6441
Date: August 10, 2008

This report documents progress in July 2008 on contract BBGCON1807S6441 between BBG and The Tor Project.

C.2.0 New package releases and related software.

Torbutton 1.2.0rc5 (released July 6) provides improved addon compatibility, better preservation of Firefox preferences that we touch, fixing issues with Tor toggle breaking for some option combos, and an improved 'Restore Defaults' button. This version also features Firefox 3 cookie jar support, and support for storing cookie jars in memory.
http://archives.seul.org/or/talk/Jul-2008/msg00026.html

Vidalia 0.1.6 (released July 8) fixes a bug introduced in 0.1.3 that could cause excessive CPU usage or crashing on some platforms; continues to prepare Vidalia's strings for easier translation; adds a Romanian GUI and installer translation; and updated the Farsi, Finnish, French, German, and Swedish translations.
http://trac.vidalia-project.net/browser/vidalia/tags/vidalia-0.1.6/CHANGELOG

Tor 0.2.0.29-rc (released July 8) fixes two big bugs with using bridges, fixes more hidden-service performance bugs, and fixes a bunch of smaller bugs.
http://archives.seul.org/or/talk/Jul-2008/msg00038.html

Torbutton 1.2.0rc6 (released July 12) features fixes for a nasty history loss bug, an exception during Tor toggle, javascript being disabled in some tabs, better pref handling, and more.
http://archives.seul.org/or/talk/Jul-2008/msg00049.html

Tor 0.2.0.30 (released July 15) is the first stable release of the 0.2.0.x branch. The previous stable branch (0.1.2.x) went stable in April of 2007. We are still waiting for Torbutton and Vidalia to stabilize before announcing the Windows and OS X packages on the or-announce announcements list. We expect to do that in August.

Tor Browser Bundle 1.1.1 (released July 20) updates Vidalia to release 0.1.6, updates Pidgin

Portable to 2.4.3, updates Pidgin OTR plugin to 3.2, updates Tor to 0.2.1.2-alpha, updates Torbutton to 1.2.0rc6, and sets TZ=UTC environment variable in RelativeLink (needed by Torbutton).
https://svn.torproject.org/svn/torbrowser/trunk/README

Torbutton 1.2.0 (released July 30) is finally a stable release for the new Torbutton tree that includes application-level privacy protections.
https://svn.torproject.org/svn/torbutton/trunk/src/CHANGELOG

C.2.1 The Contractor shall continue design, development and implementation of enhancements to the existing Tor software to increase its suitability as a tool for Internet users in countries with government-sponsored Internet censorship to circumvent censorship controls, based on the existing research and documentation performed during the previous contract period (e.g. as described in the paper "Design of a blocking-resistant anonymity system").

We continued enhancements to the Chinese and Russian Tor website translations. We added Vidalia, Torbutton, and website translations into Farsi.

From the Tor 0.2.0.29-rc ChangeLog:
"When a hidden service was trying to establish an introduction point, and Tor had built circuits preemptively for such purposes, we were ignoring all the preemptive circuits and launching a new one instead. Bugfix on 0.2.0.14-alpha."
"When a hidden service was trying to establish an introduction point, and Tor *did* manage to reuse one of the preemptively built circuits, it didn't correctly remember which one it used, so it asked for another one soon after, until there were no more preemptive circuits, at which point it launched one from scratch. Bugfix on 0.0.9.x."

The upcoming Tor 0.2.1.3-alpha and 0.2.1.4-alpha releases include more fixes for hidden service performance and robustness, have slightly improved bootstrap status event behavior, and start hunting down a horrible bug that looks like it could leak private information:
https://bugs.torproject.org/flyspray/index.php?do=details&id=779

C.2.2 The Contractor shall submit system architecture and technical design documentation for Tor enhancements specifically related to anti-censorship improvements in C.2.1 to the Authorized Representative of the Contracting Officer (AR/CO) for review and approval before development and implementation. Significant changes to the design that are discovered during development must be documented and reviewed by the AR/CO as soon as the Contractor becomes aware of the need for these revisions.

Proposal 145 (Separate "suitable as a guard" from "suitable as a new guard") suggests one approach for separating the role of "is still useful as an entry guard" from "should be an option when choosing a new entry guard". This step will help us load balance over the network better.
https://svn.torproject.org/svn/tor/trunk/doc/spec/proposals/145-newguard-flag.txt

Proposal 146 (Add new flag to reflect long-term stability) discusses how to ship the Tor client with a set of alternate sources for initial bootstrap directory information. We already have this feature

in Tor 0.2.0.x, called the "fallback consensus", but we never enabled it because the Tor client would spend too long trying directory mirrors that were long since gone from the network. This proposal moves us closer to being able to distinguish the more long-term reliable mirrors.
https://svn.torproject.org/svn/tor/trunk/doc/spec/proposals/146-long-term-stability.txt

Proposal 147 (Eliminate the need for v2 directories in generating v3 directories) helps wean us off of needing the old deprecated v2 directory design. Currently we only use it to give advance warning to the v3 authorities about relays that haven't heard about yet, so they can fetch information about those relays before the time arrives to make an official vote about their state.
https://svn.torproject.org/svn/tor/trunk/doc/spec/proposals/147-prevoting-opinions.txt

Proposal 148 (Stream end reasons from the client side should be uniform) describes a simple fix for a potential anonymity flaw in Tor's core protocol for passing explanations from one end of a Tor circuit to the other when an application stream ends.
https://svn.torproject.org/svn/tor/trunk/doc/spec/proposals/148-uniform-client-end-reason.txt

Proposal 149 (Using data from NETINFO cells) starts talking about how to make use of the timestamp and IP address listed in Tor's new NETINFO cells. In theory we can use them to decide if our clock is skewed, and to decide if a traffic analysis man-in-the-middle attack is happening against us. In practice it appears more complex than we expected.
https://svn.torproject.org/svn/tor/trunk/doc/spec/proposals/149-using-netinfo-data.txt

Proposal 150 (Exclude Exit Nodes from a circuit) allows users to specify which relays should never be used as the last (exit) hop in a circuit. We took the proposal one step further and allowed users to also specify IP addresses and netmasks for which relays to avoid in the exit position.
https://svn.torproject.org/svn/tor/trunk/doc/spec/proposals/150-exclude-exit-nodes.txt

Proposal 151 (Improving Tor Path Selection) is a draft proposal to implement the results of Fallon Chen's Google Summer of Code project. Her plan is to measure the expected time it takes to establish a circuit, and then abandon circuits that take significantly longer than that to form. The assumption is that circuits that take a long time to set up will generally have unacceptably high latency as well.
https://svn.torproject.org/svn/tor/trunk/doc/spec/proposals/151-path-selection-improvements.txt

Proposal 154 (Automatic Software Update Protocol) starts the discussion of how to let Vidalia automatically manage updates for Tor, Polipo, Vidalia, etc. This is very important for keeping users up to date with respect to security and stability fixes. We will especially aim to do the updates over Tor, a) for privacy, and b) so users who are blocked from the Tor website will still be able to upgrade seamlessly.
https://svn.torproject.org/svn/tor/trunk/doc/spec/proposals/154-automatic-updates.txt

C.2.5 The Contractor shall design and develop revisions to the Tor network protocols to hide the network signature of Tor traffic so it is difficult for government-sponsored Internet censors to identify Tor traffic and trivially block it.

Nobody has blocked the new signature, as far as we know.

On the other hand, our internal red teaming has uncovered some potential ways to distinguish Tor's protocol from a "real" Firefox talking to Apache. See the separately sent mail for details.

C.2.10 The Contractor shall promote active growth of the Tor server network and advocacy of Tor products to increase the performance, stability, and usability of Tor.

.he Tor VoA Russian page is now up:
http://www.voanews.com/russian/tor.cfm

Answered questions from Clothilde Le Coz, the new head at RSF. She wanted to know if there is a simple easy way to make rsfblog unfilterable. (I fear I gave her some bad news.)

Talked to [redacted] **about a training she wants to run in Prague in late 2008 or early 2009, that she wants me to attend / help with. She promised to send me more info when she has it.**

Sent BBG more details about the user counting algorithm we've been running along with results. His reason was "Some folks in Congress want to fund - via a cutout - more $$ for Tor and are looking for some usage numbers." Perhaps one day we'll find out what that means. :)

Had several conference calls with Jeanne Bourgault and Kathleen Reen of Internews DC, about the potential DRL funding. Worked with Chris Walker and other Internews folks to revise our "what we'll do for what amount of money" roadmap proposals to more accurately reflect what we actually plan to do. Aimed for a slightly larger amount of money than we first aimed for. Tried to help with revising the actual main funding proposal (or even getting to see it!), but was rebuffed again. Participated in a conference call with the State Dept contact to help him understand how Tor works and what our talking points are. The politics behind the funding remain ongoing.

Got a copy of Jon McLachlan's "congestion attacks on bridges" paper from UMN. I should read it more thoroughly, and see how useful / discouraging it is at letting Tor clients run as bridge relays safely.

Started Andrew talking to Kristen Taylor at the Knight Foundation about taking some of their money in exchange for educating journalists better. Andrew has been running with that using a "let's do a weekly blog about privacy/anonymity issues" theme. The current plan we're pondering is to give the money to somebody at the Berkman Center who wants to keep the blog going.

Chatted with Nart about our tor.anonymity.cn website mirror, which is located inside China. TODO: Should we list it in our main mirrors list? It isn't reachable from outside the Firewall, since it contains bad words. Should we list it on the gettor page, for people inside China who read the cached copy of that page? Should we try to get more mirrors like it?

Went to the "Monte Verita" information security retreat in Switzerland. Gave a Tor talk. Hung out with Susan Landau and Jim Hughes from Sun (TODO: Jim offered us some Sun hardware and support), Peter Gutmann (who told us about some gcc warnings we were using in poor

judgement; since corrected by Nick), Brian Snow (who hinted that NSA was working hard to catch up on how to surveil networks like Tor, but wouldn't be there for a few years yet), Arjen Lenstra at EPFL (who is planning to set up a Tor server for research (TODO: mail them and remind them), Marshall Van Alstyne (an economist at BU / MIT who was fascinated with the incentives.pdf plan), and a variety of grad students including two from Iran who would be good for reviewing our Farsi (TODO: contact them and ask them to do so).

Worked with "mfr" (our French translator) to improve many pieces of the website. Worked with Jake to make sure the Farsi translations got up and working right.

Sent Christian Grothoff (a prof at Denver) an official "yes, if you produced a tool to find memory fragmentation issues, Tor would love to use it" letter to boost the chances for his NSF proposal. (Memory fragmentation was the big problem with bloat in 0.1.2.x and 0.2.0.x.)

Introduced Geoff Goodell at the Berkman Center to various General Counsel for Indiana University and CMU. He's trying to coordinate people around Boston-area universities so they can better ponder the implications of Tor relays at universities.

Read, reviewed, and discussed some of the WPES (Workshop on Privacy in the Electronic Society) papers. The accepted papers are up here:
http://dais.cs.uiuc.edu/wpes08/papers.html

Helped put up our mid-July NLnet reports:
https://www.torproject.org/projects/hidserv#Jul08
https://www.torproject.org/projects/lowbandwidth#Jul08

Many Tor developers and volunteers went to the Privacy Enhancing Technology Symposium in Leuven. This is the annual gathering of anonymity and privacy researchers from around the world. Lately many of the papers and talks have been about Tor research.
http://petsymposium.org/2008/

I talked to Chris Alexander (one of Ian Goldberg's students at Waterloo) at length at PETS about his potential Master's thesis topic. The topic I proposed is how to give better service to quieter streams, with the ultimate goal of letting the Tor network tolerate file-sharing users better.

I talked to Nick Hopper and Vitaly Shmatikov at PETS, to let them know about the new research items on the volunteer page.
https://www.torproject.org/volunteer#Research

Got Aaron Johnson, Paul, Nick, and Steven talking to each other about the NRL research project we're going to be helping with.

Helped coordinate the program and actual running of the "HotPETS" workshop attached to the PET Symposium. This was 2/3 of the last day, and involved much more discussion and audience participation than the other (more dry, more formal) talks.

Ian Goldberg's student Joel is finishing his Master's thesis draft right about now. He's been

working on a new Tor transport mechanism that uses UDP rather than TCP. This approach has a lot more unsolved problems, but it also solves a lot of known open problems with the current Tor transport.

Also finished dealing with deciding on and allocating the PETS stipend pool, which is sponsored by Microsoft each year. We managed to help fund 20ish people this year, including many Tor people.

We've continued talking to the authors of a variety of research papers that present attacks on low-latency anonymity systems such as Tor. One of the research groups is at Columbia University; another is in Denver; a third is in Greece. Roger covered quite a few of these topics in his Defcon talk in early August. Ideally the video for that talk will be coming out in the next month or two.

We have continued to talk to the Hong Kong newspaper contact about shipping thousands of Tor Browser Bundle USB keys. Andrew is mostly leading that at this point.

Helped Dan Kaminsky relay his DNS queries via the Tor network, while he was experimenting with measuring the prevalence of his DNS bug. Jacob helped even more by providing some scripts.

Karsten Loesing's report on 7 ways to improve the performance and robustness of Tor hidden services:
http://freehaven.net/~karsten/hidserv/discussion-2008-07-15.pdf

Four new research papers on Tor came out in July:
http://freehaven.net/anonbib/#loesing2008performance
http://freehaven.net/anonbib/#improved-clockskew
http://freehaven.net/anonbib/#mccoy-pet2008
http://freehaven.net/anonbib/#danezis-pet2008

C.2.11 *The Contractor shall improve the ease of use of Tor for end users by continuing research and development on one or both of the following products: (1) all-in-one software bundle containing Tor and supporting applications, as well as an easy-to-use installer for Microsoft Windows operating systems, as well as option to install and run from a Universal Serial Bus (USB) flash device; (2) bootable CD-ROM image ("LiveCD") which contains a minimal operating system, Tor, and supporting applications. Both would have all appropriate applications pre-configured to use Tor out of the box with only minimal additional configuration required by the end user. If Contractor determines it is not feasible to develop both products, Contractor will provide detailed written technical analysis and explanation to the AR/CO. The Contractor shall make an initial public release of at least one implementation of one of these products during the term of this contract.*

Tor Browser Bundle 1.1.1 (released July 20) updates Vidalia to release 0.1.6, updates Pidgin Portable to 2.4.3, updates Pidgin OTR plugin to 3.2, updates Tor to 0.2.1.2-alpha, updates Torbutton to 1.2.0rc6, and sets TZ=UTC environment variable in RelativeLink (needed by Torbutton).

The first Incognito (Gentoo-based Tor LiveCD) release of 2008 is also nearing completion, and we expect to see it released in August.

Finally, we contracted to Martin Peck and Kyle Williams to start work on the Tor VM project. The idea is to run a Linux kernel and a Tor client inside a thin VM (like QEMU) on Windows, and then transparently intercept outgoing connections and redirect them into Tor. This approach will a) make proxy-avoiding side-channel and sidejacking attacks less devastating, and b) isolate the Tor client from the rest of the OS to provide a more robust security approach. Current design document is under development at
https://svn.torproject.org/svn/torvm/trunk/doc/design.html

C.2.12 *The Contractor shall continue to develop and implement improvements to the bridge relay and bridge directory authority mechanisms to improve the usability, performance and reliability of the Tor network by users in countries with government-imposed Internet censorship.*

From the Tor 0.2.0.29-rc ChangeLog:
"If you have more than one bridge but don't know their keys, you would only launch a request for the descriptor of the first one on your list. (Tor considered launching requests for the others, but found that it already had a connection on the way for $0000...0000 so it didn't open another.) Bugfix on 0.2.0.x."
"If you have more than one bridge but don't know their keys, and the connection to one of the bridges failed, you would cancel all pending bridge connections. (After all, they all have the same digest.) Bugfix on 0.2.0.x."

C.2.13 *The Contractor shall research and document additional options for the scalability of the Tor* network *beyond 2 million concurrent users, including analysis of splitting the network into multiple segments, switching to datagram-based protocols, and improving the load balancing within the network.*

Many of the proposals detailed in Section C.2.2 are designed to improve scalability, load balancing, directory overhead, and general efficiency. We plan to get most of those proposals implemented in the 0.2.1.x branch, which we hope will become stable by around the end of the year. The next steps are to make sure we have all the plans understood and worked out, so we can make sure there aren't contradictions that we'll encounter down the road when trying to implement them all.

C.2.14 The *Contractor shall continue research into the option of providing incentives for Tor users to run Tor relay servers. If further research indicates that this should be pursued, the Contractor shall develop a project plan and timeline for this work. If further research indicates this option should be abandoned, the Contractor shall document and explain in writing the reasoning behind this decision.*

One of the papers presented at the PET Symposium was entitled "PAR: Payment for Anonymous Routing." It's written by a Columbia University research team, and describes an e-cash mechanism that might be able to provide similar incentives to our earlier design, but get around the limitation where the "special" users stand out and can be tracked. Roger plans to visit their research group in September and start collaborating with them.
http://cs.gmu.edu/~astavrou/research/Par_PET_2008.pdf

C.2.15 *The Contractor shall develop a more reliable download mechanism for the Tor browser bundle for users on slow and/or unreliable network connections, by means of a split download of multiple smaller files, implementation of a lightweight download manager, reduction in the software bundle file size, or other method as chosen by the Contractor.*

We have established our "gettor" email auto-responder script that lets people mail gettor@torproject.org and retrieve a copy of Tor from their mailbox. We still need to ponder more usability issues, such as translation.
https://www.torproject.org/finding-tor

We have also automated the process of checking Tor website mirrors: there's a new update-mirrors.pl script in the website directory that generates a list of mirrors ordered by when they last synced with the main website.
https://www.torproject.org/mirrors

C.2.16 The *Contractor shall test the Tor browser bundle on multiple computer systems and analyze these systems afterwards for any changes to the system that may have been made inadvertently by use of the Tor browser bundle. The Contractor shall document any such changes found and develop a plan to reduce the footprint of Tor browser bundle use.*

We continued evaluating the footprints here:
https://svn.torproject.org/svn/torbrowser/trunk/docs/traces.txt

In particular, we added a new "Registry modifications" section to that file, describing some new traces that appear to be left behind after operating Tor Browser Bundle, even from the USB key. One of the most worrying is the "user assist" registry key that gets set, and (incredible as it sounds) is obfuscated by rot-13 before being set.

C.2.17 *The Contractor shall develop or adapt existing open source software to implement a web-based portal to manage the translations of text into multiple languages for the user interface text of* software *of Torbutton and Vidalia and other software that may in the future be included in the Tor browser bundle. The web site must allow non-technical users the ability to contribute translations by providing text to be translated in English, as well as any needed context on the use of the text, and allowing users to enter the translation into their language from their web browser.*

We have our translation server up and online:
https://translation.torproject.org/

We revised our translation tutorial here:
https://www.torproject.org/translation-portal

Users continued to submit updated translations for many different languages.

We also added the strings for Vidalia's installer; this required writing several scripts to convert from the "nsh" (nullscript installer language) format to the "po" (preferred by Pootle) format and back.

The Tor Project
122 Scott Circle
Dedham, MA 02026 USA
https://www.torproject.org/

From: Roger Dingledine, Tor Project Leader
To: Kelly DeYoe, program officer, BBG
RE: contract BBGCON1807S6441
Date: December 10, 2008

This report documents progress in November 2008 on contract BBGCON1807S6441 between BBG and The Tor Project.

C.2.0 New package releases and related software.

Tor 0.2.1.7-alpha (released November 8) fixes a major security problem in Debian and Ubuntu packages (and maybe other packages) noticed by Theo de Raadt, fixes a smaller security flaw that might allow an attacker to access local services, adds better defense against DNS poisoning attacks on exit relays, further improves hidden service performance, and fixes a variety of other issues.
http://archives.seul.org/or/talk/Nov-2008/msg00229.html

Tor 0.2.0.32 (released November 20) fixes a major security problem in Debian and Ubuntu packages (and maybe other packages) noticed by Theo de Raadt, fixes a smaller security flaw that might allow an attacker to access local services, further improves hidden service performance, and fixes a variety of other issues.
http://archives.seul.org/or/announce/Dec-2008/msg00000.html

Vidalia 0.1.10 (released November 2) fixes some presentation bugs and some bugs in the Windows installer.
http://trac.vidalia-project.net/browser/vidalia/tags/vidalia-0.1.10/CHANGELOG

C.2.1 The Contractor shall continue design, development and implementation of enhancements to the existing Tor software to increase its suitability as a tool for Internet users in countries with government-sponsored Internet censorship to circumvent censorship controls, based on the existing research and documentation performed during the previous contract period (e.g. as described in the paper "Design of a blocking-resistant anonymity system").

We continued enhancements to the Chinese and Russian Tor website translations. Our Farsi translation from this summer is slowly becoming obsolete; we should solve that at some point.

In the Vidalia 0.1.10 stable release:
- Add a prettier dialog for prompting people for their control port password that also includes a checkbox for whether the user wants Vidalia to remember the entered password, a Help button, and a Reset button (Windows only).
- Fix a crash bug that occurred when the user clicks 'Clear' in the message log toolbar followed by 'Save All'.
- Uncheck the Torbutton options by default in the Windows bundle installer if Firefox is not installed.
- Add a Windows bundle installer page that warns the user that they should install Firefox, if it looks like they haven't already done so.

Security fixes in the Tor 0.2.1.7-alpha release:
- The "ClientDNSRejectInternalAddresses" config option wasn't being consistently obeyed: if an exit relay refuses a stream because its exit policy doesn't allow it, we would remember what IP address the relay said the destination address resolves to, even if it's an internal IP address. Bugfix on 0.2.0.7-alpha; patch by rovv.
- The "User" and "Group" config options did not clear the supplementary group entries for the Tor process. The "User" option is now more robust, and we now set the groups to the specified user's primary group. The "Group" option is now ignored. For more detailed logging on credential switching, set CREDENTIAL_LOG_LEVEL in common/compat.c to LOG_NOTICE or higher. Patch by Jacob Appelbaum and Steven Murdoch. Bugfix on 0.0.2pre14. Fixes bug 848.

C.2.2 The Contractor shall submit system architecture and technical design documentation for Tor enhancements specifically related to anti-censorship improvements in C.2.1 to the Authorized Representative of the Contracting Officer (AR/CO) for review and approval before development and implementation. Significant changes to the design that are discovered during development must be documented and reviewed by the AR/CO as soon as the Contractor becomes aware of the need for these revisions.

We have a preliminary proposal that suggests we use only one destination port per circuit. This came out of a discussion between Roger and Robert Hogan about how making an AIM connection through your circuit, and then also web browsing through it, can link the web browsing to your AIM login and you may not want that.
https://svn.torproject.org/svn/tor/trunk/doc/spec/proposals/ideas/xxx-separate-streams-by-port.txt

We picked up the "proposal 141, clients do less directory downloading" design discussion again:
http://archives.seul.org/or/dev/Nov-2008/msg00000.html
http://archives.seul.org/or/dev/Nov-2008/msg00001.html
http://archives.seul.org/or/dev/Nov-2008/msg00007.html
It looks like we have a plausible new direction to go, but nobody to write up the design proposal or implement it. I'm going to do the first go at the next design proposal in January, and hopefully somebody will have time to build it from there.

Worked with Christian Grothoff and his grad student Nate Evans to finish the first draft of their "infinite length circuit" congestion attack. I sent a copy to tor-internal and elsewhere.

C.2.5 The Contractor shall design and develop revisions to the Tor network protocols to hide the network signature of Tor traffic so it is difficult for government-sponsored Internet censors to identify Tor traffic and trivially block it.

Nobody has blocked the new signature, as far as we know.

We have built a plan for how to address potential ways for people to block Tor based on its network signature. We are aiming to have an internal list of known potential vulnerabilities by early 2009, along with suggested paths to addressing each. Then we can react to actual blocking as it occurs, and periodically update our list of potential flaws and intended solutions as we get more intuition.

C.2.10 The Contractor shall promote active growth of the Tor server network and advocacy of Tor products to increase the performance, stability, and usability of Tor.

Andrew started working with Jillian York, so she can start blogging about the great uses of Tor.
https://blog.torproject.org/blog/knight-pulse%2C-jillian%2C-and-tor

Wrote up my notes from the ITSG conference in October, and sent them out to various EFF people, Ken and Kelly, etc.

My 25C3 talk got accepted: "Security and anonymity vulnerabilities in Tor: past, present, and future". They opted not to take the Data Retention talk that I'd offered, because they weren't sure it could fill an entire talk slot. That's ok -- now I can spend 5 minutes of my more general attacks talk and address some of the data retention questions.

On Nov 3, met with Ken Farrall, a grad student at Annenberg at Penn who lived in China for quite a few years and is doing his PhD thesis on comparing dossier-building in China over the past 40 years with dossier-building in the US. According to his research so far, in the 70s, China's society contributed to your dossier much more pervasively: part of your employer's role was to judge how well you're fitting into society (not just how well you're doing your job), and to let you know the answer so you have the opportunity to shape up. In the past decade or so in China, that role has faded, and it's becoming a less pervasive-dossier society at least in that respect. In the US on the other hand, it's only been in the past decade or so that the government has started collecting widespread dossiers on who participates in what groups, who has looked at the websites they deem subversive, etc. I'm greatly oversimplifying, but I found his "the two countries are reversing their roles" analysis intriguing.

I voted on Nov 4, then took the train to NYC to meet with Shiyu Zhou (see separate mail about that).

On Nov 5, I gave a Tor talk at Columbia, based on my Defcon08 talk. I'm doing the same theme at 25C3, so these talks are practice and preparation. I'm starting to think that I should write up an html page explaining each attack or issue clearly, so ordinary users can get a better handle on things. Maybe in 2009, as part of transitioning the FAQ from the wiki to the main website.

I also met with Sambuddho Chakravarty, Angelos's grad student who has been doing the

"approximating a global passive adversary" work. He gave me an updated version of his paper. We talked a lot about the feasibility of the attack, and how to make it more automated and more reproducible. Right now it really is Sambuddho running some scripts and eyeballing the numbers to decide if it worked. If we could make it more automated, then we could start making the attack statistical in nature, and we might have a better handle on what "it doesn't work if the flow is less than 30KB/s sustained" actually means. I sent him a follow-up to see if we can help or to see if he's made any progress.

On Nov 6, I met with HRIC in NY. Shirley Hao (MIT grad) was my contact there, but she was just leaving for some other non-profit in the Bay Area. Charlie McAteer remains, and has a pretty good handle on how Tor works. I also got a chance to demo Vidalia and Tor to Mi Ling Tsui, HRIC's Communications Director, who was really excited to see how simple and intuitive the Torbutton interface was. Alas, we left it like we did last time: "let us know if you know anybody who needs to learn more about this stuff."

Continued working with Andrew to come up with a version of the roadmap that sufficiently anonymizes the sponsors who need that, and a version of the press release that is sufficiently vague about the parts that the sponsors want kept vague. Continued wrestling with said funders about whether they really need the secrecy; looks like the outcome will be 'yes'.

Continued working with Chris Walker to refine our list of Sesawe deliverables for the Sept to Feb timeframe. Yes, it sure is a long way from September by now. You'd think we could wrap up the negotiations and get some contracts in place. December^WJanuary, we hope.

We got an invitation to an "info-activism" training week in India in February:
http://www.informationactivism.org/
Alas, it overlaps with FC so I can't make it. We are sending Jake though.

Started the process of putting our website and docs under the CC-BY license:
http://creativecommons.org/licenses/by/3.0/us/

Alas, I haven't actually done the final step yet of changing the website footer. If somebody else wants to do that, please feel free. Else it's on my (already overfull) January todo list.

On Nov 13 I did a talk at Xerox PARC, since Philippe Golle has been bugging me for multiple years now to come do a talk.

On Nov 13 Jake and I met with Dave Jevans and Steve Southam of Ironkey. Ironkey sells a very secure USB token that comes with Tor and a few other programs built-in. They run their own entirely separate Tor network with about 30 relays in a half dozen places around the world. We learned more about their operations, and I think also convinced them that handling their own Tor network and maintaining a private Tor branch required more effort than they realized. Steve is devoted full-time to their Tor network, but he really didn't seem up on the latest Tor vulnerabilities and issues. Dave (their CEO) actually seemed to have much higher clue about the technical issues. Perhaps he will now put more energy into making sure they get their Tor stuff right. One interesting note is that they do their authentication from the Tor client (via hardware token) to the exit node -- and the exit node doesn't let them exit unless they authenticate first. So

a) I could sign up all the ironkey nodes on the public Tor network right now, mark them as badexit, and we could freeload off their network (but that wouldn't be very nice), and b) even if they do get their undocumented group key signature scheme right, so they can't distinguish one user from another, it still makes me nervous that their design bypasses the entry and middle hop when doing their authentication. Weren't we supposed to do three hops to distribute trust?

On Nov 14 I did a talk at eBay for their security / malware analysis group. They started out skeptical about the value of Tor (well, in particular, the group's manager did) but after my 90 minute talk he had totally bought it and told us how great it was we were doing what we're doing. It turns out that eBay's malware analysis group already uses Tor for looking at malware-related websites without revealing that it's eBay doing the looking. I asked Chris Paget (our host) if we can tell the outside world that eBay uses Tor, and he was going to check with their PR people (no word yet -- in retrospect I should not have asked). Chris was also going to put some entries into their 2009 budget to either a) get some Tor consulting about how to use Tor well for their situations, or b) run a non-exit relay.

On Nov 14 I visited EFF. We had a long chat about whether EFF should be endorsing any other anonymity system besides Tor and Anonymizer.com. They have apparently been getting a lot of requests lately from apparently snakeoil groups that want EFF to give our free trial versions of their snakeoil tool. In the discussion, we learned that Anonymizer.com has been bought by what appears to be a defense contractor that specializes in data mining (!). That was the point where EFF started to ask if it should drop its endorsement of Anonymizer. I don't know how that has turned out.

Jake and I failed to go to the circumvention book sprint that Laurent et al organized in New York, since I was in the Bay Area that week. But they produced a fine start at a book:
http://en.flossmanuals.net/CircumventionTools/

Continued brainstorming with Jake and Andrew about metrics we might use for a "node manager" position as part of the Sesawe project. We want to take some money so we can pay Jake to focus more on relay advocacy and support; but they want to know exactly how successful we will be before they'll give us the money. Work on this item continues.

Continued talking to Aaron Swartz and Virgil Griffith about their tor2web design, and various design questions like "can we configure our Firefox to only proxy .onion urls?"
http://tor2web.org/

Wendy Seltzer and Isaac Mao participated in the Chinese Blogger conference:
http://www.cnbloggercon.org/2008/en

Jake and I attended Gunner's "Nonprofit Software Development Summit" November 17-19:
http://devsummit08.aspiriationtech.org/index.php/Main_Page
I talked to a lot of people about Tor. Useful contacts include:
- Ariel Glenn, Wikimedia. I talked to her about my "Wikipedia should put up more roadblocks for IP addresses they hate, not just make the whole world black or white" idea. She liked it. I should write it up someday, but I knew that.
- Nathan "Dorjee", who helps run Students for a Free Tibet in NYC. I'm going to try to get to

NYC in Januaryish and do a more focused talk for his group, now that we've actually met each other and he is willing to answer my emails.
- Adam Hyde, flossmanuals. He led the circumvention book sprint.
- Allen Gunn. He organized this conference, and also helps run the every-few-years Summersource conferences that I never seem to make it to. He knows a lot of people, and having him like Tor is good.
- A big pile of San Francisco area people who ought to be running Tor relays if only Jake could lean on them enough. Working on that.

On Nov 24, I met with Rob D'Ovidio, a professor at Drexel who teaches criminal justice.
http://www.drexel.edu/coas/culturecomm/faculty/dovidio.asp
http://www.pages.drexel.edu/~rd64/RDHome.html
I gave him the first half of my "standard Tor talk", and I think convinced him that I had interesting things to say. Rob has lots of contact with the local-area "high technology crime" groups, and will hopefully be able to get me hooked up with some of them so I can teach them more about our perspective on high technology crime.

Followed up on the thread from Daniel Brandt at Scroogle. Apparently they were experiencing some jerk crawling Google via Scroogle via Tor, and they had to block Tor access until they had a handle on what was happening. They eventually solved it by noting that the jerk was searching for the same thing over and over, so they could just block by search query. Hopefully I've educated them enough that if something happens in the future, a) they will block more intelligently, and b) they will consider contacting Tor if they have continued problems.

C.2.11 *The Contractor shall improve the ease of use of Tor for end users by continuing research and development on one or both of the following products: (1) all-in-one software bundle containing Tor and supporting applications, as well as an easy-to-use installer for Microsoft Windows operating systems, as well as option to install and run from a Universal Serial Bus (USB) flash device; (2) bootable CD-ROM image ("LiveCD") which contains a minimal operating system, Tor, and supporting applications. Both would have all appropriate applications pre-configured to use Tor out of the box with only minimal additional configuration required by the end user. If Contractor determines it is not feasible to develop both products, Contractor will provide detailed written technical analysis and explanation to the AR/CO. The Contractor shall make an initial public release of at least one implementation of one of these products during the term of this contract.*

Work by Martin Peck and Kyle Williams on the Tor VM project continues. We have a working prototype available now with a walk-through and screenshots:
http://peertech.org/files/demo/testinfo.html

We also released an early test version:
http://www.janusvm.com/tor_vm/

C.2.12 *The Contractor shall continue to develop and implement improvements to the bridge relay and bridge directory authority mechanisms to improve the usability, performance and reliability of the Tor network by users in countries with government-imposed Internet censorship.*

The number of active bridge relays is still going up: from 94 Running bridges at the end of

September to 105 Running bridges at the end of October to 135 Running bridges at the end of November.

C.2.13 *The Contractor shall research and document additional options for the scalability of the Tor* network *beyond 2 million concurrent users, including analysis of splitting the network into multiple segments, switching to datagram-based protocols, and improving the load balancing within the network.*

Started talking to Csaba Kiraly, a grad student in Italy who is working on measurements to improve Tor's performance. He's been focusing on flow control (Tor's circuit and stream windows), and has some great suggestions about reducing the congestion in the network: by *decreasing* our circuit window, we will (counterintuitively) reduce the number of outstanding cells stuck waiting in the network. He's writing it up as a real research paper, and hopefully we'll see something public soon.

C.2.14 The *Contractor shall continue research into the option of providing incentives for Tor users to run Tor relay servers. If further research indicates that this should be pursued, the Contractor shall develop a project plan and timeline for this work. If further research indicates this option should be abandoned, the Contractor shall document and explain in writing the reasoning behind this decision.*

In November, Roger met with the Columbia University research group that's been working on a related project called PAR.
http://freehaven.net/anonbib/#raykova-pet2008

I tried to push them into a much more simple design. The one I had in mind was the gold-star design except authorities give out multi-use ecash tokens. For example, for being a good relay, you get 7 tokens, one for each of the next seven days, that work as many times as you like on the designated day. You use your token to prove to the entry node that you're cool, and then it uses its token to prove to the next hop that it's cool, and so on. (You don't really know who has a token, but you can guess that most good relays do.) As a bonus, now you can earn tokens with your fast colo server, and use the tokens from the comfort of your own home. One unsolved problem here is that somebody can put their token on slashdot, and then everything goes to hell for that day. (Messier scenarios involve somebody putting out a competing Tor client where one token is sent to all those clients for use each day.) We could imagine a mechanism by which the community of relays notices when a given token is too widely used; that approach means you need to show the token to the entry guard though, rather than doing some cooler "proof I have a valid token but no you don't get to see it" trick. Or we could just suck it up and hope nobody wants to repeatedly ruin the network like this :), and if they do then it's time for something heavier-powered like PAR. Alas, while the grad students there don't object to simpler designs in theory, the realities of their constraints mean that the complex confusing ones are the best candidates for research papers.

After the PAR meeting, Johnny Ngan and Dan Wallach and I finished the last draft of our incentives.pdf paper, and tech reported it:
http://seclab.cs.rice.edu/lab/publications/
My next step, sometime in January, is to put up a blog entry on it, and let or-talk and others know about it.

C.2.15 *The Contractor shall develop a more reliable download mechanism for the Tor browser bundle for users on slow and/or unreliable network connections, by means of a split download of multiple smaller files, implementation of a lightweight download manager, reduction in the software bundle file size, or other method as chosen by the Contractor.*

Started the process of switching http://tor.eff.org/ from a redirect into a full mirror. Having more mirrors is good, especially if we're getting blocked in certain countries. Need to keep following up on this.

We continued extensive work on Thandy this month.

We have a Thandy repository up at
http://updates.torproject.org/thandy/
and its keys and location ship with the thandy client.

(The current repository is still for testing only, and we'll discard the keys and generate new ones when we want to put it up for real. We'll also get an ssl cert for it and hopefully put it on a more secure host.)

The client-side of Thandy (teaching it how to decide which packages and bundles are out of date, and teaching it to download new files and check all the right signatures) exists now too. It supports download resuming, doing the download over Tor, etc.

The big picture is that thandy will remember what versions of each package and bundle are installed. Vidalia will periodically launch thandy-client so it can check for updates. When there are new packages, thandy will tell Vidalia (via stdout currently, since Vidalia launched it). Then when the time is right, Vidalia will launch thandy-client with a --install option, and thandy will know how to run the installers for each type of package (currently "rpm", "win32", and "none" are supported):
https://svn.torproject.org/svn/updater/trunk/doc/interface.txt

The long-term plan is to have every platform have a package system that is capable of answering "What version of the software is installed?" On Windows, that would either be the new MSI installer file we're working on:
https://svn.vidalia-project.net/svn/vidalia/trunk/pkg/win32/vidalia.wxs.in
or our current NSI installer, with a new registry key patch we're working on.

If an upgrade attempt fails (due to a broken package, broken system, sudden power loss, etc), thandy will try again the next time you tell it to install. With luck, it will work later, or an upgraded version of the package that _does_ work will come to be, and thandy will fetch and install that one instead.

We're working on patching our current Windows installer so it knows how to answer what version is installed. Then it will be easier for all the components to work together.

In short: many more components of our auto updater are coming together, but they aren't all

together yet.

C.2.16 The *Contractor shall test the Tor browser bundle on multiple computer systems and analyze these systems afterwards for any changes to the system that may have been made inadvertently by use of the Tor browser bundle. The Contractor shall document any such changes found and develop a plan to reduce the footprint of Tor browser bundle use.*

No changes.

We've started to think about moving the Tor Browser Bundle from Firefox 2 to Firefox 3. This will mean we should measure new traces. We'll do it once Torbutton is known to be more stable on Firefox 3, which should happen in early 2009 once we bring Mike Perry on board.

C.2.17 *The Contractor shall develop or adapt existing open source software to implement a web-based portal to manage the translations of text into multiple languages for the user interface text of* software *of Torbutton and Vidalia and other software that may in the future be included in the Tor browser bundle. The web site must allow non-technical users the ability to contribute translations by providing text to be translated in English, as well as any needed context on the use of the text, and allowing users to enter the translation into their language from their web browser.*

We have our translation server up and online:
https://translation.torproject.org/
https://www.torproject.org/translation-portal
We now have a Romanian translation.

The Tor Project
122 Scott Circle
Dedham, MA 02026 USA
https://www.torproject.org/

From: Roger Dingledine, Tor Project Leader
To: Kelly DeYoe, program officer, BBG
RE: contract BBGCON1807S6441
Date: November 10, 2008

This report documents progress in October 2008 on contract BBGCON1807S6441 between BBG and The Tor Project.

C.2.0 New package releases and related software.

No new development releases month. We have been focusing on getting the upcoming November releases to be good.

C.2.1 The Contractor shall continue design, development and implementation of enhancements to the existing Tor software to increase its suitability as a tool for Internet users in countries with government-sponsored Internet censorship to circumvent censorship controls, based on the existing research and documentation performed during the previous contract period (e.g. as described in the paper "Design of a blocking-resistant anonymity system").

We continued enhancements to the Chinese and Russian Tor website translations. We also have a second Chinese translator for the website now, so hopefully we will get more prompt translations there. Our Farsi translation from this summer is slowly becoming obsolete; we should solve that at some point.

We added a new "30 second summary" web page for Tor:
https://www.torproject.org/30seconds
and a new "easy download" page since the original is so complex:
https://www.torproject.org/easy-download

In the upcoming Vidalia 0.2.0 development release:
- Support changing UI languages without having to restart Vidalia.
- Updated Czech, Polish, Romanian and Turkish translations.

In the upcoming Vidalia 0.1.10 stable release:
- Add a prettier dialog for prompting people for their control port password that also includes a checkbox for whether the user wants Vidalia to remember the entered password, a Help button,

and a Reset button (Windows only).
- Fix a crash bug that occurred when the user clicks 'Clear' in the message log toolbar followed by 'Save All'.
- Uncheck the Torbutton options by default in the Windows bundle installer if Firefox is not installed.
- Add an Windows bundle installer page that warns the user that they should install Firefox, if it looks like they haven't already done so.

It looks like Australia is soon to be joining the ranks of countries with a nationwide filtering regime:
http://arstechnica.com/news.ars/post/20081016-net-filters-required-for-all-australians-no-opt-out.html

C.2.2 The Contractor shall submit system architecture and technical design documentation for Tor enhancements specifically related to anti-censorship improvements in C.2.1 to the Authorized Representative of the Contracting Officer (AR/CO) for review and approval before development and implementation. Significant changes to the design that are discovered during development must be documented and reviewed by the AR/CO as soon as the Contractor becomes aware of the need for these revisions.

We finished the first iteration of our auto-updater spec:
https://svn.torproject.org/svn/updater/trunk/specs/thandy-spec.txt
We detail our current auto-updater progress in Section C.2.15 below.

Proposal 156 (Tracking blocked ports on the client side) moves us closer to having clients be able to automatically detect which ports are blocked by their local firewall, so they can bootstrap faster and avoid picking entry guards that aren't reachable for them. The the next steps here are to a) decide if this overall approach is the right approach, and b) revise the patch to be more memory-friendly.
https://svn.torproject.org/svn/tor/trunk/doc/spec/proposals/156-tracking-blocked-ports.txt

C.2.5 The Contractor shall design and develop revisions to the Tor network protocols to hide the network signature of Tor traffic so it is difficult for government-sponsored Internet censors to identify Tor traffic and trivially block it.

Nobody has blocked the new signature, as far as we know.

We have built a plan for how to address potential ways for people to block Tor based on its network signature. We are aiming to have an internal list of known potential vulnerabilities by early 2009, along with suggested paths to addressing each. Then we can react to actual blocking as it occurs, and periodically update our list of potential flaws and intended solutions as we get more intuition.

C.2.10 The Contractor shall promote active growth of the Tor server network and advocacy of Tor products to increase the performance, stability, and usability of Tor.

Roger finally sent a follow-up mail to "Amir", the Iranian fellow in Norway whom Roger met in February. It looks like that lead might be too old to successfully follow though. Oops.

We followed up with Keiji Takeda, the Japanese professor that Joi Ito passed us to with respect to running a Tor directory authority in Japan. My current conclusion is that if nobody there is excited enough to answer our mails consistently, then it's a bad location for a directory authority.

Started brainstorming with Jake and Andrew about metrics we might use for a "node manager" as part of the Sesawe (formerly iFree) project. We want to take some money so we can pay Jake to focus more on relay advocacy and support; but they want to know exactly how successfully we will be before they'll give us the money. Work on this item continues into November (and probably beyond).

Answered questions from Rafal (of Psiphon) about how much he should pay good engineers: my recommendation was that you actually have to pay them well, or you won't get any good engineers. Maybe he'll take it to heart, and Psiphon will end up with some security clue for their next iteration.

Roger started a "Brainstorming about Tor, Germany, and data retention" thread on or-dev:
http://archives.seul.org/or/dev/Oct-2008/msg00001.html
which eventually turned into a blog post:
https://blog.torproject.org/blog/tor%2C-germany%2C-and-data-retention
as well as a (rejected) 25C3 submission. While I had originally been thinking of the issue in terms of what the ISP of a Tor relay might do, the discussion also came up about what responsibilities a Tor relay operator has with respect to the vague new data retention laws:
http://archives.seul.org/or/talk/Oct-2008/threads.html#00126
The ultimate result was a clarified perspective on logging inside Tor:
http://archives.seul.org/or/talk/Oct-2008/msg00274.html

We finally tracked down and solved the mysterious DoS attacks on some of the Tor directory authorities:
http://archives.seul.org/or/talk/Oct-2008/msg00056.html

Roger agreed to help Christian Grothoff and his grad student to flesh out their "infinite length circuit attack" paper and defenses. My goal is to help get the attack details and numbers written down clearly, so we will have a headstart on understanding how bad it is and how much we need to fix. More on that in November.

Roger contacted James Mulvenon at cira-dc.com, as Ken Berman had suggested I chat with him. No answer yet. Oh well.

We started chatting with Aaron Swartz about his "tor2web" proxy idea for letting non-Tor users access hidden service content:
http://tor.theinfo.org/
Somebody should follow up on that more to encourage him to keep at it.

Announced Joel Reardon's thesis on or-talk, and followed up with him to point him to some pieces of anonbib he needs to read more, to tell him about 25C3, and to remind him to publish his new measurement tools lest they become lost to time. We'll see how that goes.

Roger and Karsten got the patches from proposal 155 into svn, and ultimately into the upcoming 0.2.1.7-alpha release. These were the bulk of the October progress for that NLnet project:
https://www.torproject.org/projects/hidserv.html.en#Oct08

Roger finally answered Ira Rubinstein, a law professor at NYU, about all his questions about "Anonymity and Accountability on the Internet". Tried to explain that the DDoS traceback mechanisms worked on a different level than Tor, so they don't actually threaten us. It all gets murky when you tie it into "network identity" too. Ira since invited me to come drop by his lecture in early November; but I bowed out due to time pressure. Oh well.

Mike deleted the router-stability file for his directory authority (ides), which should provide temporary relief from bug 696 (which was causing most of the Stable flags to be assigned wrong, and in turn was causing instant messaging and related connections over Tor to be way more flaky than they should be):
https://bugs.torproject.org/flyspray/index.php?do=details&id=696
If his router-stability file gets corrupted again, we will have learned something.

Roger helped Philippe Golle handle the incoming Financial Crypto submissions, and assign reviews to our program committee. We got 102 submissions in total, which was quite a few more than we expected.
http://fc09.ifca.ai/

Roger went to a talk at Penn's Annenberg school, on how the FCC screwed up by claiming jurisdiction to smack down Comcast when it didn't really have the jurisdiction. Conclusion was that the courts will overturn the FCC's plan, and in the meantime nobody in Congress will act "since the FCC is clearly taking care of everything". Lose-lose. Met Christopher Yoo, a law professor there who is a self-proclaimed opponent of net neutrality. He seems like a pretty reasonable guy, all in all.

Roger met with Jeremiah Young of CDHR on Oct 21; see separate mail about that.

Roger met with a group of law enforcement agents on Oct 22-23 to discuss online anonymity and how it could be useful to them. See separate mail for details.

Roger, Jacob, and Mike went to the Google Summer of Code Mentor Summit on Oct 24-26 in Mountain View, where we met with a few hundred other GSoC mentors and generally shared information about Tor and how to make good use of summer students working on free software tools.

We also went to dinner with Niels Provos while we were there, to talk about options for the "Google gives you a captcha if you're using Tor" problem. It looks like the right answer there will be for Torbutton to automate some workaround. Once Mike joins us, we can work on that harder.

Roger and Nick attended the ACM CCS academic security conference in Alexandria, VA on Oct 27-30.

Roger then visited IBB on Oct 31 to discuss the roadmap further and keep everybody up to date.

Had an extended IRC chat with Nart Villeneuve on Oct 31 about next directions for Psiphon. He's working to try to get them to commit to a set of security properties they want, so he can document them and they can get started deciding what designs will or won't achieve them. It looks like they don't want to give up on any security properties yet, which makes it really hard to plan any designs for the new Psiphon. He also didn't have any good explanation for why Psiphon is a for-profit company. More news here later on I hope.

Helped Andrew work on a press release for our iFree/Sesawe work. The press release still hasn't happened, because other groups in the plan are still trying to figure out how little press they can get away with. Hopefully something will happen in December!

Peter Eckersley came up with an attack on our defense for proposal 110:
https://svn.torproject.org/svn/tor/trunk/doc/spec/proposals/110-avoid-infinite-circuits.txt
I should write that up in more detail so we can think about how to fix it.

Andrew started working with Jillian York, so she can start blogging about the great uses of Tor. More news in November, e.g.
https://blog.torproject.org/blog/knight-pulse%2C-jillian%2C-and-tor

Matt Edman printed Vidalia T-shirts, and sent them out to the folks who have helped work on Vidalia lately. He is also working with a volunteer to clean up the Vidalia website, make new logos, clean up the installer graphics, etc.

Jacob and Andrew worked with Bene Cipolla at the US State Dept to help her understand Tor, censorship, circumvention, etc. She is working on a video series including "How to Circumvent an Internet Proxy".

Andrew wrote a blog post about anonymity in South Korea:
https://blog.torproject.org/blog/online-anonymity-debate-south-korea

Steven and Andrew had a phone meeting with Article 19:
http://www.article19.org/
to introduce them to Tor and see if it makes sense to work together more closely.

C.2.11 *The Contractor shall improve the ease of use of Tor for end users by continuing research and development on one or both of the following products: (1) all-in-one software bundle containing Tor and supporting applications, as well as an easy-to-use installer for Microsoft Windows operating systems, as well as option to install and run from a Universal Serial Bus (USB) flash device; (2) bootable CD-ROM image ("LiveCD") which contains a minimal operating system, Tor, and supporting applications. Both would have all appropriate applications pre-configured to use Tor out of the box with only minimal additional configuration required by the end user. If Contractor determines it is not feasible to develop both products, Contractor will provide detailed written technical analysis and explanation to the AR/CO. The Contractor shall make an initial public release of at least one implementation of one of these products during the term of this contract.*

Work by Martin Peck and Kyle Williams on the Tor VM project continues. We have a working prototype available now with a walk-through and screenshots:
http://peertech.org/files/demo/testinfo.html

We plan to release a more public alpha installer in November.

C.2.12 *The Contractor shall continue to develop and implement improvements to the bridge relay and bridge directory authority mechanisms to improve the usability, performance and reliability of the Tor network by users in countries with government-imposed Internet censorship.*

The number of active bridge relays is still going up: from 94 Running bridges at the end of September to 105 Running bridges at the end of October.

C.2.13 *The Contractor shall research and document additional options for the scalability of the Tor* network *beyond 2 million concurrent users, including analysis of splitting the network into multiple segments, switching to datagram-based protocols, and improving the load balancing within the network.*

From the Tor 0.2.1.7-alpha ChangeLog:
"The "ClientDNSRejectInternalAddresses" config option wasn't being consistently obeyed: if an exit relay refuses a stream because its exit policy doesn't allow it, we would remember what IP address the relay said the destination address resolves to, even if it's an internal IP address. Bugfix on 0.2.0.7-alpha; patch by rovv."

Peter Eckersley fixed a few more bugs in Tor Weather:
https://weather.torproject.org/
It's getting closer to being stable. When it is, it will be an integral part of maintaining a healthy network, since right now a lot of relays disappear because their operators don't even know they're down.

C.2.14 The *Contractor shall continue research into the option of providing incentives for Tor users to run Tor relay servers. If further research indicates that this should be pursued, the Contractor shall develop a project plan and timeline for this work. If further research indicates this option should be abandoned, the Contractor shall document and explain in writing the reasoning behind this decision.*

In November, Roger plans to meet with the Columbia University research group that's been working on a related project. We're also hoping to get a public tech report for the incentives.pdf design out in November.

C.2.15 *The Contractor shall develop a more reliable download mechanism for the Tor browser bundle for users on slow and/or unreliable network connections, by means of a split download of multiple smaller files, implementation of a lightweight download manager, reduction in the software bundle file size, or other method as chosen by the Contractor.*

We changed our auto update design from code-name Glider to code-name Thandy, since there's a World of Warcraft cheat program named Glider and it might be a problem for WoW players that

try to use Tor.

We've got the PKI and server-side for the auto updater in place. We wrote up a howto walking through how to set up the server-side for the updater, including how to assign roles and generate keys:
https://svn.torproject.org/svn/updater/trunk/doc/HOWTO

We've also decided that Python should work fine for the client-side too. Mike Perry found some techniques to include only exactly the python libs we need, rather than the whole mess of python libs:
http://www.py2exe.org/index.cgi/BetterCompression
and Martin Peck has been messing with saving some additional space by sharing the openssl lib between Tor and Thandy.

The next steps for November are:
- Roger is going to figure out what PKI we want for the first round of testing (what roles, which keys, how many, who, etc), and deploy a Thandy server so we can put some basic packages on it for testing.
- Nick is going to finish the client-side of Thandy, in terms of teaching it how to decide which packages and bundles are out of date, and teaching it to download new files and check all the right signatures.
- Martin is going to package Thandy plus all the right python libs in an easy Windows exe that hopefully isn't too big.
- Matt Edman is going to add a simple interface to Vidalia for client-side Thandy configuration: stuff like a GUI for telling the user that new updates have appeared and letting the user click "yes, please update me now", etc.
- Nick and Matt are going to brainstorm more about the interface between Vidalia and Thandy. For example, which program should keep state about the versions of each package that are installed, which program should be responsible for noticing if an install or upgrade attempt fails, etc.

All the steps but the last I think are going to be pretty straightforward. This last step has the most potential pitfalls in it, since we're trying to keep Thandy general and platform-independent yet *something* (either Thandy or Vidalia, or something in between) has to tackle all the crazy Windows-specific pieces.

It also looks like we should move the Tor packages and bundles from NSIS (Nullsoft installer) to MSI installer, as MSI can handle versioning and automatic installs (and uninstalls!) more gracefully. It's not yet clear yet if we're going to try to squeeze that installer shift into the November development timeframe.

C.2.16 The *Contractor shall test the Tor browser bundle on multiple computer systems and analyze these systems afterwards for any changes to the system that may have been made inadvertently by use of the Tor browser bundle. The Contractor shall document any such changes found and develop a plan to reduce the footprint of Tor browser bundle use.*

No changes.

We've started to think about moving the Tor Browser Bundle from Firefox 2 to Firefox 3. This will mean we should measure new traces. We'll do it once Torbutton is known to be more stable on Firefox 3, which should happen in early 2009 once we bring Mike Perry on board.

C.2.17 *The Contractor shall develop or adapt existing open source software to implement a web-based portal to manage the translations of text into multiple languages for the user interface text of* software *of Torbutton and Vidalia and other software that may in the future be included in the Tor browser bundle. The web site must allow non-technical users the ability to contribute translations by providing text to be translated in English, as well as any needed context on the use of the text, and allowing users to enter the translation into their language from their web browser.*

We have our translation server up and online:
https://translation.torproject.org/
https://www.torproject.org/translation-portal

Users continued to submit updated translations for many different languages.

The Tor Project
122 Scott Circle
Dedham, MA 02026 USA
https://www.torproject.org/

From: Roger Dingledine, Tor Project Leader
To: Kelly DeYoe, program officer, BBG
RE: contract BBGCON1807S6441
Date: October 10, 2008

This report documents progress in September 2008 on contract BBGCON1807S6441 between BBG and The Tor Project.

C.2.0 New package releases and related software.

Vidalia 0.1.9 (released September 2) fixes a big pile of bugs and inconveniences in the earlier releases. This new release marks the first "stable" release of Vidalia, in that we have now branched into a stable (0.1.x) branch and a development (0.2.x) branch.
http://trac.vidalia-project.net/browser/vidalia/tags/vidalia-0.1.9/CHANGELOG

Tor 0.2.0.31 (released September 3) addresses two potential anonymity issues, starts to fix a big bug we're seeing where in rare cases traffic from one Tor stream gets mixed into another stream, and fixes a variety of smaller issues.
http://archives.seul.org/or/announce/Sep-2008/msg00000.html

Tor 0.2.1.6-alpha (released September 30) further improves performance and robustness of hidden services, starts work on supporting per-country relay selection, and fixes a variety of smaller issues.
http://archives.seul.org/or/talk/Oct-2008/msg00093.html

C.2.1 The Contractor shall continue design, development and implementation of enhancements to the existing Tor software to increase its suitability as a tool for Internet users in countries with government-sponsored Internet censorship to circumvent censorship controls, based on the existing research and documentation performed during the previous contract period (e.g. as described in the paper "Design of a blocking-resistant anonymity system").

We continued enhancements to the Chinese and Russian Tor website translations.

From the Vidalia 0.1.9 ChangeLog:
"Correct the location of the simplified Chinese help files so they will actually load again."

From the Tor 0.2.1.6-alpha ChangeLog:
"Start work to allow node restrictions to include country codes. The syntax to exclude nodes in a country with country code XX is "ExcludeNodes {XX}". Patch from Robert Hogan. It still needs some refinement to decide what config options should take priority if you ask to both use a particular node and exclude it."
This feature should allow users in China to specify that they don't want to enter (and/or exit) in China, which in theory could provide stronger security for them.

From the Tor 0.2.1.6-alpha ChangeLog:
"Allow ports 465 and 587 in the default exit policy again. We had rejected them in 0.1.0.15, because back in 2005 they were commonly misconfigured and ended up as spam targets. We hear they are better locked down these days."
This feature lets people use GMail with Tor in more flexible ways. This approach is especially important for people trying to send email in certain configurations when their network wants to block or monitor them.

From the Tor 0.2.1.6-alpha ChangeLog:
"Provide circuit purposes along with circuit events to the controller."
This change will allow Vidalia to mark circuits in its graphical interface, so users don't get confused about why Tor is building strange circuits in the background when it's really just doing encrypted directory updates.

Matt and Andrew fixed a bug in the Vidalia bundle installer where it tried to detect if Firefox was installed, and unclick the "install Torbutton" option if not, but it didn't detect right. Now if Firefox is missing we put up a warning explanation about how you really ought to be using Tor with Firefox.

We also finally started working on a fix for the Vidalia bug where if Vidalia launches Tor and then crashes later, when you start Vidalia again it'll cryptically ask for your control password.
https://wiki.torproject.org/noreply/TheOnionRouter/TorFAQ#TorPasswordPrompt
The first fix is to add a "reset" button to the cryptic message, that kills Tor for you and restarts it, and a "help" button that explains what's going on. These will be out in the next development Vidalia release, hopefully in October.

Camilo Viecco submitted a patch for our RPM spec (build) file to let us build Red Hat / SuSE packages for 64-bit architectures. Andrew included these patches in 0.2.1.6-alpha.

C.2.2 The Contractor shall submit system architecture and technical design documentation for Tor enhancements specifically related to anti-censorship improvements in C.2.1 to the Authorized Representative of the Contracting Officer (AR/CO) for review and approval before development and implementation. Significant changes to the design that are discovered during development must be documented and reviewed by the AR/CO as soon as the Contractor becomes aware of the need for these revisions.

Karsten has continued work on performance and robustness improvements for hidden services. His latest design proposal involves addressing four known flaws in the current design. We plan to address them in October.

https://svn.torproject.org/svn/tor/trunk/doc/spec/proposals/155-four-hidden-service-improvements.txt

Roger spent much of September working on the new comprehensive development roadmap (sent September 17), with deliverables and milestones and lead developers. The document aims to give our various funders a sense of where they fit in and where Tor is going. The next step there for October is to build a plan for how to share the roadmap-full pdf with the wider Tor community -- and all that telling people about our new funding and funders ends up entailing.

C.2.5 The Contractor shall design and develop revisions to the Tor network protocols to hide the network signature of Tor traffic so it is difficult for government-sponsored Internet censors to identify Tor traffic and trivially block it.

Nobody has blocked the new signature, as far as we know.

We have built a plan for how to address potential ways for people to block Tor based on its network signature. We are aiming to have an internal list of known potential vulnerabilities by early 2009, along with suggested paths to addressing each. Then we can react to actual blocking as it occurs, and periodically update our list of potential flaws and intended solutions as we get more intuition.

C.2.10 The Contractor shall promote active growth of the Tor server network and advocacy of Tor products to increase the performance, stability, and usability of Tor.

Steven Murdoch taught a lecture at the FIDIS/IFIP Brno Summer School in the Czech Republic.
http://www.buslab.org/SummerSchool2008/
The presentation was on anti-censorship in general especially on Tor. The students seemed to be interested so he encouraged them to look at Tor and see if there is anything they'd like to work on. We will see if anything comes from that.

We've also been discussing creating a Facebook application, for allowing relay operators to show off that they are running a Tor relay and hopefully encourage more to do so. We think this is a good enough idea to try building it, so Steven has started to do so. As well as adding bling to a user's profile, it would also allow us to map the network of node operators. This is one of the more promising research fields to resist Sybil attacks, see e.g.
"A Sybil-proof one-hop DHT, Chris Lesniewski-Laas"
http://pdos.csail.mit.edu/papers/sybil-dht-socialnets08.pdf

Anonym, the Incognito developer, heard rumors of an Incognito-related arrest in Shanghai. We passed the details on to Isaac Mao for confirmation or investigation. Isaac thinks it isn't a real issue at least until we hear a name or anything else that we can check out.

Jacob discussed running a Tor node with the FSF. It took some time for this to materialize. "The key to making this happen was easy. I simply did all of the heavy lifting, I wrote an email with a sample configuration file and the methods for installing the packages - it resulted in a new high bandwidth Tor server. Great news! Hopefully we'll be able to convince them to start offering a mirror of ftp.gnu.org over a hidden service. ;-)"

Jacob wrote up a summary of experiences deploying Tor in an area where circumvention and bridges were needed. It was an intentionally vague summary, to protect those involved.

Steven had a related story regarding host-based security from his trainings in Kyrgyzstan and Poland. See also
http://www.f-secure.com/weblog/archives/00001494.html

Jacob was in a story by Declan about Internet Traceback plans:
"The Chinese Government, the NSA, Verisign and the ITU are getting together to trace users"
http://news.cnet.com/8301-13578_3-10040152-38.html

The current issue of Make Magazine has an article on how to use Tor:
http://www.make-digital.com/make/vol15/?pg=102

Roger and Andrew continued to work to produce a 2009 budget that can squeeze in Mike, Andrew, Karsten, and Martin even if we only get iFree and IBB funding. It'll be tight but I think it'll work; then if other funding comes in too it'll be a bit more comfortable. The board won't officially vote on a 2009 budget until Decemberish.

Roger and Andrew visited IBB and iFree in DC in mid September, including attending the official iFree launch.

We started the process of working with Psiphon to try to get useful stuff on their iFree development roadmap, and to see where our efforts I overlap. I may be heading to Ottawa or Toronto in November or December to discuss with them further.

I worked with Eric Johnson to help him refine the list of filtering survey questions we're planning to ask in-country experts. My main addition was a question about privacy while circumventing: "if users are concerned that somebody will notice that they're circumventing, does it matter to them whether somebody watching their Internet knows what site they're going to? Does it matter to them whether the folks providing the circumvention tool can know this?"

I also helped point Eric at the various in-country experts we know, to jumpstart him into getting the right contacts: Isaac Mao, Andrew Lih, Rebecca MacKinnon, Helmi Noman.

Answered Eric's question about how Tor + GTunnel could possibly make sense together:
http://www.internetfreedom.org/GTunnel
By funnelling all the Tor connections back to the GTunnel central servers at the end, they get rid of the "exit relays can read your plaintext" issue, but they introduce an anonymity bottleneck.

Gave Eric a big pile of "non-technical things iFree could help us with", since they're excited to hear about useful things they could put their effort into.

Frank Rieger, Karsten, and I brainstormed about what research and development items need to

happen next for hidden services. We're hoping to find some more funding to keep Karsten working on this topic in 2009 and beyond.

Roger, Jacob, and Mike are on track to attend the Google Summer of Code Mentor Summit in Mountain View at the end of October.

Roger and Andrew talked to Ali Alyami, the exec director of cdhr.info. Ali is focused on Saudi Arabia. Roger is going to meet with one of their tech people in San Diego in October.

Roger is also going to be on a panel for law enforcement at an FBI conference in San Diego at the end of October. Hopefully this step will help us get the word out to a wider audience, and also prepare the officers here for the concept that Internet security could be useful to them too.

Tried to find other contacts in Saudi Arabia. They're one of the countries that iFree is going to tackle in year one, but nobody really knows good people there. I asked David Molnar and a few other people who used to live there for contacts, but so far haven't gotten any good leads.

Helped Kasimir add new Tor controller features so Torstatus can switch to using the v3 directory system:
http://trunk.torstatus.kgprog.com/

C.2.11 *The Contractor shall improve the ease of use of Tor for end users by continuing research and development on one or both of the following products: (1) all-in-one software bundle containing Tor and supporting applications, as well as an easy-to-use installer for Microsoft Windows operating systems, as well as option to install and run from a Universal Serial Bus (USB) flash device; (2) bootable CD-ROM image ("LiveCD") which contains a minimal operating system, Tor, and supporting applications. Both would have all appropriate applications pre-configured to use Tor out of the box with only minimal additional configuration required by the end user. If Contractor determines it is not feasible to develop both products, Contractor will provide detailed written technical analysis and explanation to the AR/CO. The Contractor shall make an initial public release of at least one implementation of one of these products during the term of this contract.*

Steven is working on a new branch of Vidalia that can be used in Tor Browser Bundle, for launching Firefox directly without needing the extra installer scripts called "Firefox Portable". If we get this working, then we can hopefully make progress on running multiple Firefoxes at once (one used for Tor launched by TBB, and one used for non-Tor).
http://trac.vidalia-project.net/browser/vidalia/branches/alt-launcher

Jacob Appelbaum worked on a set of instructions for rebranding Firefox, if we decide that we need to call the browser that ships in the Tor Browser Bundle something other than "Firefox". The instructions aren't complete, for example because we need more replacement logos.
https://svn.torproject.org/svn/torbrowser/trunk/build-scripts/branding/
It looks like the process of rebranding Firefox 3 is much more straightforward. We have "move to FF3" on our TBB roadmap.

Work by Martin Peck and Kyle Williams on the Tor VM project continues. We have a very early prototype available now:

http://peertech.org/files/demo/testinfo.html
and we hope to give it some more testing and better documentation in the coming months.

C.2.12 *The Contractor shall continue to develop and implement improvements to the bridge relay and bridge directory authority mechanisms to improve the usability, performance and reliability of the Tor network by users in countries with government-imposed Internet censorship.*

From the Tor 0.2.1.6-alpha ChangeLog:
"Fix a bug when parsing ports in tor_addr_port_parse() that caused Tor to fail to start if you had it configured to use a bridge relay. Fixes bug 809. Bugfix on 0.2.1.5-alpha."

The number of active bridge relays is going up, now that Tor 0.2.0.x has become the recommended stable version. For a few data points, we had 40 Running bridges at the end of July, 75 Running bridges at the end of August, and 94 Running bridges at the end of September.

C.2.13 *The Contractor shall research and document additional options for the scalability of the Tor* network *beyond 2 million concurrent users, including analysis of splitting the network into multiple segments, switching to datagram-based protocols, and improving the load balancing within the network.*

Joel Reardon, Ian Goldberg's student at Waterloo, has finished the final version of his thesis "Improving Tor using a TCP-over-DTLS tunnel":
http://uwspace.uwaterloo.ca/handle/10012/4011
We funded this research (along with 4x matching funding from MITACS in Canada) in the hopes that it would move us close enough to being able to switch to a UDP design that we can put it on the Tor development roadmap at some point. Many large challenges remain, but this is also promising work in that it shows that we can expect very serious performance improvements if we go this route.

We've started hunting more thoroughly for solutions to Bug 676:
https://bugs.torproject.org/flyspray/index.php?do=details&id=696
The issue is that some of the v3 directory authorities are keeping bad statistics on uptimes and stability of relays, which means they are not assigning the Stable or Guard flag correctly to them. The result is that the networkstatus consensus mislabels them, and clients end up not choosing relays or circuits in an efficient manners. This bug not only results in bad performance for clients, but also results in overloading some relays, leading to worse performance.

From the Tor 0.2.1.6-alpha ChangeLog:
"Implement most of Proposal 152: allow specialized servers to permit single-hop circuits, and clients to use those servers to build single-hop circuits when using a specialized controller. Patch from Josh Albrecht. Resolves feature request 768."
"Fixed some memory leaks -- some quite frequent, some almost impossible to trigger -- based on results from Coverity."

Several security- and integrity-related bugfixes from Tor 0.2.0.31:
"Make sure that two circuits can never exist on the same connection with the same circuit ID, even if one is marked for close. This is conceivably a bugfix for bug 779. Bugfix on 0.1.0.4-rc."

"Relays now reject risky extend cells: if the extend cell includes a digest of all zeroes, or asks to extend back to the relay that sent the extend cell, tear down the circuit. Ideas suggested by rovv." "If not enough of our entry guards are available so we add a new one, we might use the new one even if it overlapped with the current circuit's exit relay (or its family). Anonymity bugfix pointed out by rovv."

C.2.14 The *Contractor shall continue research into the option of providing incentives for Tor users to run Tor relay servers. If further research indicates that this should be pursued, the Contractor shall develop a project plan and timeline for this work. If further research indicates this option should be abandoned, the Contractor shall document and explain in writing the reasoning behind this decision.*

No progress this month.

C.2.15 *The Contractor shall develop a more reliable download mechanism for the Tor browser bundle for users on slow and/or unreliable network connections, by means of a split download of multiple smaller files, implementation of a lightweight download manager, reduction in the software bundle file size, or other method as chosen by the Contractor.*

The "gettor" email auto-responder is up and mostly working. We still need to do translations for it and other usability features.
https://www.torproject.org/finding-tor

Nick continued work on codename Glider, our auto update design. We're still trying to nail down the server-side design, before we move to how the client-side will behave. In theory, since the auto updater will let people update over Tor, it will allow people blocked from the Tor website to still get updates.

C.2.16 The *Contractor shall test the Tor browser bundle on multiple computer systems and analyze these systems afterwards for any changes to the system that may have been made inadvertently by use of the Tor browser bundle. The Contractor shall document any such changes found and develop a plan to reduce the footprint of Tor browser bundle use.*

No changes.

We've started to think about moving the Tor Browser Bundle from Firefox 2 to Firefox 3. This will mean we should measure new traces. We'll do it once Torbutton is known to be more stable on Firefox 3.

C.2.17 *The Contractor shall develop or adapt existing open source software to implement a web-based portal to manage the translations of text into multiple languages for the user interface text of* software *of Torbutton and Vidalia and other software that may in the future be included in the Tor browser bundle. The web site must allow non-technical users the ability to contribute translations by providing text to be translated in English, as well as any needed context on the use of the text, and allowing users to enter the translation into their language from their web browser.*

We have our translation server up and online:
https://translation.torproject.org/

https://www.torproject.org/translation-portal

Users continued to submit updated translations for many different languages.

Torbutton

نسخه جاری: 1.0.4 (1 جون 2006).

نویسنده: Scott Squires.

ای میل: squires در همین دامین.

نصب: از لینک صفحه Torbutton در addons.mozilla.org استفاده کنید.

منبع: به مخزن مراجعه کنید یا از لینک مستقیم استفاده کنید.

مراجع نوشتاری: creditschangelog | license |

Torbutton این امکان را به کاربران Firefox می دهد که با یک کلیک ارتباط با Tor را قطع و وصل کنند. به این ترتیب که پس از نصب یک نشانگر در نوار پایین صفحه پیامهای نظیر "Tor Enabled" (سبز) و "Tor Disabled" (قرمز) را نشان خواهد داد. در این حالت کاربر با کلیک روی نشانگرمی تواند وضعیت را تغییر دهد. چنانچه کاربر (یا یک extension دیگر) وضعیت proxy را تغییر دهند این تغییر بصورت اتومائیک در نشانگر منعکس خواهد شد.

برخی کاربران ممکن است یک دکمه در نوار ابزار را به نشانگر ترجیح بدهند. این دکمه هم در Torbutton در نظر گرفته شده است. کافی است با right-click روی نوار ابزار مورد نظر و انتخاب "Customize..." دکمه Torbutton را روی نوار بکشید. همچنین با رفتن در Tools-> Extensions و انتخاب Torbutton و کلیک روی Preferences می توان نشانگر را مخفی کرد.

نسخه های جدید Firefoxes امکان ارسال DNS resolves را از طریق socks proxy دارند. Torbutton از این امکان استفاده می کند (اگر نسخه شما آن را پشتیبانی کند).

نصب

لینک برای نصب: از لینک صفحه Torbutton در addons.mozilla.org استفاده کنید.

در واقع Torbutton تنها تنظیمات proxy در Firefox را مدیریت می کند. دقت کنید که Tor و privoxy باید جداگانه نصب شوند (به مراجع نوشتاری Tor مراجعه کنید).

همین؟

قرار بوده است که Torbutton خیلی ساده باشد. آنچه در بالا توضیح داده شد تمام چیزی است که بیشتر کاربران نیاز دارند بدانند.

بصورت پیش فرض Torbutton از انتخابهایی استفاده می کند که توسط توسعه دهندگان Tor برای نسخه Firefox شما پیشنهاد شده اند. اگر Tor client شما روی یک localhost نیست یا اینکه از یک درگاه غیر استاندارد استفاده می کند یا می خواهید privoxy را دور بزنید یا هر دلیل دیگری برای تنظیم دستی proxy دارید می توانید این کار را ازطریق پنجره تنظیمات extension انجام دهید. اگر شما می خواهید چند پروکسی را تنظیم کنید (مثلا اگرارتباط اینترنت شما از یک پروکسی استفاده می کند) نیاز دارید از یک راه حل پیشرفته تر نظیر SwitchProxy استفاده کنید.

سوالات مکرر

می خواهم خط "No proxy for" را تغییر بدهم. آیا پنجره ای برای این کار هست یا باید تنظیمات proxy را بصورت معمول تغییر بدهم؟ پیشنهادی دارید؟

Torbutton از طریق تغییر تنظیمات Firefox کار می کند. اما Torbutton تنظیم "No proxy for" را دست نمی زند. بنابر این شما می توانید از طریق پنجره تنظیمات proxy در Firefox با این تنظیم هر کاری بکنید.

The Tor Project
122 Scott Circle
Dedham, MA 02026 USA
https://torproject.org/

From: Roger Dingledine, Tor Project Leader
To: Kelly DeYoe, program officer, IBB
RE: contract BBGCON1807S6441
Date: January 10, 2008

This report documents progress in December 2007 on contract BBGCON1807S6441 between IBB and The Tor Project.

C.2.0 New package releases and related software.

Tor 0.2.0.13-alpha (released Dec 21) adds a fourth v3 directory authority run by Geoff Goodell, fixes many more bugs, and adds a lot of infrastructure for upcoming features.

Tor 0.2.0.14-alpha (released Dec 23) and 0.2.0.15-alpha (released Dec 25) fix a bunch of bugs with the features added in 0.2.0.13-alpha.

C.2.1 The Contractor shall continue design, development and implementation of enhancements to the existing Tor software to increase its suitability as a tool for Internet users in countries with government-sponsored Internet censorship to circumvent censorship controls, based on the existing research and documentation performed during the previous contract period (e.g. as described in the paper "Design of a blocking-resistant anonymity system").

Continuing enhancements have been made to the Tor website Chinese translation.

Tor 0.2.0.13-alpha features:
"Tor can now be configured to read a GeoIP file from disk in one of two formats. This can be used by controllers to map IP addresses to countries. Eventually, it may support exit-by-country."
"When configured, bridge relays remember which countries users are coming from, and report aggregate information in their extra-info documents, so that the bridge authorities can learn where Tor is blocked."

Tor 0.2.0.13-alpha bugfixes:
"When we were reading router descriptors from cache, we were ignoring the annotations -- so for example we were reading in bridge-purpose descriptors as general-

purpose descriptors. Bugfix on 0.2.0.8-alpha."
"If we can't expand our list of entry guards (e.g. because we're using bridges or we have StrictEntryNodes set), don't mark relays down when they fail a directory request. Otherwise we're too quick to mark all our entry points down. Bugfix on 0.1.2.x."

New Tor module "bridgedb". The main bridge authority exports its bridge descriptors, and a networkstatus summary describing which ones are reachable and running, and the bridgedb tracks this information and gives out bridge addresses based on time-and-network-location (https://bridges.torproject.org/) or email address (bridges@torproject.org).

Continuing work on the Vidalia interface, to smooth out interface bugs and make the upcoming 0.1.0 Vidalia release more usable. Also, Vidalia now uses SSL when doing its GeoIP lookups to the central Vidalia server.

C.2.2 *The Contractor shall submit system architecture and technical design documentation for Tor enhancements specifically related to anti-censorship improvements in C.2.1 to the Authorized Representative of the Contracting Officer (AR/CO) for review and approval before development and implementation. Significant changes to the design that are discovered during development must be documented and reviewed by the AR/CO as soon as the Contractor becomes aware of the need for these revisions.*

We have an initial plan for how to let people download Tor plus its documentation from the DirPort of any bridge relay or normal relay:
https://www.torproject.org/svn/trunk/doc/spec/proposals/127-dirport-mirrors-downloads.txt

We have the beginnings of a specification for "bridge communities": in this idea, small groups of volunteers could set up their own bridge directory authority, and clients would learn about all the bridges in that community and never need to interact with the official bridge authorities:
https://www.torproject.org/svn/trunk/doc/spec/proposals/128-bridge-families.txt

The development version of Torbutton now has some preliminary design documentation, mostly consisting of developer-oriented explanations of the various features and design options. We've uncovered a variety of Firefox bugs that we'll be bringing up with the Mozilla team in January.
https://torbutton.torproject.org/dev/design

C.2.3 *The Contractor shall develop and implement the bridge relay mechanism, as designed during the previous contract period, to allow individual Tor users to easily reconfigure their Tor client to automatically relay traffic from users in countries with government-imposed Internet censorship so as to circumvent that censorship.*

Tor 0.2.0.13-alpha bugfixes:
"We were ignoring our RelayBandwidthRate for the first 30 seconds after opening a circuit -- even a relayed circuit. Bugfix on 0.2.0.3-alpha."
"When we decided to send a 503 response to a request for servers, we were then also

sending the server descriptors: this defeats the whole purpose. Fixes bug 539; bugfix on 0.1.2.x."

Tor 0.2.0.13-alpha features:
"Bridge relays now behave like clients with respect to time intervals for downloading new consensus documents -- otherwise they stand out. Bridge users now wait until the end of the interval, so their bridge relay will be sure to have a new consensus document."
"Add a new config option BridgeRelay that specifies you want to be a bridge relay. Right now the only difference is that it makes you answer begin_dir requests, and it makes you cache dir info, even if your DirPort isn't on."

C.2.4 *The Contractor shall develop and implement the bridge directory authority mechanism, as designed during the previous contract period, to allow Tor clients configured as bridge relays (as described in C.2.3) to communicate their existence to the bridge directory authority, and to provide a subset of addresses of available bridge relays to Tor users in countries with government-imposed Internet censorship so that they may access the Tor network.*

Tor 0.2.0.13-alpha features:
"Three new config options (AlternateDirAuthority, AlternateBridgeAuthority, and AlternateHSAuthority) that let the user selectively replace the default directory authorities by type, rather than the all-or-nothing replacement that DirServer offers."
"Bridge directory authorities now do reachability testing on the bridges they know. They provide router status summaries to the controller via "getinfo ns/purpose/bridge", and also dump summaries to a file periodically."

Tor 0.2.0.14-alpha features:
"If bridge authorities set BridgePassword, they will serve a snapshot of known bridge routerstatuses from their DirPort to anybody who knows that password. Unset by default."
"If we receive a general-purpose descriptor and then receive an identical bridge-purpose descriptor soon after, don't discard the next one as a duplicate."

C.2.5 *The Contractor shall design and develop revisions to the Tor network protocols to hide the network signature of Tor traffic so it is difficult for government-sponsored Internet censors to identify Tor traffic and trivially block it.*

We continued to deploy the new design for the normalized TLS handshake.

There are still some steps of the "TLS blending in" arms race that we aren't yet planning to take -- for example, Firefox offers a cipher mode that openssl has never heard of, so we'll need to fake that somehow.

This development and deployment will be continuing into 2008.

C.2.6 *The Contractor shall design enhancements to Tor's cell-based protocol to improve performance on substandard network connections including those with low bandwidth and/or*

high latency and/or high packet loss.

No changes.

C.2.7 *The Contractor shall continue development of enhancements to improve the scalability of the Tor network toward the goal of supporting 2 million or more concurrent end users. This requirement is only a goal for system scalability and is not a requirement on number of actual concurrent users of the Tor network.*

We set up the Tor relay "lefkada" (run by Geoff Goodell) as the fourth v3 directory authority.

Tor 0.2.0.13-alpha feature:
"New config options AuthDirBadDir and AuthDirListBadDirs for authorities to mark certain relays as "bad directories" in the networkstatus documents. Also supports the "!baddir" directive in the approved-routers file."

Tor 0.2.0.13-alpha bugfix:
"Stop thinking that 0.1.2.x directory servers can handle "begin_dir" requests. Should ease bugs 406 and 419 where 0.1.2.x relays are crashing or mis-answering these types of requests."

C.2.8 *The Contractor shall work with IBB staff and other IBB contractors to identify tasks in support of this program that might be developed collaboratively with Contractor. Tasks in areas such as documentation, bug fixes, software testing, and any other areas involving specific knowledge of foreign government-sponsored Internet censorship may be especially appropriate for this purpose.*

No reports for this month.

C.2.9 *The Contractor shall communicate tasks identified for delegation to IBB in C.2.8 to the AR/CO and negotiate time frames for their completion. The Contractor shall monitor and coordinate work performed by IBB staff on delegated tasks and integrate it into Tor software releases as appropriate.*

No reports for this month.

C.2.10 *The Contractor shall promote active growth of the Tor server network and advocacy of Tor products to increase the performance, stability, and usability of Tor.*

Roger spoke at 24C3 in Berlin to an audience of several hundred developers and privacy enthusiasts. Basically he gave an overview of some of the big technical things we did in 2007, some of the policy/legal issues that we're tackling, and some of the technical things that need to come next. The focus was on Germany, so it included some discussion of the upcoming data retention problems, and of the general issue with police in Germany seizing servers.

Roger also met with researchers at Georgia Tech who study botnets and other bad people on the Internet. Their research group uses Tor to perform their investigations more safely and more successfully.

C.2.11 *The Contractor shall improve the ease of use of Tor for end users by continuing research and development on one or both of the following products: (1) all-in-one software bundle containing Tor and supporting applications, as well as an easy-to-use installer for Microsoft Windows operating systems, as well as option to install and run from a Universal Serial Bus (USB) flash device; (2) bootable CD-ROM image ("LiveCD") which contains a minimal operating system, Tor, and supporting applications. Both would have all appropriate applications pre-configured to use Tor out of the box with only minimal additional configuration required by the end user. If Contractor determines it is not feasible to develop both products, Contractor will provide detailed written technical analysis and explanation to the AR/CO. The Contractor shall make an initial public release of at least one implementation of one of these products during the term of this contract.*

Steven Murdoch made the first prototype of the Tor Browser Bundle, a standalone USB-oriented image for Windows that includes Tor, Vidalia, Firefox, and Torbutton.

Additional features so far includes providing packages in English, Chinese, and Farsi; delaying Firefox's start until Tor has a circuit open and it's working; adding Firefox bookmarks for Tor, Torcheck and the hidden wiki; and disabling FirefoxPortable's splash screen.

https://tor-svn.freehaven.net/svn/torbrowser/trunk/README

Sample packages at
http://www.cl.cam.ac.uk/~sjm217/volatile/tor_browser/

The Tor Project
122 Scott Circle
Dedham, MA 02026 USA
https://torproject.org/

From: Roger Dingledine, Tor Project Leader
To: Kelly DeYoe, program officer, IBB
RE: contract BBGCON1807S6441
Date: March 10, 2008

This report documents progress in February 2008 on contract BBGCON1807S6441 between IBB and The Tor Project.

C.2.0 New package releases and related software.

Tor 0.2.0.20-rc (released Feb 24) is the first release candidate for the 0.2.0 series. It makes more progress towards normalizing Tor's TLS handshake, makes hidden services work better again, helps relays bootstrap if they don't know their IP address, adds optional support for linking in openbsd's allocator or tcmalloc, allows really fast relays to scale past 15000 sockets, and fixes a bunch of minor bugs reported by Veracode.
http://archives.seul.org/or/talk/Feb-2008/msg00279.html

Tor 0.2.0.19-alpha (released Feb 9) makes more progress towards normalizing Tor's TLS handshake, makes path selection for relays more secure and IP address guessing more robust, and generally fixes a lot of bugs in preparation for calling the 0.2.0 branch stable.
http://archives.seul.org/or/talk/Feb-2008/msg00134.html

Torbutton 1.1.13 (released Feb 1), 1.1.14 (released Feb 24), and 1.1.15 (released Feb 26) fix many more potential privacy and identity leaks, mostly based on exploits found by Greg Fleischer. They also add support for automatic updates via the usual Firefox extension upgrade approach.
https://torbutton.torproject.org/dev/CHANGELOG

C.2.1 *The Contractor shall continue design, development and implementation of enhancements to the existing Tor software to increase its suitability as a tool for Internet users in countries with government-sponsored Internet censorship to circumvent censorship controls, based on the existing research and documentation performed during the previous contract period (e.g. as described in the paper "Design of a blocking-resistant anonymity system").*

Continuing enhancements have been made to the Tor website Chinese and Russian translations.

Work continued toward the upcoming Vidalia 0.1.0 release (which came out March 1): support for launching Firefox and Polipo as supporting applications; support for learning from Tor when the first circuit is ready so it can inform the user; and many other bugfixes including a few security fixes.
http://trac.vidalia-project.net/browser/vidalia/releases/vidalia-0.1.0/CHANGELOG

The Tor 0.2.0.19-alpha release contained many security-related cleanups based on an anonymously submitted code review from a static analysis tool. The Tor 0.2.0.20-rc release contained even more security-related cleanups, based on an external security analysis and audit by Veracode. Hopefully cleanups at this stage will reduce the number of times we need to push out an urgent new stable "0.2.0" release for security reasons.

From the Tor 0.2.0.19-alpha ChangeLog:
"When connecting to a bridge without specifying its key, insert the connection into the identity-to-connection map as soon as a key is learned. This prevents the Tor user's log from showing a confusing complaint periodically."
"When our consensus networkstatus has been expired for a while, stop being willing to build circuits using it. Now clients won't give themselves away by behaving uniquely if they start up with an old networkstatus view."

From the Tor 0.2.0.20-rc ChangeLog:
"Choose which bridge to use proportional to its advertised bandwidth, rather than uniformly at random. This should speed up Tor for bridge users. Also do this for people who set StrictEntryNodes."

C.2.2 *The Contractor shall submit system architecture and technical design documentation for Tor enhancements specifically related to anti-censorship improvements in C.2.1 to the Authorized Representative of the Contracting Officer (AR/CO) for review and approval before development and implementation. Significant changes to the design that are discovered during development must be documented and reviewed by the AR/CO as soon as the Contractor becomes aware of the need for these revisions.*

Continued work on a roadmap of all the future features and extensions we know we need. It's still mostly in outline form at this point:
https://www.torproject.org/svn/trunk/doc/design-paper/roadmap-future.pdf

Also sent Kelly a list on Feb 25 of specific items that Tor wants to work on in 2008 and that IBB would likely find interesting.

C.2.3 *The Contractor shall develop and implement the bridge relay mechanism, as designed during the previous contract period, to allow individual Tor users to easily reconfigure their Tor client to automatically relay traffic from users in countries with government-imposed Internet censorship so as to circumvent that censorship.*

From the Tor 0.2.0.19-alpha ChangeLog:
"If we're a relay, avoid picking ourselves as an introduction point, a rendezvous point, or as the final hop for internal circuits."
"Directory caches now fetch certificates from all authorities listed in a networkstatus consensus, even when they do not recognize them. This bugfix is particularly important for bridge users, since the bridges are their only contact point for fetching new directory information."

From the Tor 0.2.0.20-rc ChangeLog:
"Servers that don't know their own IP address should go to the authorities for their first directory fetch, even if their DirPort is off or if they don't know they're reachable yet. This will help them bootstrap better."

C.2.4 *The Contractor shall develop and implement the bridge directory authority mechanism, as designed during the previous contract period, to allow Tor clients configured as bridge relays (as described in C.2.3) to communicate their existence to the bridge directory authority, and to provide a subset of addresses of available bridge relays to Tor users in countries with government-imposed Internet censorship so that they may access the Tor network.*

We moved the BridgeDB service to our machine in Austria so it can use a real legitimate SSL certificate on https://bridges.torproject.org/

From the Tor 0.2.0.20-rc ChangeLog:
"We were comparing the raw BridgePassword entry with a base64'ed version of it, when handling a "/tor/networkstatus-bridges" directory request. Now compare correctly. This bugfix should allow bridge communities (formerly known as bridge families) to work better. Noticed by Veracode."

C.2.5 *The Contractor shall design and develop revisions to the Tor network protocols to hide the network signature of Tor traffic so it is difficult for government-sponsored Internet censors to identify Tor traffic and trivially block it.*

From the Tor 0.2.0.19-alpha ChangeLog:
"Do not include recognizeable strings in the commonname part of Tor's x509 certificates."

From the Tor 0.2.0.20-rc ChangeLog:
"Enable the revised TLS handshake based on the one designed by Steven Murdoch in

proposal 124, as revised in proposal 130. It includes version negotiation for OR connections as described in proposal 105. The new handshake is meant to be harder for censors to fingerprint, and it adds the ability to detect certain kinds of man-in-the-middle traffic analysis attacks. The version negotiation feature will allow us to improve Tor's link protocol more safely in the future."

In March we plan to enable the "encrypted directory fetch" feature by default, so Tor will resume working in countries where Smartfilter is prevalent.

C.2.6 *The Contractor shall design enhancements to Tor's cell-based protocol to improve performance on substandard network connections including those with low bandwidth and/or high latency and/or high packet loss.*

No changes.

C.2.7 *The Contractor shall continue development of enhancements to improve the scalability of the Tor network toward the goal of supporting 2 million or more concurrent end users. This requirement is only a goal for system scalability and is not a requirement on number of actual concurrent users of the Tor network.*

From the Tor 0.2.0.20-rc ChangeLog:
"Tune parameters for cell pool allocation to minimize amount of RAM overhead used."
"Add OpenBSD malloc code from phk as an optional malloc replacement on Linux: some glibc libraries do very poorly with Tor's memory allocation patterns. Pass --enable-openbsd-malloc to get the replacement malloc code."
"Stop imposing an arbitrary maximum on the number of file descriptors used for extremely high-throughput servers. Bug reported by Olaf Selke; patch from Sebastian Hahn."

From the Tor 0.2.0.19-alpha ChangeLog:
"Patch from "Andrew S. Lists" to catch when we contact a directory mirror at IP address X and he says we look like we're coming from IP address X. This was causing some Tor relays to test their reachability by testing the wrong address, and never actually publish to the main list."

C.2.8 *The Contractor shall work with IBB staff and other IBB contractors to identify tasks in support of this program that might be developed collaboratively with Contractor. Tasks in areas such as documentation, bug fixes, software testing, and any other areas involving specific knowledge of foreign government-sponsored Internet censorship may be especially appropriate for this purpose.*

No reports for this month.

C.2.9 *The Contractor shall communicate tasks identified for delegation to IBB in C.2.8 to the*

AR/CO and negotiate time frames for their completion. The Contractor shall monitor and coordinate work performed by IBB staff on delegated tasks and integrate it into Tor software releases as appropriate.

No reports for this month.

C.2.10 The Contractor shall promote active growth of the Tor server network and advocacy of Tor products to increase the performance, stability, and usability of Tor.

Roger met with security researchers in Norway at the annual "HackCon" gathering (hackcon.org). HackCon awarded Tor their annual prize for most important project. During the meeting, Roger met with an individual from the Middle East who provided feedback and contacts for users who need to learn more about the Tor Browser Bundle.

Roger and Nick both attended Shmoocon (shmoocon.org) to continue spreading the word about Tor current events. Tor was mentioned prominently in several of the more high-profile presentations. We also met with Dan Kaminsky about arranging pro-bono source code audits for Tor and about making modifications to Qemu so our USB and LiveCD distributions below can be made more flexible.

We continued working toward being able to hire Jake Appelbaum and Matt Edman as contractors starting in April or May. Jake will be working on a translation portal, auto update for Tor and supporting applications, a Windows buildbot, and other advocacy projects. Matt will be working on Vidalia maintenance, bugfixes, and new features --- for example, providing a GUI interface for the above auto update feature, letting users change their preferred language in Vidalia without requiring an application restart, and providing a better GUI for showing Tor's start-up progress.

We started preparing our Google Summer of Code 2008 application, in collaboration with The Electronic Frontier Foundation. We hope to get 4-6 student interns working with us over the summer, funded by Google.

C.2.11 *The Contractor shall improve the ease of use of Tor for end users by continuing research and development on one or both of the following products: (1) all-in-one software bundle containing Tor and supporting applications, as well as an easy-to-use installer for Microsoft Windows operating systems, as well as option to install and run from a Universal Serial Bus (USB) flash device; (2) bootable CD-ROM image ("LiveCD") which contains a minimal operating system, Tor, and supporting applications. Both would have all appropriate applications pre-configured to use Tor out of the box with only minimal additional configuration required by the end user. If Contractor determines it is not feasible to develop both products, Contractor will provide detailed written technical analysis and explanation to the AR/CO. The Contractor shall make an initial public release of at least one implementation of one of these products during the term of this contract.*

We cleaned up the Tor Browser Bundle's webpage and instructions based on feedback from users who were visiting Iran and Burma. Also started preparations to make it easy for our translators to provide an alternate languages. As of March 10, we have English, German, Italian, Polish, and Russian translations. We are working to coordinate an Arabic translation too.
https://torbrowser.torproject.org/

The new Tor Browser Bundle 0.0.7 (released Feb 8) and 0.0.8 (released Feb 15) include security updates for Firefox (2.0.12), security updates for Torbutton (1.1.13), automate generation of internationalized bundles, allow optional extensions to be placed in build-scripts/extensions, build Polipo with regular expression support (activating the forbiddenFile option), and update Polipo configuration based on suggestions from Incognito's Polipo configuration:
https://tor-svn.freehaven.net/svn/torbrowser/branches/stable/README

```
00000000  16 03 01 00 31 01 00 00  2D 03 01 44 C8 4D B5 39   ....1... -..D.M.9
00000010  BB FD 16 24 5B 78 CD 0A  1E 6C E6 D5 4D 0E 49 FC   ...$[x.. .l..M.I.
00000020  97 56 3F 37 6E 11 44 F1  D3 F5 16 00 00 06 00 39   .V?7n.D. .......9
00000030  00 33 00 16 01 00                                  .3....

00000000  16 03 01 00 2A 02 00 00  26 03 01 44 C8 4C C5 47   ....*... &..D.L.G
00000010  56 19 EE 18 CB DC 75 74  AB 3D B5 96 1D 8C 6E B1   V.....ut .=....n.
00000020  BE 97 F4 D4 23 8A F6 00  00 00 00 00 00 33 00 16   ....#... .....3..
00000030  03 01 03 A0 0B 00 03 9C  00 03 99 00 01 C4 30 82   ........ ......0.
00000040  01 C0 30 82 01 29 A0 03  02 01 02 02 04 44 C8 36   ..0..).. .....D.6
00000050  82 30 0D 06 09 2A 86 48  86 F7 0D 01 01 05 05 00   .0...*.H ........
00000060  30 2A 31 0C 30 0A 06 03  55 04 0A 13 03 54 4F 52   0*1.0... U....TOR
00000070  31 1A 30 18 06 03 55 04  03 14 11 6A 61 69 6C 61   1.0...U. ...jaila
00000080  69 20 3C 69 64 65 6E 74  69 74 79 3E 30 1E 17 0D   i <ident ity>0...
00000090  30 36 30 37 32 37 30 33  34 34 30 32 5A 17 0D 30   06072703 4402Z..0
000000A0  36 30 37 32 37 30 35 34  34 30 32 5A 30 1F 31 0C   60727054 402Z0.1.
000000B0  30 0A 06 03 55 04 0A 13  03 54 4F 52 31 0F 30 0D   0...U... .TOR1.0.
000000C0  06 03 55 04 03 13 06 6A  61 69 6C 61 69 30 81 9F   ..U....j ailai0..
000000D0  30 0D 06 09 2A 86 48 86  F7 0D 01 01 01 05 00 03   0...*.H. ........
000000E0  81 8D 00 30 81 89 02 81  81 00 A3 B0 C7 85 6D 51   ...0.... ......mQ
000000F0  02 3F 4D B7 87 5D C7 4B  BE D5 61 42 A2 DA 03 C3   .?M..].K ..aB....
00000100  06 8E 3C 44 47 9F 6E E8  1D 56 20 5A 4B 81 56 AE   ..<DG.n. .V ZK.V.
00000110  86 8C C6 EA 1E C9 A2 65  62 4F 62 83 F5 CA 92 74   .......e bOb....t
00000120  C0 B8 22 76 E6 90 3E 91  4E 69 50 8E 76 19 B4 93   .."v..>. NiP.v...
00000130  8B CF 0B AD 79 73 00 22  C9 6F 17 52 1E EB 30 AB   ....ys." .o.R..0.
00000140  E1 CD AE 17 91 9F 3D 77  E0 44 9A 8B F2 DC F5 84   ......=w .D......
00000150  97 5D 0E B7 9F 6A 8B DE  68 E9 03 35 5A B5 73 53   .]...j.. h..5Z.sS
00000160  00 ED 27 EA CF 5E 35 32  DB 85 02 03 01 00 01 30   ..'..^52 .......0
00000170  0D 06 09 2A 86 48 86 F7  0D 01 01 05 05 00 03 81   ...*.H.. ........
00000180  81 00 DA E8 D1 E5 1E 36  0E 80 F2 5F 00 6E CF F6   .......6 ..._.n..
00000190  75 8E B0 14 08 69 1F F6  BF 08 96 CE ED 7D 5E BB   u....i.. .....}^.
000001A0  86 99 67 1B C4 5D B5 D9  39 38 2B F8 82 8A 80 5E   ..g..].. 98+....^
000001B0  55 68 62 6F F9 C0 C9 72  F0 72 23 59 CA 33 8C 79   Uhbo...r .r#Y.3.y
000001C0  C8 ED 8B D4 FB CF 68 58  57 95 C8 4B D4 51 E7 8E   ......hX W..K.Q..
000001D0  7F DD 95 2E 54 6A 44 CA  89 B2 49 C7 72 72 68 BA   ...TjD. ..I.rrh.
000001E0  75 10 49 84 44 C5 2E 43  5D E6 EB 8A B8 1B 01 C9   u.I.D..C ].......
000001F0  FF D1 DE 84 78 51 3A 18  98 0F 89 D8 D8 15 89 41   ....xQ:. .......A
00000200  D5 75 00 01 CF 30 82 01  CB 30 82 01 34 A0 03 02   .u...0.. .0..4...
00000210  01 02 02 04 44 C8 36 82  30 0D 06 09 2A 86 48 86   ....D.6. 0...*.H.
00000220  F7 0D 01 01 05 05 00 30  2A 31 0C 30 0A 06 03 55   .......0 *1.0...U
00000230  04 0A 13 03 54 4F 52 31  1A 30 18 06 03 55 04 03   ....TOR1 .0...U..
00000240  14 11 6A 61 69 6C 61 69  20 3C 69 64 65 6E 74 69   ..jailai  <identi
00000250  74 79 3E 30 1E 17 0D 30  36 30 37 32 37 30 33 34   ty>0...0 60727034
00000260  34 30 32 5A 17 0D 30 37  30 37 32 37 30 33 34 34   402Z..07 07270344
00000270  30 32 5A 30 2A 31 0C 30  0A 06 03 55 04 0A 13 03   02Z0*1.0 ...U....
00000280  54 4F 52 31 1A 30 18 06  03 55 04 03 14 11 6A 61   TOR1.0.. .U....ja
00000290  69 6C 61 69 20 3C 69 64  65 6E 74 69 74 79 3E 30   ilai <id entity>0
000002A0  81 9F 30 0D 06 09 2A 86  48 86 F7 0D 01 01 01 05   ..0...*. H.......
000002B0  00 03 81 8D 00 30 81 89  02 81 81 00 E8 90 09 CE   .....0.. ........
000002C0  EC DB 50 6D 09 AC 9F EE  27 D1 48 A0 EB ED 77 E1   ..Pm.... '.H...w.
000002D0  D8 B3 39 7F F0 66 A9 FB  F3 B3 79 61 48 DA 12 0E   ..9.f.. ..yaH...
000002E0  B9 36 5C 21 CD FC E5 0C  0E 17 1C 6B 52 3D C4 14   .6\!.... ...kR=..
000002F0  99 58 41 81 91 20 D5 44  59 5E 03 5B 0D 7C 5F DF   .XA.. .D Y^.[.|_.
```

```
00000300  7A 4C 64 86 BD B1 5C 3B  1F 1E B1 18 6E 43 FC 91   zLd...\; ....nC..
00000310  C7 9B F9 5F A0 B8 E2 18  D9 1D 37 F9 D8 CB 9C E3   ..._.... ..7.....
00000320  57 26 20 C7 83 44 9E BC  2F AA 16 6D 07 E8 FF 10   W& ..D.. /..m....
00000330  1D D5 43 A9 CA B9 A2 39  5B 45 D2 01 02 03 01 00   ..C....9 [E......
00000340  01 30 0D 06 09 2A 86 48  86 F7 0D 01 01 05 05 00   .0...*.H ........
00000350  03 81 81 00 37 F5 DF 17  9F 3B 62 B3 07 54 94 3D   ....7... .;b..T.=
00000360  40 CE F7 26 76 D4 68 B4  6A 6A 2E 92 34 ED 22 C5   @..&v.h. jj..4.".
00000370  25 BC 1C 79 5A 03 42 A1  D5 B9 73 54 91 C9 B6 78   %..yZ.B. ..sT...x
00000380  78 1A 2A 9E 4D 6B 5C 4B  23 6B D2 6F 4F 7A BD 1B   x.*.Mk\K #k.oOz..
00000390  7B C7 54 7B C2 F4 6C E9  5C C2 90 54 D0 D9 BE 36   {.T{..l. \..T...6
000003A0  25 79 C8 41 0A 23 38 BB  21 8C E2 89 58 40 8C 2E   %y.A.#8. !...X@..
000003B0  0E 65 B1 22 6D 8C B2 29  76 E0 7C 8D 2D C0 D8 D6   .e."m..) v.|.-...
000003C0  D7 B9 F0 5F DF AD 5A 10  D7 E9 FB 59 1A 8E B4 6A   ..._..Z. ...Y...j
000003D0  23 87 5D E5 16 03 01 01  8D 0C 00 01 89 00 80 FF   #.]..... ........
000003E0  FF FF FF FF FF FF FF C9  0F DA A2 21 68 C2 34 C4   ........ ...!h.4.
000003F0  C6 62 8B 80 DC 1C D1 29  02 4E 08 8A 67 CC 74 02   .b.....) .N..g.t.
00000400  0B BE A6 3B 13 9B 22 51  4A 08 79 8E 34 04 DD EF   ...;.."Q J.y.4...
00000410  95 19 B3 CD 3A 43 1B 30  2B 0A 6D F2 5F 14 37 4F   ....:C.0 +.m._.7O
00000420  E1 35 6D 6D 51 C2 45 E4  85 B5 76 62 5E 7E C6 F4   .5mmQ.E. ..vb^~..
00000430  4C 42 E9 A6 37 ED 6B 0B  FF 5C B6 F4 06 B7 ED EE   LB..7.k. .\......
00000440  38 6B FB 5A 89 9F A5 AE  9F 24 11 7C 4B 1F E6 49   8k.Z.... .$.|K..I
00000450  28 66 51 EC E6 53 81 FF  FF FF FF FF FF FF FF 00   (fQ..S.. ........
00000460  01 02 00 80 8C FA 7C 3F  E4 7A BE 0C 80 28 73 79   ......|? .z...(sy
00000470  E1 10 DE 47 5A 90 5D 7D  CC 6E 89 2B ED E5 80 6E   ...GZ.]} .n.+...n
00000480  E4 34 6D 81 54 08 AB A2  07 B5 17 E2 A0 58 D6 EE   .4m.T... .....X..
00000490  90 71 41 A0 43 05 87 E7  ED 4D 9B 56 78 54 A3 A0   .qA.C... .M.VxT..
000004A0  AD 0E 0A 56 1E 20 E1 C3  A8 E4 D1 E7 C9 37 DE 6F   ...V. .. .....7.o
000004B0  98 C6 63 8A 73 6B 03 C2  9A 17 E8 A3 F3 98 6D 8F   ..c.sk.. ......m.
000004C0  BA D1 E9 32 46 0A D9 32  0F 0B E4 B4 D2 75 DC 7B   ...2F..2 .....u.{
000004D0  7A BE D3 51 FF F7 80 3C  5C 2B 41 6A 44 BA 22 78   z..Q...< \+AjD."x
000004E0  99 24 D0 1A 00 80 61 A4  A4 52 43 91 14 E6 30 60   .$....a. .RC...0`
000004F0  83 B2 6F 87 4D 4C 43 53  5A 13 E8 73 73 D1 BA 0D   ..o.MLCS Z..ss...
00000500  CC 62 82 33 89 81 6A 6D  B2 B4 3A D2 9D A9 2C 3E   .b.3..jm ..:...,>
00000510  FB A8 45 24 34 BA D1 0D  0F 99 65 E7 52 E2 AD 8F   ..E$4... ..e.R...
00000520  05 87 3E 8E F3 40 0A 62  B3 99 7E 49 C5 A3 00 B8   ..>..@.b ..~I....
00000530  58 4E 9B D0 C7 47 A5 F6  24 58 1C 50 69 E0 10 AE   XN...G.. $X.Pi...
00000540  36 8C 80 B9 3A 80 96 92  4D 09 7D 59 18 F5 0D A4   6...:... M.}Y....
00000550  17 1B 27 AA 17 27 A7 A5  D6 19 4D F6 9F 38 D8 45   ..'..'.. ..M..8.E
00000560  2B D6 F6 D6 36 82 16 03  01 00 0F 0D 00 00 07 04   +...6... ........
00000570  03 04 01 02 00 00 0E 00  00 00                     ........ ..

00000000  16 03 01 03 A0 0B 00 03  9C 00 03 99 00 01 C4 30   ........ .......0
00000010  82 01 C0 30 82 01 29 A0  03 02 01 02 02 04 44 C8   ...0..). ......D.
00000020  4D B4 30 0D 06 09 2A 86  48 86 F7 0D 01 01 05 05   M.0...*. H.......
00000030  00 30 2A 31 0C 30 0A 06  03 55 04 0A 13 03 54 4F   .0*1.0.. .U....TO
00000040  52 31 1A 30 18 06 03 55  04 03 14 11 63 6C 69 65   R1.0...U ....clie
00000050  6E 74 20 3C 69 64 65 6E  74 69 74 79 3E 30 1E 17   nt <iden tity>0..
00000060  0D 30 36 30 37 32 37 30  35 32 33 30 30 5A 17 0D   .0607270 52300Z..
00000070  30 36 30 37 32 37 30 37  32 33 30 30 5A 30 1F 31   06072707 2300Z0.1
00000080  0C 30 0A 06 03 55 04 0A  13 03 54 4F 52 31 0F 30   .0...U.. ..TOR1.0
00000090  0D 06 03 55 04 03 13 06  63 6C 69 65 6E 74 30 81   ...U.... client0.
000000A0  9F 30 0D 06 09 2A 86 48  86 F7 0D 01 01 01 05 00   .0...*.H ........
000000B0  03 81 8D 00 30 81 89 02  81 81 00 D9 07 16 53 71   ....0... ......Sq
000000C0  1F E1 85 EB A3 78 01 69  70 0A 4E 50 54 95 B6 CD   .....x.i p.NPT...
```

```
000000D0  B3 5B C0 25 19 21 20 01  14 FD 64 EC F6 1D A0 EA   .[.%.! . ..d.....
000000E0  A0 F7 43 B0 A9 A7 87 46  F2 E0 95 6F 0C C2 CC 9A   ..C....F ...o....
000000F0  C0 46 44 0B 23 64 6B 80  E7 7C FE 0E 35 D7 FF DD   .FD.#dk. .|..5...
00000100  82 38 8E 84 8C 29 DA 3D  11 1C 63 92 FA C6 F1 BE   .8...).= ..c.....
00000110  89 FB C0 34 85 89 0F E0  2B 0C BC 59 0A F2 D8 67   ...4.... +..Y...g
00000120  C3 83 EA F6 30 81 8B 98  23 E9 94 CE 0F 87 58 11   ....0... #.....X.
00000130  07 28 CB 88 22 02 7F B6  36 C3 E9 02 03 01 00 01   .(.".. 6.......
00000140  30 0D 06 09 2A 86 48 86  F7 0D 01 01 05 05 00 03   0...*.H. ........
00000150  81 81 00 71 5A A9 A6 4D  16 5C C3 24 10 AD D1 D6   ...qZ..M .\.$....
00000160  29 18 AD 18 3A 9F 9E F1  F0 58 82 AD A2 2E A4 D7   )...:... .X......
00000170  66 A6 A4 0B B8 8D 67 7F  97 91 1E 45 0C 33 21 A0   f.....g ...E.3!.
00000180  EA 8D 1B A4 4A AA 51 3B  A0 E3 AF A4 A2 C7 A6 B8   ....J.Q; ........
00000190  57 43 35 12 24 89 98 73  4D AA 1C 6F A3 A7 07 04   WC5.$..s M..o....
000001A0  EC CB 7C 21 A0 E4 07 33  D9 EB 89 E8 14 BD E3 CB   ..|!...3 ........
000001B0  5D 00 F7 09 24 2F 01 49  17 17 DC 54 40 17 CA C0   ]...$/.I ...T@...
000001C0  A9 3F 4B 62 EB 47 91 6B  49 88 50 3C 8C 1B 01 8A   .?Kb.G.k I.P<....
000001D0  E1 92 A0 00 01 CF 30 82  01 CB 30 82 01 34 A0 03   ......0. ..0..4..
000001E0  02 01 02 02 04 44 C8 4D  B4 30 0D 06 09 2A 86 48   .....D.M .0...*.H
000001F0  86 F7 0D 01 01 05 05 00  30 2A 31 0C 30 0A 06 03   ........ 0*1.0...
00000200  55 04 0A 13 03 54 4F 52  31 1A 30 18 06 03 55 04   U....TOR 1.0...U.
00000210  03 14 11 63 6C 69 65 6E  74 20 3C 69 64 65 6E 74   ...clien t <ident
00000220  69 74 79 3E 30 1E 17 0D  30 36 30 37 32 37 30 35   ity>0... 06072705
00000230  32 33 30 30 5A 17 0D 30  37 30 37 32 37 30 35 32   2300Z..0 70727052
00000240  33 30 30 5A 30 2A 31 0C  30 0A 06 03 55 04 0A 13   300Z0*1. 0...U...
00000250  03 54 4F 52 31 1A 30 18  06 03 55 04 03 14 11 63   .TOR1.0. ..U....c
00000260  6C 69 65 6E 74 20 3C 69  64 65 6E 74 69 74 79 3E   lient <i dentity>
00000270  30 81 9F 30 0D 06 09 2A  86 48 86 F7 0D 01 01 01   0..0...* .H......
00000280  05 00 03 81 8D 00 30 81  89 02 81 81 00 C1 17 07   ......0. ........
00000290  AD 40 00 20 BC A3 9D 31  55 3C 1D F7 53 48 3B 64   .@. ...1 U<..SH;d
000002A0  D1 EC D3 B1 AF AD 3B E5  30 06 B4 14 97 27 BE FA   ......;. 0....'..
000002B0  6D 7E A1 F5 95 AC A0 6F  43 B9 BF 8F 2A 12 44 A1   m~.....o C...*.D.
000002C0  BA 14 9D 61 DA F4 CD E7  71 4F 1E A8 8E 40 11 AA   ...a.... qO...@..
000002D0  B5 57 CC D7 53 4E 35 DB  DC 4C CF 26 C2 A3 A8 F1   .W..SN5. .L.&....
000002E0  6E 19 58 AE A9 F3 F8 CB  9C 89 1A 9F B5 4C E3 C6   n.X..... .....L..
000002F0  DE 4E AF 5A 57 84 15 69  A0 F7 91 D8 59 48 A9 3A   .N.ZW..i ....YH.:
00000300  E2 EF 61 E2 02 59 A5 01  F1 EE 70 62 95 02 03 01   ..a..Y.. ..pb....
00000310  00 01 30 0D 06 09 2A 86  48 86 F7 0D 01 01 05 05   ..0...*. H.......
00000320  00 03 81 81 00 4D D9 DE  7C DF EC 8D 38 8D 9F 42   .....M.. |...8..B
00000330  60 A3 FF B7 BB 07 51 30  FB 71 B8 2F 19 76 5A FB   `.....Q0 .q./.vZ.
00000340  74 11 EE 56 CB 63 A0 3A  AA D4 7A D2 41 06 5B 2A   t..V.c.: ..z.A.[*
00000350  83 D0 8E 99 76 C1 15 4D  82 F7 CC 60 D3 83 38 2B   ....v..M ...`..8+
00000360  0F 62 44 2F 3B 1A B3 AF  83 7C F7 ED 27 C0 9E 8A   .bD/;... .|..'...
00000370  60 80 92 44 83 39 E0 49  F5 05 1D 90 DB 9A 39 89   `..D.9.I ......9.
00000380  90 91 CD 98 DE 85 99 BB  56 F5 AB 7D 9E 59 62 8A   ........ V..}.Yb.
00000390  B7 E4 E0 45 2D C1 57 5B  E3 DC 82 50 00 D9 87 B6   ...E-.W[ ...P....
000003A0  92 3F 70 6A D8 16 03 01  00 86 10 00 00 82 00 80   .?pj.... ........
000003B0  E7 3B 8A 75 53 92 FD F3  45 7D 91 C1 01 B6 A9 FD   .;.uS... E}......
000003C0  50 49 03 2D 7A 03 0A 12  D5 75 D9 E0 F0 E5 F6 B1   PI.-z... .u......
000003D0  10 B7 F6 4D 82 C9 8F 38  DB 80 8A 3A E4 D5 03 55   ...M...8 ...:...U
000003E0  08 97 E8 EB 31 D1 37 6F  B1 56 E7 15 14 3A 08 3A   ....1.7o .V...:.:
000003F0  9F AE 36 A4 98 33 BA 3A  A3 03 53 26 A2 F4 1C E0   ..6..3.: ..S&....
00000400  6F 48 E7 79 DA 46 E0 EF  54 B5 BB 14 80 D4 D7 5B   oH.y.F.. T......[
00000410  28 3A 87 90 B7 48 1E F9  55 2A 02 23 A9 D0 FC F1   (:...H.. U*.#....
00000420  9A C2 7D 62 B3 B6 A5 76  8A 91 FF AC 85 1C A9 ED   ..}b...v ........
```

```
00000430  16 03 01 00 86 0F 00 00  82 00 80 D8 A1 E0 7B DF   ........ ......{.
00000440  D9 C8 3E 21 1B 0F 4D BE  DD 94 A3 AD F5 EB AB 2B   ..>!..M. .......+
00000450  6F 59 7B 4A 91 BA 7B 72  59 42 11 D1 C5 0B 38 09   oY{J..{r YB....8.
00000460  E6 F3 F9 AB BC F5 1B 9E  9E 22 84 BE 0C 0F BD 75   ........ ."......u
00000470  03 63 AD EF 66 A8 E7 ED  97 34 FD 94 9D CF 18 03   .c..f... .4......
00000480  B0 E7 DD F9 94 2F 64 C3  B4 AB A9 B4 4E B2 B5 FB   ...../d. ....N...
00000490  D1 4B FF 88 D5 60 E0 61  B7 83 34 81 C0 A9 FE 05   .K...`.a ..4.....
000004A0  2E 8D 59 F9 47 59 9F 44  53 32 11 60 C0 2A 76 96   ..Y.GY.D S2.`.*v.
000004B0  AE DE F5 28 2C 1C A9 D0  1D 63 32 14 03 01 00 01   ...(,... .c2.....
000004C0  01 16 03 01 00 30 6E 40  C8 10 50 35 25 D4 E3 AA   .....0n@ ..P5%...
000004D0  76 EE 45 D4 E9 63 DF 4F  3C 6B 53 D0 FA E9 30 8D   v.E..c.O <kS...0.
000004E0  58 9B C9 78 EF 9E C0 6F  FB BA C9 A5 C0 46 87 91   X..x...o .....F..
000004F0  CB CE E2 3D 85 D0                                  ...=..

00000000  14 03 01 00 01 01 16 03  01 00 30 7B 1F 04 59 B1   ........ ..0{..Y.
00000010  61 44 B3 5E E0 44 31 51  5B B7 8B 1B 1A 67 CE 0A   aD.^.D1Q [....g..
00000020  71 D3 A5 A0 C1 79 64 01  2C FA 1E 9A 40 94 CD 36   q....yd. ,...@..6
00000030  9C AC 14 04 80 D7 04 7D  DE FC D3                  .......} ...

00000000  17 03 01 00 20 61 68 63  0D 30 CA 73 67 3C C1 4D   .... ahc .0.sg<.M
00000010  35 E1 BD 5F 38 CE 1A 57  1B D5 94 86 FA AF A6 C9   5.._8..W ........
00000020  1B 72 69 E2 11 17 03 01  02 20 F8 5F 14 FC A4 70   .ri..... . ._...p
00000030  64 3C 25 A3 14 F7 CB 57  1A E1 F4 92 72 2F 77 3E   d<%....W ....r/w>
00000040  19 09 D5 3B 6A DB 78 01  07 A3 45 00 E4 87 39 77   ...;j.x. ..E...9w
00000050  03 CB 60 26 37 D8 EE FB  2B B8 7D 1E 77 8C 4E F4   ..`&7... +.}.w.N.
00000060  1B 5A FD 8D 0A BC 9B F0  1C 04 47 B9 A5 77 86 CA   .Z...... ..G..w..
00000070  9B 6A 2F 42 6E 26 7A 22  0D F0 BD 93 A2 54 4F 3A   .j/Bn&z" .....TO:
00000080  7C 7F AA EC 10 E2 FD B6  C1 B7 31 AB B7 3D 3D 2B   |...... ..1..==+
00000090  90 12 4B F7 B1 BD 91 E8  B6 8A FA CE B1 40 8E 6D   ..K..... .....@.m
000000A0  2D 5F 3A 77 56 E7 AA 2A  02 99 9B 52 B0 C3 08 BC   -_:wV..* ...R....
000000B0  E9 79 BA 1D 26 DE 6F E5  95 1D 27 3E D8 25 0D F3   .y..&.o. ..'>.%..
000000C0  41 FE BA A5 04 DF E6 1F  0B 95 49 38 D1 6C 19 29   A....... ..I8.l.)
000000D0  AC B5 29 F6 05 24 BE 0A  4B C4 02 5C 99 07 79 9A   ..)..$.. K..\..y.
000000E0  BE 17 76 D0 63 E1 12 6E  D5 5B FA 80 E1 A1 71 40   ..v.c..n .[....q@
000000F0  ED 6B 03 B0 08 8A 3D 4F  1E 55 D7 56 EA 5D 66 3B   .k....=O .U.V.]f;
00000100  D7 19 4C 3E 18 AE A7 BA  15 EC 0F C0 1F 07 CD F5   ..L>.... ........
00000110  3B 24 A0 E6 03 1D C9 66  9B 80 63 00 CC 42 76 61   ;$.....f ..c..Bva
00000120  08 C1 5B 76 39 96 30 E0  75 64 0B EB 17 20 EA 8E   ..[v9.0. ud... ..
00000130  89 B9 4F B2 5D 5B E3 47  D8 DE E4 4D 3C 4D 1F B5   ..O.][.G ...M<M..
00000140  3C 08 77 37 41 4E 39 92  A8 E8 8D 55 A8 7E 91 37   <.w7AN9. ...U.~.7
00000150  6B A1 46 AC 4B 6C D6 DA  48 57 2A 5B A6 B0 6C 67   k.F.Kl.. HW*[..lg
00000160  15 CC D1 64 05 06 AD 48  2A F3 D2 43 E8 0E 1E B4   ...d...H *..C....
00000170  63 E9 65 CA DD 58 D0 6B  5B 9D 3A 98 4C FA 50 61   c.e..X.k [.:.L.Pa
00000180  F8 36 4F 32 CD 0B 48 A8  A6 EF 71 6E 85 C0 FC 96   .6O2..H. ..qn....
00000190  A5 B5 52 E8 AA E3 71 E3  8F 31 CB 9F ED 5D E3 B4   ..R...q. .1...]..
000001A0  93 00 D7 80 73 0A C2 39  62 53 35 39 72 52 FA 57   ....s..9 bS59rR.W
000001B0  ED 92 4D 70 30 D1 70 C0  F0 23 94 9C 55 B4 F0 07   ..Mp0.p. .#..U...
000001C0  C3 5E DA 15 2F B1 44 16  74 C8 4A FC CA FA 0B 2B   .^../.D. t.J....+
000001D0  78 59 B8 68 25 43 DC B5  BE B0 0F 10 CB E0 06 38   xY.h%C.. .......8
000001E0  95 B8 84 B7 EF DE AD 94  4C 94 60 7B 36 EA 59 B5   ........ L.`{6.Y.
000001F0  EC FB E5 13 92 EE EB C7  91 AC B1 83 46 AE 03 67   ........ ....F..g
00000200  15 11 C8 E7 6E E0 09 52  01 67 D1 AB 2B 6E 60 D1   ....n..R .g..+n`.
00000210  B0 B2 A7 7F 35 14 2B 59  5D C0 6A E2 61 9F 29 A6   ...5.+Y ].j.a.).
00000220  39 BC C7 98 89 CB 9C EC  C1 95 89 BA 8A A4 C7 F7   9....... ........
```

```
00000230  0B F5 E7 A4 2D 9F 3E 85  26 DA EA E5 12 BD 0D F7   ....-.>. &.....
00000240  A2 70 13 E4 F2 40 74 5E  FA DE                    .p...@t^ ..

00000000  17 03 01 00 20 1A 8A E2  BF AF ED 2C BF E1 93 55   .... ... ...,...U
00000010  CD 3B 9F AE A2 78 E8 29  CD 46 9B A4 14 D1 1D EE   .;...x.) .F......
00000020  5B 5F 4F D4 52 17 03 01  02 20 86 D6 9F 0E B5 C6   [_O.R... . ......
00000030  25 AC 35 A7 60 DA C6 54  74 6C 5F C4 AD B2 A2 D5   %.5.`..T tl_.....
00000040  77 D6 0F 71 09 20 77 2D  0D 3B 81 91 82 84 76 38   w..q. w- .;....v8
00000050  3C 34 A3 31 02 96 E7 C2  EC 60 D7 FF 71 D4 D3 93   <4.1.... .`..q...
00000060  AF A3 6F 13 C1 7C 11 38  E0 E2 61 93 29 36 F3 3B   ..o..|.8 ..a.)6.;
00000070  5B 4F 4E 74 BB 2E F2 67  F5 4C 57 32 59 FD 27 29   [ONt...g .LW2Y.')
00000080  65 64 29 CC B8 03 6C 88  6D 4B D7 3E E8 21 2F 12   ed)...l. mK.>.!/.
00000090  B1 B3 4F 60 86 19 5F FF  8D E7 6B E0 59 83 48 90   ..O`.._. ..k.Y.H.
000000A0  C4 B2 A9 9F 58 37 0C 49  D5 3A BB C9 78 3D DF DD   ....X7.I .:..x=..
000000B0  05 1A 35 9A 2E 68 35 72  C7 CA A6 BB FD 82 4F 07   ..5..h5r ......O.
000000C0  94 D7 8B 68 B1 56 C4 13  57 8C CF 70 12 4F D1 80   ...h.V.. W..p.O..
000000D0  0E BF 44 3C CC 79 24 6F  E3 37 F0 A3 B6 12 A0 BC   ..D<.y$o .7......
000000E0  C6 E5 0C 9A B1 39 66 65  1F 6F CE 08 E0 CF B9 2F   .....9fe .o...../
000000F0  95 AB 3D A5 45 49 29 E6  AC FE 06 B9 94 19 12 0E   ..=.EI). ........
00000100  11 F0 5C 1C 7D 61 68 F8  EF 12 2E 0D 3E 89 74 A1   ..\.}ah. ....>.t.
00000110  6D EE 88 25 1E A7 A8 03  E5 89 E2 EF ED 53 B1 18   m..%.... .....S..
00000120  B8 64 62 32 BF 6C D3 3D  1C BC 8F FC D7 24 BF AE   .db2.l.= .....$..
00000130  61 4E B7 97 BA CB 5D 16  3F 5A 9A CE 4D DB DB 25   aN....]. ?Z..M..%
00000140  BA 85 B5 39 E6 AA 4A 44  89 46 6F 19 49 A8 BA 03   ...9..JD .Fo.I...
00000150  DF F5 58 28 0F EE 2D CA  E9 CC BC 36 B1 E4 A1 43   ..X(..-. ...6...C
00000160  72 DE 27 2C B4 41 E8 E9  61 C7 58 21 0E C3 B3 E6   r.',.A.. a.X!....
00000170  31 DA D9 3E 2C 04 A0 88  51 CE D4 AA F4 D2 6B 8A   1..>,... Q.....k.
00000180  D9 51 87 EE DB 86 9A 4E  CD 41 93 11 19 46 79 04   .Q.....N .A...Fy.
00000190  A4 CF 2B 6E 0F 23 38 39  54 E3 E4 66 1F 5C E9 33   ..+n.#89 T..f.\.3
000001A0  C9 2D B9 F5 2D AA 1E 8F  E9 E6 55 D2 86 DF 3F CA   .-..-... ..U...?.
000001B0  1C 19 75 39 2C 29 5C D9  5B 09 54 58 47 13 89 7B   ..u9,)\. [.TXG..{
000001C0  8A 3C 49 DF 36 0D 72 E1  F9 10 53 48 DF AE CD 2E   .<I.6.r. ..SH....
000001D0  0C C5 8E 3A 59 31 C5 3E  D8 AA EF 9B 88 E3 88 3A   ...:Y1.> .......:
000001E0  AB 8A 55 7C F3 6E C8 D0  94 B8 CC 09 06 A8 6F 61   ..U|.n.. ......oa
000001F0  C9 E5 E9 B9 C6 28 D9 3D  F0 96 8D C0 9C 29 42 C0   .....(.= .....)B.
00000200  F8 FB 65 66 BF D6 1D B8  FB 52 A5 7B D9 75 95 36   ..ef.... .R.{.u.6
00000210  BE DB 20 75 D4 AD D9 62  86 CD 54 5B 77 55 EC AB   .. u...b ..T[wU..
00000220  88 42 10 B0 FE 7B 6B 28  3C 9E 6A A4 1F AC 7A E9   .B...{k( <.j...z.
00000230  B4 84 FE 31 E0 A1 79 2B  74 CD 6F 47 56 6B 76 47   ...1..y+ t.oGVkvG
00000240  04 7D A1 D6 08 E8 88 44  3F 53                    .}.....D ?S

00000000  17 03 01 00 20 9E 56 A4  62 FC 14 BD 5F 06 BC 69   .... .V. b..._..i
00000010  16 77 A7 19 4F 49 45 D6  B7 84 22 9C DD 66 A1 07   .w..OIE. .."..f..
00000020  C2 17 84 1E DA 17 03 01  02 20 3E FA 52 BC 42 3E   ........ . >.R.B>
00000030  C3 F6 5C F1 64 05 96 4D  30 A5 0D 79 51 9C 8D C3   ..\.d..M 0..yQ...
00000040  B2 D7 46 F6 7C 6B A8 BB  8D E5 A1 1E FB 9E D2 A3   ..F.|k.. ........
00000050  F0 40 AA D2 41 98 E6 93  62 64 C3 70 30 BC 7F AB   .@..A... bd.p0...
00000060  1F 1F 53 2E 49 38 82 75  5D DD F2 4B A7 EB 83 C7   ..S.I8.u ]..K....
00000070  DE F9 19 04 73 87 A9 9A  02 D0 BC 66 C7 88 D2 E6   ....s... ...f....
00000080  FB EB 3E 18 B0 39 AA B0  98 85 B4 E3 14 5F B1 D0   ..>..9.. ....._..
00000090  06 19 CF 92 B3 94 99 DE  7B C3 5C 91 7C EE 97 23   ........ {.\.|..#
000000A0  85 15 4A 33 00 21 9A 33  12 0D 72 26 4F C2 83 D5   ..J3.!.3 ..r&O...
000000B0  3A 41 99 CF 91 38 78 AE  40 ED DE DC CF D5 28 5D   :A...8x. @.....(]
000000C0  50 E5 8D 4F 91 F4 36 7C  0A 3F 95 F1 C8 6A EE 7B   P..O..6| .?...j.{
```

```
000000D0  AB 74 12 54 0D FD C9 FC  6A C7 BB 36 F7 70 FB F0   .t.T.... j..6.p..
000000E0  D2 02 BE 2B 97 CF AB 9C  75 CE 93 F1 7A FE 82 10   ...+.... u...z...
000000F0  19 89 2A 19 E5 51 36 B4  37 DC 61 43 41 DC 1F 21   ..*..Q6. 7.aCA..!
00000100  3E 66 8C C9 D4 3D 84 C4  5B FE 9B 36 D4 B0 B1 41   >f...=.. [..6...A
00000110  3C F6 19 2D EA 77 D4 75  9C B8 55 DC 64 02 DE FB   <..-.w.u ..U.d...
00000120  04 6E AB 78 33 D6 1D DF  0B 22 E3 1D F7 8E C6 B4   .n.x3... ."......
00000130  D4 9D 96 91 5A 3E 4C 02  37 C9 76 2F A8 22 1F 55   ....Z>L. 7.v/.".U
00000140  D1 B4 4E 2D 1E B5 71 C9  C9 9D 02 81 51 DD B0 2B   ..N-..q. ....Q..+
00000150  8B 39 77 A8 71 5E F3 F3  96 1C 04 ED 9F 55 D3 85   .9w.q^.. .....U..
00000160  E0 DD 87 63 70 08 02 B0  A7 09 C2 E4 C8 30 15 1C   ...cp... .....0..
00000170  80 66 31 0B 1B 77 C6 CF  95 7D AA 26 45 8D 80 0E   .f1..w.. .}.&E...
00000180  64 2F 71 73 E9 7F 7D D5  35 50 38 1B 4A 5B 32 07   d/qs.}. 5P8.J[2.
00000190  4E 5F 5A 3A A7 5C 20 88  CD 5C 69 F9 4D 70 D0 9E   N_Z:.\ . .\i.Mp..
000001A0  32 57 BB 09 73 E3 3D E4  F5 4B 32 68 4B 47 5F 71   2W..s.=. .K2hKG_q
000001B0  4A 6F 71 D9 FF D1 E8 FD  52 1B 04 6D D3 93 64 5F   Joq..... R..m..d_
000001C0  43 49 C9 7F B8 56 41 56  C2 68 3A C8 BD EA D3 B5   CI..VAV .h:.....
000001D0  7F 9C 65 82 69 DA 2B A0  E5 73 F5 9B E4 0F 0E 1B   .e.i.+. .s......
000001E0  D3 2C AB 52 BA 11 DA D4  25 75 5D 90 EB 6A E8 80   .,.R.... %u]..j..
000001F0  55 81 9B 0B C5 3E A4 A5  18 D2 24 BF 8A A3 71 4A   U....>.. ..$...qJ
00000200  1E 16 41 DB 5B 24 5A B1  40 0B C3 90 81 60 D8 88   ..A.[$Z. @....`..
00000210  CB 7D 11 E7 27 90 47 86  13 DA D5 D0 91 1C 06 E3   .}..'.G. ........
00000220  6A 07 86 FB AE A3 5F B9  56 D6 76 89 C6 3C 97 67   j....._. V.v..<.g
00000230  14 47 5B DC 30 E2 B0 44  30 4A D3 D9 D2 0A 95 E0   .G[.0..D 0J......
00000240  CD 6D 2E 9B C3 86 78 0E  CF 66 17 03 01 00 20 06   .m....x. .f.... .
00000250  D6 E6 9E 9A 1D 56 B1 CF  D1 C5 9E 8A 93 B7 19 EC   .....V.. ........
00000260  BC 7C 4D A4 B7 46 E8 AB  C3 F9 3B 08 30 5F 72 17   .|M..F.. ..;.0_r.
00000270  03 01 02 20 9B E6 9F 6C  4B 24 BA 44 50 D0 87 58   ... ...l K$.DP..X
00000280  2C 03 6A 94 76 F4 81 09  86 9D DC D3 E9 CF F9 7B   ,.j.v... .......{
00000290  FC 8F D7 E7 C0 5A 0F 6C  9E 0F 42 8C 63 FB 01 3C   .....Z.l ..B.c..<
000002A0  8F BA E5 AA 12 8E 04 98  05 6A BE AC 59 69 07 C4   ........ .j..Yi..
000002B0  0B AC 65 AC A3 13 A7 12  86 9A 56 15 C3 D2 D6 4F   ..e..... ..V....O
000002C0  57 E8 01 6E 6E 61 20 C7  6F A2 F8 03 11 C3 03 05   W..nna . o.......
000002D0  F8 54 63 99 0E 55 39 9C  A5 02 D5 A7 78 E8 5C 99   .Tc..U9. ....x.\.
000002E0  DF E4 8D 18 87 60 4C 8C  EE 9D F6 8A 98 97 5C FA   .....`L. ......\.
000002F0  71 46 37 6A 82 34 13 BA  1F D7 87 33 6C 92 37 71   qF7j.4.. ...3l.7q
00000300  EA 93 65 A1 1E 54 61 78  9F 3A 54 B2 AD 0D 9C A2   ..e..Tax .:T.....
00000310  F7 DE 72 85 32 98 8B 86  DE BF B6 C7 4F 6D 17 4D   ..r.2... ....Om.M
00000320  0B C1 12 42 C6 91 4C 03  0B 85 C3 D8 71 F2 04 6C   ...B..L. ....q..l
00000330  85 6C 85 82 6B FD 8D EF  93 03 EE 66 03 CF FF C2   .l..k... ...f....
00000340  E6 C0 4B 22 6C 26 49 4F  45 19 24 FF 9A A5 0F 6B   ..K"l&IO E.$....k
00000350  90 AE CB 9D BC 18 9C D0  3F E6 15 90 F1 86 74 08   ........ ?.....t.
00000360  C9 70 3C 85 49 9D AC CC  D2 BF 87 E5 B2 4F 20 DA   .p<.I... .....O .
00000370  E5 2A AF AC 30 4B E0 64  A4 B2 ED 68 67 40 26 31   .*..0K.d ...hg@&1
00000380  69 1E 13 0B D3 D6 47 DE  0E F6 60 F2 A1 41 E8 BE   i.....G. ..`..A..
00000390  49 65 94 EA 2F 58 2C 54  77 FD BB EE EF AE 74 5C   Ie../X,T w.....t\
000003A0  E8 5C D5 78 23 CE 15 DE  74 8B 96 6E 45 89 B5 79   .\.x#... t..nE..y
000003B0  13 9E 28 14 3E 88 F3 8C  8F A7 37 FA 8D 24 48 80   ..(.>... ..7..$H.
000003C0  22 25 42 94 B8 11 40 2A  41 E3 5F FD 9E B3 2C 74   "%B...@* A._...,t
000003D0  64 41 90 FA 65 64 2D 59  7D 6B 72 DD 22 6E 17 BB   dA..ed-Y }kr."n..
000003E0  87 33 1D 53 84 B3 F7 C9  F4 FE 24 68 E8 AF FD C5   .3.S.... ..$h....
000003F0  67 B5 6F DC 9C 72 AE C1  ED 07 D9 3E 6B 59 A4 BD   g.o..r.. ...>kY..
00000400  94 70 AC 62 91 B7 F9 52  B9 F9 F6 FD 6B 0A 43 51   .p.b...R ....k.CQ
00000410  22 5D 1A 43 4B 83 6C 1A  80 08 90 F6 E8 D3 A9 8A   "].CK.l. ........
00000420  B6 15 7F D6 0F A8 B9 70  83 AD 93 3A 38 8E 50 B2   .......p ...:8.P.
```

```
00000430  B3 63 57 A3 C1 46 EB 36  90 99 03 2B EE F1 FB 92   .cW..F.6 ...+....
00000440  D9 C1 B2 01 28 21 15 B5  50 BA 05 89 03 33 7C 79   ....(!.. P....3|y
00000450  0B 75 49 90 5D 96 65 87  46 BE 78 6B F8 FC 0A E4   .uI.].e. F.xk....
00000460  F8 64 56 8F E6 F5 9F A0  E6 61 2F 44 37 DD DF 4D   .dV..... .a/D7..M
00000470  BC 8D 59 BF 79 2A 0D 82  C5 AD 1D D8 E8 6B F0 4D   ..Y.y*.. .....k.M
00000480  10 24 D5 DF FD 4E F6 25  4C C3 9E 60 04 87 32 FE   .$...N.% L..`..2.
00000490  6B B4 A5 B3                                       k...

00000000  17 03 01 00 20 86 0A 59  83 22 2D 4C B3 A5 31 B6   .... ..Y ."-L..1.
00000010  47 FB D5 52 35 6E 06 C1  99 E6 06 27 70 44 BF FC   G..R5n.. ...'pD..
00000020  CC FE B8 91 C0 17 03 01  02 20 00 A9 A3 71 36 8F   ........ . ...q6.
00000030  24 02 BC 25 80 FD BD 7B  FB 3B 19 DE CF CD 81 2A   $..%...{ .;.....*
00000040  86 A9 A7 AD 4A 4D F2 D7  2D 0D 55 83 59 A8 35 4E   ....JM.. -.U.Y.5N
00000050  D1 60 92 D4 07 C5 C3 62  C4 4A F2 BA 43 39 DB 94   .`.....b .J..C9..
00000060  75 C0 0A A5 E2 76 8F 95  8B D6 8B 0C ED 4E D7 C0   u....v.. .....N..
00000070  49 7D 75 66 AF B2 9A 95  BD 52 88 6E 79 FE 0D 6B   I}uf.... .R.ny..k
00000080  05 0E 3F 57 4C 41 C1 E2  F3 AB 1E 4B 7F E0 5E A3   ..?WLA.. ...K.^.
00000090  76 77 2E 6B 2F FD E5 85  2C E0 89 AA 79 2E 8E 88   vw.k/... ,...y...
000000A0  24 A9 18 6B 6F E8 4D 6D  71 01 7F 97 61 FD 64 76   $..ko.Mm q..a.dv
000000B0  B6 C9 70 7E 23 0A 8D 88  A7 4D 65 43 27 8B B9 67   ..p~#... .MeC'..g
000000C0  8D 5C 3E 96 8D D4 0A 5D  E8 B0 5C 84 6B 02 16 A4   .\>....] ..\.k...
000000D0  26 48 D7 CE DA A0 42 55  D4 39 C5 B7 72 FF BB 5A   &H....BU .9..r..Z
000000E0  2E 99 01 22 5F 7D 06 76  8C 33 EF D0 95 26 C7 B4   ..."_}.v .3...&..
000000F0  95 3E C0 0E 5D C0 A5 B0  44 76 2A 74 14 D3 AF E4   .>..]... Dv*t....
00000100  AD 53 FC A1 18 8C A1 21  47 3B D0 C3 17 94 FD 5A   .S.....! G;.....Z
00000110  02 10 DB 98 12 39 B0 EB  CF FD 11 36 D1 BD 8C 8C   .....9.. ...6....
00000120  7A D7 3B 58 23 44 54 42  DC 22 79 C2 17 99 BA 8B   z.;X#DTB ."y.....
00000130  05 CC 81 58 EE 35 3B ED  30 1C 67 4F 51 5B 79 45   ...X.5;. 0.gOQ[yE
00000140  DB D6 4E 92 23 52 38 E1  5F 86 BC B0 6C BC 69 EB   ..N.#R8. _...l.i.
00000150  BA 4D 85 03 45 06 08 DE  4A 84 50 AE 5C AC F7 63   .M..E... J.P.\..c
00000160  16 01 72 38 BB 1F 01 C8  A0 A5 56 62 70 E1 CF 5B   ..r8.... ..Vbp..[
00000170  4D 30 E7 22 11 79 4B 5A  25 2C CB 70 E6 9A EA 61   M0.".yKZ %,.p...a
00000180  44 9F 6B 36 B8 AE 91 2C  74 08 87 46 2E 8F 64 49   D.k6..., t..F..dI
00000190  C6 77 31 92 44 D3 86 F4  9C AF A3 E5 95 FD CF D9   .w1.D... ........
000001A0  B8 14 76 DD A5 E1 09 50  4F AE A3 EE C1 E0 17 B4   ..v....P O.......
000001B0  B1 38 00 CB 04 B9 10 F7  F0 0B DB 38 45 84 91 8D   .8..... ...8E...
000001C0  C8 7F C1 EB E2 9B E0 D2  70 BE 42 87 79 05 1A CB   ....... p.B.y...
000001D0  DA E0 B8 C6 58 AC B6 87  65 02 05 A9 42 81 28 26   ....X... e...B.(&
000001E0  2D 8E 94 C9 14 72 90 35  5E 14 FE A4 B6 7B BC 7E   -....r.5 ^....{.~
000001F0  A9 69 D1 37 16 FD 7C A4  A4 34 9E C5 BD 5F EC C5   .i.7..|. .4..._..
00000200  92 5D D1 CB A6 CE FD 6D  F7 7C 05 FD 4D 07 BC 32   .]....m .|..M..2
00000210  11 71 BE 2B F5 78 B7 7B  21 21 68 9D 55 F6 53 B4   .q.+.x.{ !!h.U.S.
00000220  58 B3 5C 6C 80 3C F0 50  CD 5B F8 05 EC DB 2B F2   X.\l.<.P .[....+.
00000230  C7 5F A0 97 32 EC 5F F7  FD 6F F6 D7 C0 99 5E E5   ._..2._. .o....^.
00000240  85 9B C6 08 3A 10 7A 11  A2 82                     ....:.z. ..

00000000  17 03 01 00 20 3F DD 48  5B 65 88 D1 97 1B 40 E2   .... ?.H [e....@.
00000010  F9 B5 98 CE 89 68 61 0E  52 9F FA 1D 46 3F 4B 7B   .....ha. R...F?K{
00000020  7F E5 91 5E D5 17 03 01  02 20 13 AA E0 97 D6 AC   ..^.... . ......
00000030  C2 A8 15 60 67 C4 A9 68  F4 43 15 02 31 B1 B5 C9   ...`g..h .C..1...
00000040  96 8C 74 CD 23 50 18 B4  93 01 D2 68 B6 97 92 91   ..t.#P.. ...h....
00000050  6C AE C4 2F 99 F0 DC 0F  5C 8F 2D E8 EF B6 FC 97   l../.... \.-.....
00000060  1C 47 58 67 FA 67 23 18  D8 47 18 EE 78 B0 F4 38   .GXg.g#. .G..x..8
00000070  6D 40 E9 D0 A2 D2 3C B1  2C 2F 09 94 28 D1 C4 FC   m@....<. ,/..(...
```

```
00000080  36 4C 56 4D 04 5D 90 B7  03 BD B0 A9 87 3E 21 A0   6LVM.].. .....>!.
00000090  F3 42 71 0A D4 FD 69 FF  74 92 F4 FD 8B 81 29 76   .Bq...i. t.....)v
000000A0  B5 BF 2D 0F 03 E7 C8 47  0D 8D 8F 6A 72 9D 0C 98   ..-....G ...jr...
000000B0  AC 94 5D 15 84 2B 0C D4  EF EC 8A 49 21 18 11 93   ..]..+.. ...I!...
000000C0  42 14 AD 3D F3 88 A4 14  26 F1 BD 1B 11 C9 03 85   B..=.... &.......
000000D0  C8 85 9D 80 04 95 A0 AE  F1 33 E9 59 00 3D 49 70   ........ .3.Y.=Ip
000000E0  FF 71 4C F6 E4 51 56 0C  C0 E3 6B 77 0E 4A 0A C4   .qL..QV. ..kw.J..
000000F0  7D 02 C4 A1 AE 44 C2 81  7B 8D AC FF 65 C8 23 E1   }....D.. {...e.#.
00000100  5B FF E1 81 91 8C F3 F2  AC 2D 93 86 29 D0 6F 69   [....... .-..).oi
00000110  85 94 AC 96 04 58 DA E8  73 99 95 41 51 25 5E 51   .....X.. s..AQ%^Q
00000120  E9 1C 2B 0A AC C8 2F 39  AD E7 C1 8F 98 ED FF C4   ..+.../9 ........
00000130  A2 F1 08 76 A7 C3 EA 17  A3 E6 B9 B4 E0 63 C4 88   ...v.... .....c..
00000140  C6 43 1D 86 B7 63 FB C3  4E 48 BC EF A8 C1 75 5A   .C...c.. NH....uZ
00000150  35 D3 6C 6E 8E 6A 5F EA  57 CC 3B 38 92 EA BF 70   5.ln.j_. W.;8...p
00000160  7F 86 68 83 0D 96 77 7D  F8 1A E9 D5 EC 38 84 A9   .h...w} .....8..
00000170  01 DF 44 94 15 D4 95 8E  15 21 97 18 5C 5F 77 61   ..D..... .!..\_wa
00000180  72 7B 86 BE 95 39 50 36  7E 48 C1 1E EF C6 94 1C   r{...9P6 ~H......
00000190  14 6E 9D 5C 95 2B 8F CA  23 BD 6F C0 D5 C7 D7 DC   .n.\.+.. #.o.....
000001A0  DA 41 BC 11 AB FD E9 EE  E6 F5 73 61 75 5A 4E 43   .A...... ..sauZNC
000001B0  A2 24 AD 1C 73 C9 C3 3F  A6 F5 7B 22 3E 31 D8 75   .$..s..? ..{">1.u
000001C0  08 E4 D2 49 CD 36 E3 6E  91 63 8D C5 BD D7 C7 9A   ...I.6.n .c......
000001D0  E1 5A 83 74 C9 12 65 42  C6 47 8B B3 CE FD 8E DE   .Z.t..eB .G......
000001E0  35 1B 6A 66 0D D7 F0 4C  A7 52 74 08 EA 35 57 1E   5.jf...L .Rt..5W.
000001F0  F1 18 CD 27 7D C4 3B 74  AE 40 F9 C5 EB 81 B9 A2   ...'}.;t .@......
00000200  32 FF C5 1E 9E BE AB D2  5E 15 DD DA AD 57 12 59   2....... ^....W.Y
00000210  3C A6 DA 14 02 1C 83 76  54 9B DA B7 9F B0 72 61   <......v T.....ra
00000220  05 85 6B 03 DD 67 40 4C  FD CE 92 27 90 A8 63 51   ..k..g@L ...'..cQ
00000230  18 30 95 44 F5 CD 2E 5F  58 D8 49 36 AB 17 DC 30   .0.D..._ X.I6...0
00000240  F7 A4 85 A4 3A 08 CD 19  A9 7E                     ....:... .~

00000000  17 03 01 00 20 52 6F 0F  E3 A8 24 0F 47 87 B4 00   .... Ro. ..$.G...
00000010  35 DB 12 42 5F 4F E7 1E  8D 0C ED 3E DE B6 0C AA   5..B_O.. ...>....
00000020  4F 68 8A 2A 35 17 03 01  02 20 07 5C C0 4F CB 28   Oh.*5... . .\.O.(
00000030  90 36 60 AC 89 43 C8 74  B8 C3 A6 F7 0E 88 A7 D8   .6`..C.t ........
00000040  2F 1C 7C 6C A7 21 BE 5A  87 F4 3C D8 B5 3B B9 10   /.|l.!.Z ..<..;..
00000050  6A CB 18 B0 23 68 61 F0  FC 22 E6 2E 0A E4 8A 95   j...#ha. ."......
00000060  F4 9F DD 0E 40 50 B0 A2  2D 67 E8 7B 0B D1 D2 DA   ....@P.. -g.{....
00000070  55 8A 5B 9F 94 20 57 18  46 F9 FB 4B 0B 97 60 E7   U.[.. W. F..K..`.
00000080  05 D3 72 29 70 AC 63 7E  3E 91 28 34 D1 FE A9 32   ..r)p.c~ >.(4...2
00000090  C7 8F 63 3F E5 7F 1D DB  51 C7 0C 47 08 F9 F9 7F   ..c?... Q..G...
000000A0  4D 8C 0B 63 C7 73 6C C3  B4 36 5F F6 78 A6 1D C4   M..c.sl. .6_.x...
000000B0  3D 0E AF E9 88 67 49 11  99 1A A2 C8 19 08 70 8F   =....gI. ......p.
000000C0  CF D7 FC C0 4E DF 36 86  EA 00 63 FD 71 89 BE BA   ....N.6. ..c.q...
000000D0  8A CC 89 14 E5 F8 5B 36  87 DE C3 3C 84 F0 92 1C   ......[6 ...<....
000000E0  2F DD A7 05 84 CD 85 67  78 A9 C9 A7 D0 8C FE D2   /......g x.......
000000F0  DA 72 F3 91 39 0F F4 35  D0 54 0B F9 B1 53 DD 00   .r..9..5 .T...S..
00000100  21 DB FC CB 5D 78 EA BC  8E 85 4D 30 F3 2B 9A 42   !...]x.. ..M0.+.B
00000110  2F 7A AC A1 24 E4 94 AD  E0 B9 52 DB DD 6F D5 45   /z..$... ..R..o.E
00000120  99 84 3B 70 3D E5 7A 1D  2E BD 63 82 64 C0 2E F2   ..;p=.z. ..c.d...
00000130  8C 9E E4 AE AE 13 9C E1  37 DA 93 25 9C C0 DE BA   ........ 7..%....
00000140  76 69 CB 5B 28 4C 34 B0  A5 F8 D0 CA D5 2F 0F 56   vi.[(L4. ...../.V
00000150  97 20 5C D0 78 E7 E8 66  02 90 64 DD C2 D7 0B 3E   . \.x..f ..d....>
00000160  4F FD 83 BB BB BD C8 79  3E 04 BA BA 65 BC EA B5   O......y >...e...
00000170  4F 1C 63 38 39 83 D4 19  B4 61 C2 BC 09 E6 2A 47   O.c89... .a....*G
```

```
00000180  EF 91 70 AA 91 D3 76 29  C0 5F C5 35 6B 3F BD CF   ..p...v) ._.5k?..
00000190  C6 F8 07 D8 57 AE DF B2  6D 69 A1 1D 40 30 E2 15   ....W... mi..@0..
000001A0  E6 A1 62 84 92 62 0A ED  55 A0 D3 A9 08 6E DB B8   ..b..b.. U....n..
000001B0  78 06 C6 44 C8 30 99 E7  E6 D6 28 E0 6D 14 33 15   x..D.0.. ..(.m.3.
000001C0  56 32 92 76 51 44 B5 C0  AD 02 63 77 16 2D A5 79   V2.vQD.. ..cw.-.y
000001D0  DB F1 7B 54 20 59 AA E0  F1 FC 97 C6 41 62 40 5F   ..{T Y.. ....Ab@_
000001E0  10 AA B3 F6 C2 59 E2 25  FB F2 EA C4 D1 02 09 9C   .....Y.% ........
000001F0  E7 37 83 63 46 FF F6 A6  FB 25 23 EB 6C C7 94 75   .7.cF... .%#.l..u
00000200  75 AB 86 C4 D1 0D 1F A3  AA 4C A2 59 89 B2 52 13   u........ .L.Y..R.
00000210  20 FF 7F 04 62 E7 5F 42  90 48 8A AD EA 79 2A 62    ..b._B .H...y*b
00000220  60 99 18 0A 95 5B F9 0A  F4 73 A3 A4 E8 61 56 ED   `....[.. .s...aV.
00000230  C1 E4 DD C5 92 86 59 4D  50 87 B6 A1 3C EE B9 EB   ......YM P...<...
00000240  8B 49 55 9C D7 DA 08 4C  92 6C                     .IU....L .l

00000000  17 03 01 00 20 E8 9B 9D  F0 F2 00 B8 76 F9 60 64   .... ... ....v.`d
00000010  9A 7E 26 43 95 7C 21 82  38 E5 04 48 F0 D6 49 58   .~&C.|!. 8..H..IX
00000020  A5 D4 87 21 E7 17 03 01  02 20 95 F7 F3 25 10 70   ...!.... . ...%.p
00000030  B0 FC B0 2F A5 9B 45 8E  43 70 3F 31 63 4E 80 A6   .../..E. Cp?1cN..
00000040  45 2A A0 DC 9A BD F2 03  9C 3E F5 FE 33 0A 4B 3A   E*..... .>..3.K:
00000050  81 93 54 C4 A3 B7 59 2D  77 05 8D DB 39 F9 64 77   ..T...Y- w...9.dw
00000060  90 58 11 BA 4E 63 53 31  B3 0F D9 26 54 5F E0 56   .X..NcS1 ...&T_.V
00000070  69 97 CC B4 0B A7 A0 2E  A8 5F 59 15 84 CE 6B A2   i..... ._Y...k.
00000080  73 95 77 2C F2 5C 21 6F  67 B6 06 C5 3B 6C C0 40   s.w,.\!o g...;l.@
00000090  E7 CC C6 51 29 DC 45 33  53 7F D2 1E 52 29 F7 22   ...Q).E3 S..R)."
000000A0  BD 34 8F C4 9B 33 2D D9  1F E4 56 EA BE 0E 04 4D   .4...3-. ..V....M
000000B0  A3 E6 7E 66 AB 16 AC FB  BF 6A 5C C7 06 24 DC B8   ..~f.... .j\..$..
000000C0  AB 27 54 91 5E 9E 17 A5  91 CE E0 F2 F1 FD D7 03   .'T.^... ........
000000D0  44 09 44 E4 57 AB D0 27  29 30 A7 D7 04 AD D2 98   D.D.W..' )0......
000000E0  70 1A 9F DE DA 2F D5 4A  C7 39 B9 B0 E2 19 1B 0C   p..../.J .9......
000000F0  D7 27 AD CF 9F AF 35 62  BF D1 4A C1 72 83 75 9D   .'....5b ..J.r.u.
00000100  1F A5 9C 75 06 D5 3F 40  17 89 A1 EB 69 DE BD 24   ...u..?@ ....i..$
00000110  96 E5 AC 98 94 95 8D 05  59 F1 6B 43 D2 72 5D B7   ........ Y.kC.r].
00000120  17 C5 B1 F6 4B 04 D5 11  ED 2B 3C A7 C5 71 91 A4   ....K... .+<..q..
00000130  D0 D3 3E 75 8B C0 3C FD  56 16 D8 70 1B D2 6C 89   ..>u..<. V..p..l.
00000140  88 B9 3B B5 88 B8 59 00  28 02 8E 1D 3E 21 10 1A   ..;...Y. (...>!..
00000150  74 4B 86 1E 87 8A 7C 0F  3F 80 E5 79 DD E7 7F B1   tK....|. ?..y...
00000160  2B F5 CA 13 84 8A 53 D4  24 0D 3B EE 5A AC 3A FC   +.....S. $.;.Z.:.
00000170  BD E1 48 72 36 00 9A 2D  AD 38 96 12 8C 19 32 5C   ..Hr6..- .8....2\
00000180  41 14 AD BD 20 30 A1 59  FD 07 12 D1 0A A6 F6 20   A... 0.Y ....... 
00000190  2A 4B 7C 82 C0 A6 90 96  7F E3 D0 B6 12 64 B4 B0   *K|..... ....d..
000001A0  FB 7D BC 2B 1E 0B 73 54  59 44 DC 72 2F 64 0F 3F   .}.+..sT YD.r/d.?
000001B0  2C 15 3F E4 49 69 54 57  8A D1 29 F6 61 51 0C 17   ,.?.IiTW ..).aQ..
000001C0  73 13 A6 61 00 A2 4C E9  2D D5 3E A2 46 A7 F7 DC   s..a..L. -.>.F...
000001D0  C3 19 1A 3F CD A7 E1 05  C7 DF CA 07 95 61 32 FB   ...?.... .....a2.
000001E0  04 AB 43 7C 61 3D EB 7B  9C E9 6D 19 0B 51 75 19   ..C|a=.{ ..m..Qu.
000001F0  B8 B0 BB 25 59 57 AF 03  BF 80 07 B6 2D A2 E9 A9   ...%YW.. ....-...
00000200  44 17 A4 E1 5F 57 ED 07  A3 33 A4 C2 7F 5E 9F 93   D..._W.. .3..^..
00000210  E8 99 6A 45 2F AE 7D 04  60 E3 3F 5E 17 0B 5E 85   ..jE/.}. `.?^..^.
00000220  A6 39 DE B1 4D 77 71 16  C2 AF 0A 54 30 C9 90 36   .9..Mwq. ...T0..6
00000230  86 D3 83 64 D6 D4 D4 ED  DA 90 CE 18 0F E8 A9 11   ...d.... ........
00000240  80 9A D8 3A 1B 1D 91 A9  8A 5D                     ...:.... .]

00000000  17 03 01 00 20 F2 32 0A  3D 36 A6 14 CC 6F 92 5D   .... .2. =6...o.]
00000010  8F F9 EE D7 4A 17 98 C5  BE E9 CA 7D 73 D2 14 B4   ....J... ...}s...
```

```
00000020  CC 00 46 B1 13 17 03 01  02 20 5A FC 48 A4 22 64   ..F..... . Z.H."d
00000030  8B 9A 3F D3 5A BF FE A8  9E F6 8E 6E 68 E7 0B 46   ..?.Z... ...nh..F
00000040  2B 7F 6E 87 75 16 95 5A  8C B4 F1 66 1E A2 A4 34   +n.u..Z ...f...4
00000050  FB 4C 40 71 96 CE 6A 1B  B1 4F 94 B6 9E 31 16 13   .L@q..j. .O...1..
00000060  07 95 C4 AE 71 7B 3E A4  BB CB 4F A1 6D B0 71 EF   ....q{>. ..O.m.q.
00000070  46 09 8D 11 97 82 59 FC  E6 60 C9 D9 8C 85 F4 9E   F.....Y. .`......
00000080  AB BD 07 32 BF 8D 33 DB  3B DC 11 67 9A E1 F1 95   ...2..3. ;..g....
00000090  58 4A BD C8 B4 8C DC 68  EF 0F 0B A1 20 56 63 A0   XJ.....h .... Vc.
000000A0  9C D6 6D 03 A2 22 9E 7D  F5 BF 09 BC A8 E8 4C 52   ..m..".} ......LR
000000B0  7E B1 80 3F 80 7D 59 9F  8B A1 FB A8 0F 8E 31 FA   ~..?.}Y. ......1.
000000C0  9A 3A 5B 16 CE DE 41 C3  4F 76 EA 3D 0F A7 0C 13   .:[...A. Ov.=....
000000D0  39 F6 F7 3A 78 0E 73 AF  E3 E9 4E 5F 3C D2 CB D4   9..:x.s. ..N_<...
000000E0  9B 47 69 13 76 52 B6 7D  25 E1 77 C9 CD E0 0E 62   .Gi.vR.} %.w....b
000000F0  B3 2E A9 5B 84 74 79 5B  1E 9A 25 DF 0B BD B5 E5   ...[.ty[ ..%.....
00000100  2E A8 4C 3E A6 D6 80 80  89 6B 76 B2 94 17 BD 4E   ..L>.... .kv....N
00000110  DC 6D D9 50 DF 0A 9F D3  2B 1E B3 9F 94 ED FA 5B   .m.P.... +......[
00000120  F9 55 DC F3 8B 80 5E 88  37 D0 4F 17 7C C9 0A D2   .U....^. 7.O.|...
00000130  16 E4 45 18 A6 31 20 09  6A E7 B4 B0 FD 56 68 DE   ..E..1 . j....Vh.
00000140  C3 56 96 70 DD F3 47 F6  3B FB 93 47 6C 05 E6 0B   .V.p..G. ;..Gl...
00000150  E0 FC 0A DE A2 BA 6F 82  A6 C9 1A 5E 62 F5 58 97   ......o. ...^b.X.
00000160  D7 FC E1 F3 70 97 74 0D  E5 7E EB 98 8A 83 1D F6   ....p.t. .~......
00000170  C5 C5 B9 B2 93 94 04 9F  81 B0 41 FD 28 84 E7 AD   ........ ..A.(...
00000180  18 B1 7A 7F 44 55 25 E1  5E A4 13 B4 52 E6 A1 3C   ..zDU%. ^...R..<
00000190  B5 C9 B6 F4 7D 55 42 FA  97 90 72 D6 72 86 FE 3C   ....}UB. ..r.r..<
000001A0  D9 12 B5 D2 A0 13 E0 6E  A6 BB 9D 92 9C 25 C2 FC   .......n .....%..
000001B0  B7 88 A4 6D 79 CB 93 21  35 3E 9F 54 FB 01 A4 96   ...my..! 5>.T....
000001C0  16 92 1D C0 B7 E6 80 00  F6 49 86 C4 3E 9A 68 BC   ........ .I..>.h.
000001D0  93 53 1E 10 3B 4B 12 F7  0D A0 0A 35 43 C2 9C 27   .S..;K.. ...5C..'
000001E0  D5 68 80 83 B2 3B 30 29  E8 E4 51 4E 5A 6F 65 4F   .h...;0) ..QNZoeO
000001F0  28 2F 03 35 16 C2 0A 40  E0 10 B9 9B 11 76 E8 8D   (/.5...@ .....v..
00000200  62 AE 08 F6 DD 9A 44 B1  A0 49 93 75 7C E9 29 1D   b.....D. .I.u|.).
00000210  11 67 D4 20 C0 23 5B E8  24 85 47 E8 E3 31 4D 8F   .g. .#[. $.G..1M.
00000220  B4 78 4E 23 11 4B 8F 04  07 EB 6B C5 DC 08 19 1F   .xN#.K.. ..k.....
00000230  0D D6 FB 27 69 34 A9 AC  BE 38 36 16 A7 8F 78 4C   ...'i4.. .86...xL
00000240  46 3B C0 39 A6 ED C0 0E  DA C9                     F;.9.... ..

00000000  17 03 01 00 20 32 88 D6  22 7F A2 27 2A 55 2F 7F   .... 2.. "..'*U/
00000010  68 1E D3 EF 0E 27 A9 02  E8 84 89 FC 1F CC 3C 38   h....'.. ......<8
00000020  A4 18 F8 B5 96 17 03 01  02 20 C2 1E DE AA FE 54   ........ . .....T
00000030  C8 4A 70 AE 2F 9B 99 B9  D7 78 B9 74 1E A3 39 A1   .Jp./... .x.t..9.
00000040  C7 CC 41 C0 F1 34 3B A3  1E 98 57 3F CE 30 26 9F   ..A..4;. ..W?.0&.
00000050  C6 66 49 8A 48 CE 73 59  40 2F 31 33 4B 10 06 38   .fI.H.sY @/13K..8
00000060  B8 EC F4 8F 64 41 66 F9  BC 8B 0C 5A 49 90 F6 45   ....dAf. ...ZI..E
00000070  1B 54 34 1A 48 ED 28 96  1F E7 A3 DE A6 FF 98 40   .T4.H.(. .......@
00000080  8A 70 BC 76 CC C1 13 37  D6 2E 5A D0 3D C5 2B 5B   .p.v...7 ..Z.=.+[
00000090  F3 EA BF 22 BB 81 0F 4F  A0 2D 71 3B 7B 0A 07 41   ..."...O .-q;{..A
000000A0  4A 03 7C 50 FA A9 DA D9  8C 64 E7 81 17 88 9B 7F   J.|P.... .d.....
000000B0  AD D3 87 E6 2B 94 C5 96  1A 35 0E 38 AF 6C 8A 3F   ....+... .5.8.l.?
000000C0  06 15 EE 4B 75 58 92 84  58 B3 48 99 6F 43 FC 09   ...KuX.. X.H.oC..
000000D0  B8 52 F2 BB 6E 92 9D 79  FA 75 1F 12 11 F7 C0 C6   .R..n..y .u......
000000E0  30 A1 A5 BD D8 F9 4E A1  EB 83 9F 47 C7 FE 81 C0   0.....N. ...G....
000000F0  4A CD 2A BD 88 22 BE 47  11 29 01 C9 57 73 D6 25   J.*.."..G .)..Ws.%
00000100  F0 59 04 53 D8 17 D8 84  72 6B 86 BD DA 20 58 FA   .Y.S.... rk... X.
00000110  CD DB 5C 34 EC 4C 13 1C  F8 DF 97 84 19 F5 52 BB   ..\4.L.. ......R.
```

```
00000120  89 C9 EF 5D 11 65 E2 A4  DA DB 15 8B 48 23 AF 9E   ...].e.. ....H#..
00000130  E6 55 0E D6 10 F5 67 9D  37 64 73 F7 E2 C4 A5 EB   .U....g. 7ds.....
00000140  52 A5 44 30 99 E4 AC 7E  45 92 D5 1E 98 12 1A 4F   R.D0...~ E......O
00000150  4F 2C 87 16 E0 EE 8F 00  F1 95 00 FC F3 F6 CF 7D   O,...... .......}
00000160  22 B5 D4 2C 2D 70 F5 D6  93 33 9C BE 46 64 2F 73   "..,-p.. .3..Fd/s
00000170  4B D6 B8 A2 30 CE 6F 43  33 B0 2D 45 0F 0C 39 40   K...0.oC 3.-E..9@
00000180  DE AA 35 2E 3F 13 F5 98  07 7D 30 9F A1 7C 16 73   ..5.?... .}0..|.s
00000190  95 11 80 C6 E0 23 F6 DF  9B 0F C0 E7 E8 8A CF E9   .....#.. ........
000001A0  5D 29 98 A8 32 83 9E 91  23 3A 99 F0 94 33 E8 E4   ])..2... #:...3..
000001B0  04 F6 73 8C C8 76 76 7F  22 BE 84 6E E7 F4 AE 13   ..s..vv "..n....
000001C0  2A CA 77 29 F3 09 23 0A  AE 14 AC 37 CA 2B 0C CA   *.w)..#. ...7.+..
000001D0  04 62 52 CC 9B A1 61 AB  F8 5F 05 4B D7 CA 10 20   .bR...a. ._.K...
000001E0  F4 04 A0 79 D1 CA 6E 69  EE 07 5F 03 A9 3C 48 11   ...y..ni .._..<H.
000001F0  94 C7 95 15 18 00 48 DF  9E 39 CD D8 A7 0B 04 41   ......H. .9.....A
00000200  4B A8 57 B4 F7 D2 43 FF  6E 16 7E EA 47 C2 90 CA   K.W...C. n.~.G...
00000210  7C 6A B6 CF AD 5F 5A 98  17 F8 68 2E 57 FB F9 70   |j..._Z. ..h.W..p
00000220  C9 17 56 1E D3 99 A2 14  59 57 E8 83 62 E7 C5 DB   ..V..... YW..b...
00000230  21 F8 1E 84 F3 69 E1 1A  F1 80 69 66 BF D8 65 A3   !....i.. ..if..e.
00000240  49 AC F9 FD 03 AA F1 0A  3C 03 17 03 01 00 20 C1   I....... <..... .
00000250  EC 33 E7 70 98 30 2B B3  EB E5 B9 CD 50 A2 93 77   .3.p.0+. ....P..w
00000260  05 73 9B E6 F2 9E 56 25  FB 6E E4 9F C4 11 8B 17   .s....V% .n......
00000270  03 01 02 20 BD 9F 00 ED  4C 44 2C D4 FA 62 00 D9   ... .... LD,..b..
00000280  EF 0F 01 DE 09 7A E7 31  AE 74 F9 F8 9C 51 1A 1F   .....z.1 .t...Q..
00000290  BE FB D6 BD 13 B1 E8 6B  7F 41 5E 70 33 F9 3E D6   .......k A^p3.>.
000002A0  D4 88 FE D2 6B D6 19 3D  0E 45 E5 D8 1C 9F 9D 95   ....k..= .E......
000002B0  D9 82 6A 2E 6C E1 05 B5  18 75 14 57 1A F0 40 5A   ..j.l... .u.W..@Z
000002C0  06 5A 67 48 9D 5B D4 4C  81 22 93 24 53 F3 F4 17   .ZgH.[.L ."..$S...
000002D0  BA 92 71 45 35 A6 F4 E3  5A 35 25 A4 72 50 44 12   ..qE5... Z5%.rPD.
000002E0  40 98 E6 7D A2 36 FD 82  19 B6 EF 73 BB 17 DC 8F   @..}.6.. ...s....
000002F0  3D 71 FF A8 64 AA 64 E2  60 C0 55 DB 1A 46 69 9A   =q..d.d. `.U..Fi.
00000300  9B 76 6B 9C 3E 2E ED 5F  7B BE 5A F3 BF 34 51 B9   .vk.>._ {.Z..4Q.
00000310  BD E1 ED A9 B0 25 FE 8C  43 D8 24 22 20 EE 39 E8   .....%.. C.$" .9.
00000320  38 D8 0A 50 E4 01 35 B5  AE A7 5A C5 6C 96 ED 09   8..P..5. ..Z.l...
00000330  07 65 AD D3 49 18 2F 04  60 EA ED 91 9A 13 8D 55   .e..I./. `......U
00000340  71 71 09 C5 44 C9 28 96  A4 E4 94 D3 4E DC 4C 14   qq..D.(. ....N.L.
00000350  8D B4 9C F8 95 78 23 4B  40 3F 09 5F 5E 76 35 09   .....x#K @?._^v5.
00000360  21 95 DD 85 6F 53 E1 30  5E 05 60 9A E1 62 5F D7   !...oS.0 ^.`..b_.
00000370  11 9B FF 1A 22 D1 62 44  AD A3 5B 8D 4A 3D D4 A1   ....".bD ..[.J=..
00000380  98 CB 02 5B DE 58 D5 88  82 C8 F8 EF F3 7C 93 CA   ...[.X.. .....|..
00000390  67 16 E9 66 B7 BC A3 5D  DB C9 14 AE 7D 08 F3 2F   g..f...] ....}../
000003A0  06 F3 64 97 B8 04 C8 13  7D BB 34 0F 2E 85 B3 9C   ..d..... }.4.....
000003B0  96 3A 35 D3 42 26 7C 7D  26 04 26 B5 FF 75 32 6A   .:5.B&|} &.&..u2j
000003C0  43 50 7C D5 5D 7A 2D 1A  1D 48 55 CA 05 82 32 6A   CP|.]z-. .HU...2j
000003D0  9E 8C 36 51 71 36 AD 2E  15 80 64 62 48 59 EE BA   ..6Qq6.. ..dbHY..
000003E0  D1 5D 46 E1 29 48 55 12  B5 FC 5E EE 38 F1 02 12   .]F.)HU. ..^.8...
000003F0  F7 44 A7 D4 A0 24 4D E6  2F 43 14 68 29 E7 3F C7   .D...$M. /C.h).?.
00000400  A4 A2 82 EA D5 70 67 09  0A D1 B8 EF 6E 80 2F 40   .....pg. ....n./@
00000410  83 CB 7A AA C8 19 57 8E  3A E6 2A 64 5E 5B E3 14   ..z...W. :.*d^[..
00000420  C3 77 0E 50 5D 53 CC 00  52 3C 0B 65 BC 40 4D AF   .w.P]S.. R<.e.@M.
00000430  87 17 22 E2 8B 7A E0 51  8B A5 40 CC A2 86 12 26   .."..z.Q ..@....&
00000440  F0 01 F0 41 D5 A5 A7 7F  89 C8 F7 A8 9C C9 2F DC   ...A.... ....../.
00000450  AF 80 30 D6 6C 63 27 AB  5D CA 74 E3 6C 20 8F 52   ..0.lc'. ].t.l .R
00000460  38 73 93 E2 36 AA 06 83  34 BD 07 F4 78 F2 95 6A   8s..6... 4...x..j
00000470  27 04 8A FF 3B F7 B0 C2  EA A2 20 BE CD 54 4E 36   '...;... .. ..TN6
```

```
00000480  B9 B8 DA 42 33 1F CE 04  F3 02 7E CD CA 26 25 73   ...B3... ..~..&%s
00000490  6A 99 F3 6B                                       j..k

00000000  17 03 01 00 20 B2 20 21  7D 60 12 F3 9E 25 40 CF   .... . ! }`...%@.
00000010  BE F9 7D AD B7 E0 48 03  11 5E 7E A0 7A 39 1B 47   ..}...H. .^~.z9.G
00000020  9A 3D D9 13 47 17 03 01  02 20 2C 17 B9 B5 80 FC   .=..G... . ,.....
00000030  98 9B F7 B5 58 AB 31 DB  23 99 E6 84 80 B5 BF 44   ....X.1. #......D
00000040  5C 2A 9A 79 A7 1F 43 7A  E5 4F 45 46 53 C1 34 29   \*.y..Cz .OEFS.4)
00000050  E2 2A 08 A1 C0 26 91 40  C5 0C 56 60 1D 7A 43 73   .*...&.@ ..V`.zCs
00000060  D2 EA 72 FD DC 1C 5A 66  00 4D D1 9B 1D 0C 95 79   ..r...Zf .M.....y
00000070  20 C9 6A 6D 09 D2 C8 79  99 FD 02 3E B1 0E 7A B6    .jm...y ...>..z.
00000080  86 4A DB B9 1F 17 5E BE  28 A0 17 5C CD C9 3D 3C   .J....^. (..\..=<
00000090  AF 23 C5 A6 8E D2 D4 BB  F1 89 B7 D2 5A 1C 89 CE   .#...... ....Z...
000000A0  CB 59 BA F3 2A 60 52 F2  FB 93 8E BA 28 DB D6 27   .Y..*`R. ....(..'
000000B0  C3 AC 42 48 01 76 9B DE  D1 BC 67 A6 3A 52 E6 90   ..BH.v.. ..g.:R..
000000C0  17 E1 56 9D 7B 7B 54 01  4E C0 17 44 CA 58 08 D2   ..V.{{T. N..D.X..
000000D0  43 61 8C 87 24 44 64 EF  FE 23 77 9F DA 3D 1B 83   Ca..$Dd. .#w..=..
000000E0  80 C9 9D 28 13 81 53 D7  8E 52 FB C2 B5 94 7A 8D   ...(..S. .R....z.
000000F0  33 E4 05 DB 15 C1 FC C1  FA 8B 09 1C 7F DA C4 4C   3....... ......L
00000100  5E BC 42 2D F6 7B 4D CA  98 B2 56 31 95 3D 18 68   ^.B-.{M. ..V1.=.h
00000110  E1 4A B9 D4 EE B6 21 A6  DE 13 E0 EF 13 C4 9F C6   .J....!. ........
00000120  BE 91 D4 45 0A C2 AC 8C  E6 61 4E AF 98 8B 82 CE   ...E.... .aN.....
00000130  F0 3C 9D 87 6E FA 64 C1  71 5A 08 59 5C 83 1B 5C   .<..n.d. qZ.Y\..\
00000140  58 54 77 DC 37 63 A2 F0  73 00 CD A1 2F 24 03 CB   XTw.7c.. s.../$..
00000150  FC C8 B0 A0 05 91 EA 40  B4 EF 75 6E 90 39 20 7C   .......@ ..un.9 |
00000160  8C 3D 8B DC 81 79 17 FB  8B 76 DC A2 5E DE B5 A5   .=...y.. .v..^...
00000170  2F E3 E5 51 8A 13 7E A7  BD 31 6E 4C A8 B6 D8 2C   /..Q..~. .1nL...,
00000180  E0 13 F1 95 F4 F7 05 BD  49 02 28 1A 6E 21 C7 CE   ........ I.(.n!..
00000190  2F A3 0D 7A 2D 0E 4C 53  36 73 11 A1 9B 9A CC C4   /..z-.LS 6s......
000001A0  EB 1D D1 4C 1E ED 53 C5  75 70 8C E8 1F 54 9E 0A   ...L..S. up...T..
000001B0  B8 CF 91 F9 00 F8 C1 CD  C0 98 37 AE 2B 2D 04 66   ........ ..7.+-.f
000001C0  29 86 89 F3 71 C3 9D ED  1E 1A 56 23 49 03 48 8C   )...q... ..V#I.H.
000001D0  E7 62 C1 B3 31 19 7C 6F  4F 50 C9 9E 69 6F 35 F5   .b..1.|o OP..io5.
000001E0  DF B5 75 1E BE 80 AF 5E  25 4A FB 1F 64 7E 84 B8   ..u....^ %J..d~..
000001F0  82 36 C5 18 6B 6A 48 BB  6D B0 07 20 62 F8 1E F1   .6..kjH. m.. b...
00000200  86 52 3C 56 8E AF 74 F1  A4 88 45 3B 86 7F 79 43   .R<V..t. ..E;..yC
00000210  11 03 32 59 D6 DE C7 72  C6 FA 21 87 0E 68 1C DD   ..2Y...r ..!..h..
00000220  5F B5 F9 C2 1E CD 6A 1C  37 87 39 F8 2A D3 3C 09   _.....j. 7.9.*.<.
00000230  CB 88 F0 C6 6B 82 9E 57  FC 01 37 25 BE 0E 26 42   ....k..W ..7%..&B
00000240  4C 4B 9B 4A 59 23 6A D9  3D 31                    LK.JY#j. =1

00000000  17 03 01 00 20 B8 1A DD  E8 75 E3 E5 F3 B2 D6 BA   .... ... .u......
00000010  3A 02 73 CE B8 4A 21 92  F4 4B D2 35 A8 7B 46 D2   :.s..J!. .K.5.{F.
00000020  1D C7 BE 1C AD 17 03 01  02 20 B2 A6 98 27 FD 78   ........ . ...'.x
00000030  9E B4 95 A0 79 C7 1C 82  BD 94 5B 62 37 8A CC C1   ....y... ..[b7...
00000040  67 E4 62 33 C9 44 94 88  01 05 58 D7 17 8A F5 2B   g.b3.D.. ..X....+
00000050  95 58 DD 90 59 21 C5 17  D2 A6 0D 56 F5 06 D9 A2   .X..Y!.. ...V....
00000060  3A 68 FA 61 86 08 B2 49  76 93 11 53 A5 91 BA 50   :h.a...I v..S...P
00000070  B2 28 0A 53 0D 53 43 6F  08 4C 31 D5 5D 77 5A 0E   .(.S.SCo .L1.]wZ.
00000080  0A 25 02 61 59 AA BE 51  BF 75 CF 1C 5D 5B 51 64   .%.aY..Q .u..][Qd
00000090  D3 A1 F5 12 C0 D5 2C AA  86 BF E9 BC F4 03 3C 6A   ......,. ......<j
000000A0  34 CE 98 E9 9C 3D E5 57  01 61 14 BA D1 09 C4 65   4....=.W .a.....e
000000B0  C7 ED E1 9A 1B 8E 27 D4  EA CA B7 EC 58 F2 3D 67   ......'. ....X.=g
000000C0  89 A3 DC FF ED 35 7D DB  72 79 81 23 87 B3 61 82   .....5}. ry.#..a.
```

```
000000D0  80 93 85 A0 9E 68 3D A5  49 0E 3B 7C 36 1F A9 8A   .....h=. I.;|6...
000000E0  2D 54 49 A9 21 68 CD F8  C6 36 9F 46 CB 71 97 DF   -TI.!h.. .6.F.q..
000000F0  CD B1 47 F4 5B 91 25 8E  AC 52 B4 89 71 A7 75 CD   ..G.[.%. .R..q.u.
00000100  4D 82 2A E2 D5 1E 94 49  56 1D 7F E9 F8 02 6D 6F   M.*....I V....mo
00000110  DF 82 58 D6 20 32 F5 D8  4A 79 40 B8 8E CA A0 6D   ..X. 2.. Jy@....m
00000120  74 4A 2E 82 46 30 F0 14  71 AE 2E 89 6B 85 4C 74   tJ..F0.. q...k.Lt
00000130  31 59 6D E3 1B D5 34 D8  07 B1 3C 92 E2 FD B7 6E   1Ym...4. ..<....n
00000140  9B 75 2A 1A 8C 50 3C 49  38 C4 A3 21 8D A6 C2 03   .u*..P<I 8..!....
00000150  C9 3A 63 3D C6 5E 0D A4  55 5A 8D A7 D2 11 06 9D   .:c=.^.. UZ......
00000160  12 D5 97 1E E3 95 11 86  21 3C DD 20 CE AE 0C F5   ........ !<. ....
00000170  99 F8 B4 AB 6D 05 C7 50  AD D2 BF 10 B1 89 C5 73   ....m..P .......s
00000180  B9 AA A7 14 D6 56 01 F8  34 57 67 3D BE 3D 95 EF   .....V.. 4Wg=.=..
00000190  92 F2 5F 34 83 14 40 C9  52 77 40 8C A2 64 FA FF   .._4..@. Rw@..d..
000001A0  F9 15 97 4D 4B C6 61 58  7D 70 A7 6E B5 96 E1 69   ...MK.aX }p.n...i
000001B0  49 C4 F1 16 C3 77 0C 0B  00 85 E8 17 AA 90 4F B4   I....w.. ......O.
000001C0  B5 32 34 A1 80 17 32 35  31 2B 06 CC 90 6E E4 E1   .24...25 1+...n..
000001D0  75 ED FA FD 68 29 A5 BA  A4 7A ED FA 8D E7 C6 A6   u...h).. .z......
000001E0  81 5E 69 33 C2 3D 3B B5  33 FB 69 FF 23 E8 C6 0B   .^i3.=;. 3.i.#...
000001F0  EB 84 5F 28 B7 75 1D 61  23 1D 60 8F 2D 29 73 9A   .._(.u.a #.`.-)s.
00000200  2F 78 9F DE D9 BD DF F3  5E FA C4 52 A8 FF 47 EC   /x...... ^..R..G.
00000210  21 D7 EA 61 0C C9 29 4E  58 06 04 EC 4F 52 82 6E   !..a..)N X...OR.n
00000220  4E B6 F1 48 76 B1 06 4A  E8 F8 6B F6 30 45 D3 53   N..Hv..J ..k.0E.S
00000230  79 BA 41 5A 93 5E 8F 93  56 C5 52 9B 82 70 C7 BC   y.AZ.^.. V.R..p..
00000240  EC C1 B6 29 FA CB CD 64  D0 32                     ...)...d .2

00000000  17 03 01 00 20 C4 5F D5  9F F5 89 8D C3 A7 89 AF   .... ._. ........
00000010  AD 93 65 D4 1E 99 9C 7D  71 CB 70 58 89 5E 65 EC   ..e....} q.pX.^e.
00000020  8D 6A BF 8B BA 17 03 01  02 20 A4 10 78 E7 08 0C   .j...... . ..x...
00000030  1C 30 94 A3 BB 6D C7 F4  89 8F 2C 87 F4 46 72 1D   .0...m.. ..,..Fr.
00000040  D5 D8 ED 78 B9 50 34 AD  73 79 7B E2 90 7E 0C D9   ...x.P4. sy{..~..
00000050  99 8B 70 47 B7 E8 9A AF  BE 6E 49 BE 51 31 DC 4C   ..pG.... .nI.Q1.L
00000060  E1 97 28 40 BA 20 2F 95  50 31 16 A8 09 D4 AF 78   ..(@. /. P1.....x
00000070  4A FA 12 86 85 BA 73 BD  BC 9F E2 53 4F DB DB E5   J.....s. ...SO...
00000080  5F 0D 4C 45 F8 E5 D0 72  61 D2 FC A7 1A 82 B6 58   _.LE...r a......X
00000090  2A FF 5D D8 56 EC FC D6  56 29 A7 B3 8E E5 F5 52   *.].V... V).....R
000000A0  CA C2 72 6A 20 93 17 C8  DB 1A 5F 37 B7 BA C9 41   ..rj ... .._7...A
000000B0  8F E1 07 17 2B 3D 9F BA  25 3D C3 63 BA 8A 8F 12   ....+=.. %=.c....
000000C0  F5 73 EF E8 99 E4 FB 9A  B5 4D 28 B9 AD E0 D9 8E   .s..... .M(.....
000000D0  F0 AF 2D 81 BB BE C3 F2  82 9A 20 69 59 99 CD 1E   ..-..... .. iY...
000000E0  9D FC CA 41 CB 9E E8 49  FC 3F 91 CB 7B 37 4B 77   ...A...I .?..{7Kw
000000F0  42 04 6C 7E 99 4A D1 82  88 18 12 7B 28 4A C9 F2   B.l~.J.. ...{(J..
00000100  C3 C6 C3 93 E0 8F 12 A2  5E BD 7D 6C 48 97 3B BB   ........ ^.}lH.;.
00000110  C0 7C C8 BC 9F 0F 22 86  96 86 BF 2F CF DB 1A 13   .|....". .../....
00000120  82 D9 FD 9C A4 90 2E 8D  95 FE 35 F7 F3 B8 2C E7   ........ ..5...,.
00000130  52 F2 29 B9 E3 20 E1 BB  17 90 AE EE 36 00 DD EF   R.).. .. ....6...
00000140  ED 57 1E 79 BC 26 76 70  20 08 F6 BD AA 0E C4 43   .W.y.&vp  ......C
00000150  DD E3 91 71 B2 23 E8 05  75 C6 A6 B5 01 C6 9F 29   ...q.#.. u......)
00000160  B3 24 9C 57 85 98 24 79  90 2E E8 7B E0 8B E6 03   .$.W..$y ...{....
00000170  37 2A C2 C6 23 DD A0 E9  A1 1A F5 CF 80 06 8B E0   7*..#... ........
00000180  86 65 03 77 E4 AA 72 DC  EF 6E B9 C6 56 D1 98 05   .e.w..r. .n..V...
00000190  D9 B3 3C 8B 13 79 D7 15  E7 C9 99 FA 45 5E 69 D9   ..<..y.. ....E^i.
000001A0  F0 30 45 37 5E 17 26 13  DC 08 95 10 D5 4B 77 C7   .0E7^.&. .....Kw.
000001B0  D4 74 E9 0B E3 89 E4 52  D4 06 D8 9D BC F1 FA 5F   .t.....R ......._
000001C0  E8 A5 1D 6A 73 7F 78 CC  FF CB 9E 78 EE 37 62 78   ...jsx. ...x.7bx
```

```
000001D0  D4 0E B2 11 D0 CA 0D B1  71 D7 72 C3 07 51 0E DF   ........ q.r..Q..
000001E0  E3 86 F5 0E 60 E5 F2 79  9C 30 5A C4 D9 29 A1 D6   ....`..y .0Z..)..
000001F0  B1 2B 9A 8B E2 0F 3F B3  B2 2A 45 D8 A2 1F 2C 36   .+....?. .*E...,6
00000200  E4 9B 85 22 64 8C 5E 58  7E D2 CA 68 73 09 5E 43   ..."d.^X ~..hs.^C
00000210  95 04 33 50 AB B4 74 B4  DC 29 65 1D 8B 4F 7A F8   ..3P..t. .)e..Oz.
00000220  08 47 53 BF 85 61 3C 08  99 05 52 ED 47 04 A0 10   .GS..a<. ..R.G...
00000230  B4 9D 8A B8 D0 3E 2A 10  67 AA CE A7 23 02 C9 E9   .....>*. g...#...
00000240  93 B6 8A 53 C0 00 75 06  0C D4 17 03 01 00 20 52   ...S..u. ...... R
00000250  87 3F 1C 72 AE A6 21 A4  52 31 39 02 E3 9A 26 BE   .?.r..!. R19...&.
00000260  1D 07 16 88 3D 2F 2B 52  EF E7 02 75 F4 8F A0 17   ....=/+R ...u....
00000270  03 01 02 20 B2 5A 41 2B  8C 39 D9 2A D4 D2 4F B8   ... .ZA+ .9.*..O.
00000280  D1 E8 B4 E2 A8 06 19 BE  03 C9 94 31 7F FA A9 E6   ........ ...1...
00000290  76 44 02 19 88 96 61 4C  27 33 F3 0E ED 3A 91 51   vD....aL '3...:.Q
000002A0  3D 7A F9 93 0B C9 5D 05  97 FB E4 24 47 CE 09 73   =z....]. ...$G..s
000002B0  21 00 15 B7 62 73 E7 CB  15 96 26 A7 41 D9 3C 34   !...bs.. ..&.A.<4
000002C0  94 4D F2 00 10 57 0A 68  13 D6 73 DE 4A BF 11 B8   .M...W.h ..s.J...
000002D0  DA 33 64 77 0F 4E E2 1C  F0 74 D2 89 4E C4 DC 5D   .3dw.N.. .t..N..]
000002E0  54 32 B9 B2 3D 9B 32 BC  4E 56 BF 86 D4 92 6F 2F   T2..=.2. NV....o/
000002F0  63 87 C3 87 5A F6 99 C0  C0 B5 92 1D E7 81 34 74   c...Z... ......4t
00000300  74 C9 51 08 4A DA A1 B2  BF B1 85 F7 2D BE F3 E2   t.Q.J... ....-...
00000310  61 1F 3B 8A 9B 3E 73 C2  A5 DE FD A6 48 04 CA 27   a.;..>s. ....H..'
00000320  2C 45 6E 49 BD FB 8B 4B  EC 70 C1 86 A2 CE DF 47   ,EnI...K .p.....G
00000330  75 70 F5 95 7F 38 F3 A7  DC 84 AB 55 67 24 7C 71   up...8.. ...Ug$|q
00000340  25 CD 93 EE E0 C7 07 61  99 DA AD 3A 8F 3D BD C4   %......a ...:.=..
00000350  F2 55 A6 47 5C 81 9F 09  06 10 D8 4E A2 2C 00 B1   .U.G\... ...N.,..
00000360  70 92 38 7E BA C2 61 42  B4 1F 7D 1C 47 33 5E EE   p.8~..aB ..}.G3^.
00000370  1E 5F 78 F3 88 23 14 F4  57 22 8A 36 95 95 A5 6F   ._x..#.. W".6...o
00000380  E6 8B A7 BD BD 31 20 CA  94 9C BF 34 D4 C6 31 7F   .....1 . ...4..1
00000390  29 58 AD 67 09 F8 FF 73  D4 34 0C 8B 43 73 F6 DB   )X.g...s .4..Cs..
000003A0  60 EF 28 BB 39 0B BD 12  92 91 06 CC 7E 82 D1 E9   `.(.9... ....~...
000003B0  04 0A 5F 5E CB 44 D7 BD  AF 09 AF 58 EE 8E 80 53   .._^.D.. ...X...S
000003C0  F9 83 26 20 E5 49 50 F9  C0 C0 DE 86 FA 64 A6 FF   ..& .IP. .....d..
000003D0  A8 AE A0 DE 57 CE 25 76  EB 0E 4A EB BD EB 4A 1A   ....W.%v ..J...J.
000003E0  CE 4F 98 08 E6 46 F3 D7  78 FC 21 E4 68 89 3A 1E   .O...F.. x.!.h.:.
000003F0  FC E8 A1 0A C9 76 6C 00  BC 5E 1C C3 5A 21 26 A2   .....vl. .^..Z!&.
00000400  7B 44 A3 45 41 99 4B 05  94 67 35 4E 26 26 DC 66   {D.EA.K. .g5N&&.f
00000410  CB 48 B7 8E 4B 8A F9 29  47 B0 A7 4D 1F FB 1B 53   .H..K..) G..M...S
00000420  5B 05 25 DB EE 28 83 56  F5 DD 70 3F 1A 19 31 A1   [.%..(.V ..p?..1.
00000430  C7 3B A4 C4 C1 15 42 90  7E 00 34 F0 68 4E 15 B3   .;....B. ~.4.hN..
00000440  BA 5F 15 6F 70 65 41 D7  1E 66 25 CA E6 E9 64 47   ._.opeA. .f%...dG
00000450  4D D1 A3 FB 33 8E 3D 59  B2 5A 9E 4B 00 77 0F D9   M...3.=Y .Z.K.w..
00000460  A2 23 6E E4 53 28 88 06  E0 2A A2 B2 AD B9 F1 CD   .#n.S(.. .*......
00000470  E5 75 A8 A3 07 7F 45 DF  A7 F7 24 92 B3 7A B0 5A   .u...E. ..$..z.Z
00000480  88 6E BD CB 2F F1 F9 D5  65 57 CB 4D 20 7D 8A D4   .n../... eW.M }..
00000490  6B B4 2C 72                                       k.,r

00000000  17 03 01 00 20 7C 13 1C  EB E9 FE B2 FE 5E 98 64   .... |.. .....^.d
00000010  5D 2A 13 48 27 80 9A FA  28 24 28 E6 A7 5D 7C 07   ]*.H'... ($(..]|.
00000020  2E AE 8B DB 6C 17 03 01  02 20 4D 4B 1B EA FD 8F   ....l... . MK....
00000030  9E 65 38 F4 2A AB FB 54  E2 11 8B B4 80 8D 6D 57   .e8.*..T ......mW
00000040  9A BF 93 84 21 1A 89 C4  EC 77 68 9F 4E 4F 61 10   ....!... .wh.NOa.
00000050  CB 5C 01 7C 7E 35 06 28  8D 8D 84 22 2B D5 06 C8   .\.|~5.( ..."+...
00000060  92 D5 58 A9 FF 50 FD B1  B6 09 B3 FC EF 65 45 2B   ..X..P.. .....eE+
00000070  F0 45 BF 1E 64 DD F8 3C  1A BE D1 6B 44 82 5C 98   .E..d..< ...kD.\.
```

```
00000080  57 31 97 19 F7 CD 40 D1  3F CB B3 C2 F9 6E A7 9C   W1....@. ?....n..
00000090  BF 3E 3E 97 FC CD E9 F1  42 A0 A0 D5 45 BE 59 B9   .>>..... B...E.Y.
000000A0  F2 42 44 E8 9E 56 9C C6  0F B0 6D 2F 3B 3E AE CF   .BD..V.. ..m/;>..
000000B0  73 47 42 D7 9C 5F E5 73  A3 68 1C 45 A3 32 DD 05   sGB.._.s .h.E.2..
000000C0  67 31 02 CA 01 E7 48 52  F1 4A ED 86 99 86 1A 82   g1....HR .J......
000000D0  36 2D 47 83 58 1A E7 54  CE 44 BE D7 BE 68 35 E7   6-G.X..T .D...h5.
000000E0  61 51 4E E8 6C 54 A9 7D  BC 12 40 19 51 BA C3 B1   aQN.lT.} ..@.Q...
000000F0  81 4A 2F D7 93 B2 69 CC  27 97 11 BD 98 FA 8F 54   .J/...i. '......T
00000100  A7 14 21 9D B0 C5 DD F5  2E E7 8C 15 73 E0 3C 3C   ..!..... ....s.<<
00000110  5F 25 AA 3D 42 24 4A AD  07 9F F9 94 EB C9 85 1C   _%.=B$J. ........
00000120  1F 16 C2 8E DE 13 A0 4D  16 DD 24 74 32 ED D9 40   .......M ..$t2..@
00000130  AF FC 7A 3E 3B F6 5B C4  F5 79 FF A4 52 F6 4C 65   ..z>;.[. .y..R.Le
00000140  63 0C 71 E6 CA 3D E1 2C  E1 37 9C C7 49 EB 4A 43   c.q..=., .7..I.JC
00000150  A2 75 57 8B 57 3D 29 87  AD C9 02 E8 8A F0 C0 F7   .uW.W=). ........
00000160  21 1B DA 04 8E 6A 51 35  51 9E 77 91 BF 76 37 E7   !....jQ5 Q.w..v7.
00000170  62 91 C4 1C 54 4F D9 C5  F5 87 59 CA CD AA 48 E4   b...TO.. ..Y...H.
00000180  D2 1B 22 00 49 BC 1C E1  8F 6B 75 FE 26 46 B9 56   ..".I... .ku.&F.V
00000190  14 21 BB A2 B9 69 82 70  E9 65 00 B2 93 BE 62 A6   .!...i.p .e....b.
000001A0  D7 37 D2 80 AE 2C 54 7E  B9 E6 31 B0 71 20 DA 20   .7...,T~ ..1.q .
000001B0  2E 7A 31 D0 7B F2 B7 21  77 A5 63 EB F4 5B 43 3D   .z1.{..! w.c..[C=
000001C0  A6 A7 23 9A E7 E9 C4 51  03 F5 96 48 D6 EB 6F C6   ..#....Q ...H..o.
000001D0  C3 68 F9 F1 C0 F4 4B 58  5D 48 4F 51 3E 81 61 89   .h....KX ]HOQ>.a.
000001E0  3E 18 A5 F6 0B CD C6 28  70 AB D1 4F 52 91 66 8F   >......( p..OR.f.
000001F0  36 73 BF A8 0E 5B 55 D1  D6 B5 F0 36 7B A7 12 45   6s...[U. ...6{..E
00000200  A6 81 D3 D6 A4 4F 2F 43  25 89 04 25 29 7A 36 5E   .....O/C %..%)z6^
00000210  D2 70 93 22 5A 46 4F 24  EE E0 02 06 BE 8F 50 B4   .p."ZFO$ ......P.
00000220  6E 56 36 61 98 97 7D 0C  3E 02 6E 6F B4 A6 67 1F   nV6a..}. >.no..g.
00000230  73 97 3C D4 46 1C 2E 68  43 94 B5 E7 DA 81 18 E7   s.<.F..h C.......
00000240  54 20 A5 61 00 D0 22 D0  8B 2F                     T .a..". ./

00000000  17 03 01 00 20 D1 AA 01  80 A2 84 FF 4F 7D CD B4   .... ... ....O}..
00000010  63 1F 13 95 AD C4 FB A2  B6 62 E7 B4 B4 6D D9 5E   c........ .b...m.^
00000020  4A 6D 66 9C 94 17 03 01  02 20 FF B8 DA F8 05 44   Jmf..... . .....D
00000030  DD 65 D2 85 6A 94 C8 89  54 1A FF 93 D3 91 1F EE   .e..j... T.......
00000040  27 64 23 B7 6E 05 39 DF  F8 47 E9 12 11 BE 8A 96   'd#.n.9. .G......
00000050  24 38 57 40 BA ED E7 BE  D5 D6 A8 3B 32 00 6D 49   $8W@.... ...;2.mI
00000060  52 B7 B4 86 47 85 F6 7F  D6 C5 08 25 C6 67 06 C4   R...G.. ...%.g..
00000070  FA DF 62 B7 34 43 0D 68  7D 9A 31 87 40 67 F1 DC   ..b.4C.h }.1.@g..
00000080  B6 3E 95 32 A3 99 58 D8  83 EB CD FF 41 91 3C 88   .>.2..X. ....A.<.
00000090  00 37 87 33 A2 10 0C ED  33 FD 95 54 75 BB 8A 07   .7.3.... 3..Tu...
000000A0  61 82 68 EA 18 4B 17 B9  7F 6E 99 61 1E 4D CB 84   a.h..K.. n.a.M..
000000B0  4C CF B0 93 BB 1D FF 7D  FD CB 49 AF 55 85 7F 12   L......} ..I.U..
000000C0  35 1D 41 CC 16 D1 7A F9  E2 72 7D 89 04 C8 1D F9   5.A...z. .r}.....
000000D0  A5 E9 55 9F D0 A2 AB AF  4C 23 B6 78 9C 2F D1 11   ..U..... L#.x./..
000000E0  C4 DE 2D 16 B4 51 28 1F  2C 45 E6 11 EC 9D 87 B1   ..-..Q(. ,E......
000000F0  A9 F1 CC 6A 89 83 CF 65  5E D8 66 5E E5 CC 54 0D   ...j...e ^.f^..T.
00000100  45 76 77 75 A5 62 38 6A  78 87 95 6E 4B 92 4F A1   Evwu.b8j x..nK.O.
00000110  78 E2 62 26 88 C0 FB E7  37 0B 56 BF 7C 73 40 FA   x.b&.... 7.V.|s@.
00000120  C5 CE 15 58 3E F4 4F F8  63 6A 91 A2 CD 71 FA C7   ...X>.O. cj...q..
00000130  FD E4 E2 1D BB CD BF 29  47 FE F8 6D AB 75 11 77   .......) G..m.u.w
00000140  EC 78 96 D7 AC F9 4E 15  B2 E0 AB 75 39 FC 19 B6   .x....N. ...u9...
00000150  40 FB 80 AA C4 67 A5 24  3F 1B A9 5C 43 9D C4 02   @....g.$ ?..\C...
00000160  B2 1F EF 8A 7B 61 B3 DB  03 78 AA 9C 71 C6 7B DB   ....{a.. .x..q.{.
00000170  0C 3F B3 D1 79 BF 99 C2  D5 81 91 2D 81 41 6F 59   .?..y... ...-.AoY
```

```
00000180  7A 87 1F 63 61 B7 51 E1  2F 83 D4 88 04 BA 07 B5   z..ca.Q. /.......
00000190  58 13 88 B2 4E 64 A9 CB  0F 87 B8 2D 7E AE 6B E2   X...Nd.. ...-~.k.
000001A0  B0 F4 89 42 75 79 88 3E  BE 91 8C E3 DE 1E 7F E2   ...Buy.> .......
000001B0  AD 8F 60 99 88 03 F2 00  DF F4 EF 2D 60 C4 8C 02   ..`..... ...-`...
000001C0  89 F4 5B 81 66 69 98 5D  22 4E 44 68 FA 3B 83 38   ..[.fi.] "NDh.;.8
000001D0  2D A4 31 AC 22 37 14 1A  76 10 46 69 38 04 35 C4   -.1."7.. v.Fi8.5.
000001E0  3B A5 C2 2F FF 25 4E A4  7D EA EE CB 75 D1 15 B8   ;../.%N. }...u...
000001F0  31 F0 BC 82 75 3F C9 39  B9 50 B9 71 48 7C BF C1   1...u?.9 .P.qH|..
00000200  3B A5 91 31 33 FC F5 B1  49 7C 75 E8 27 78 8D 15   ;..13... I|u.'x..
00000210  BC FA ED A1 BC 16 EC 07  F1 57 0B 2C 94 CC 1C 87   ........ .W.,....
00000220  4E 91 A4 CB F4 CA BA 9E  43 C7 F1 D2 A4 33 A4 7E   N....... C....3.~
00000230  23 CB 02 64 89 42 10 FF  4A FC BD A9 29 3A B9 D7   #..d.B.. J...):..
00000240  D3 FA BB 57 60 EF D4 3C  62 9B 17 03 01 00 20 1D   ...W`..< b..... .
00000250  46 A3 9A CF 01 51 54 66  C2 B7 10 9E 21 FD 53 1A   F....QTf ....!.S.
00000260  EB 68 AC 53 F7 B4 6D 5F  7E 38 A3 CA 18 3D 9A 17   .h.S..m_ ~8...=..
00000270  03 01 02 20 D2 FB DD D6  FF E6 B7 C0 49 A5 39 22   ... .... ....I.9"
00000280  74 4F 5C DD B2 7A 35 7F  89 EF 0F 10 BA 51 0C A8   tO\..z5 .....Q..
00000290  0D C1 E8 E7 5F 65 12 16  69 63 D6 5E 7E EC D3 41   ...._e.. ic.^~..A
000002A0  AC 81 DB 4F 15 8E 3C BA  9F E3 19 03 7A 24 C3 45   ...O..<. ....z$.E
000002B0  FD 8F 39 D2 B9 E8 19 0F  36 C6 3F 60 60 91 C0 D3   ..9..... 6.?``...
000002C0  7F 8A 95 72 5D AC 96 D0  25 24 32 26 09 62 49 69   ..r]... %$2&.bIi
000002D0  6E AA B0 6E 74 4B 73 85  A0 10 E1 3F 50 57 78 8D   n..ntKs. ...?PWx.
000002E0  53 E9 FF 98 F8 7D CF 2A  0B 47 7B C0 5C 0A EB 1C   S....}.* .G{.\...
000002F0  07 25 8C 54 E1 E9 2C 1A  0A 63 D2 69 E0 A4 EF 20   .%.T..,. .c.i...
00000300  62 27 75 CF 08 FE 75 C6  35 15 6C 99 87 57 BC D9   b'u...u. 5.l..W..
00000310  B2 6B 40 F9 4D 1E C3 A2  CF 1C 49 F9 46 75 A2 24   .k@.M... ..I.Fu.$
00000320  F6 0D D5 D7 93 88 A6 FF  85 01 03 D0 60 6B D3 EA   ........ ....`k..
00000330  57 93 A4 F9 8E 1E E9 1A  25 50 24 E4 7C 1F 8C AA   W....... %P$.|...
00000340  5F 8C 48 D9 44 82 A3 DE  44 68 41 E9 98 68 1F 02   _.H.D... DhA..h..
00000350  51 FA BD D8 00 90 7B 85  EF D9 6D F0 4C 5B 16 15   Q.....{. ..m.L[..
00000360  00 C9 BB 6D F1 08 00 04  04 A4 75 26 50 AF 9F 96   ...m.... ..u&P...
00000370  B8 F8 75 4D E5 02 FD C5  62 98 B7 A6 B2 99 FA DE   ..uM.... b.......
00000380  A8 27 0B DE B6 E0 D0 4C  51 96 31 32 4F 80 19 F2   .'.....L Q.12O...
00000390  4D 6F 93 27 49 69 6D D2  BE B6 E5 B7 BE 49 38 6A   Mo.'Iim. .....I8j
000003A0  03 9F EB 43 D1 95 CF 49  9C F5 49 D9 ED 54 EC 7F   ...C...I ..I..T.
000003B0  0A E4 EB 90 53 F0 4B C0  85 A4 AB 82 42 2F F6 B4   ....S.K. ....B/..
000003C0  F8 60 EA E0 CC E2 CA AC  DD 49 C4 0C 04 DC 79 66   .`...... .I....yf
000003D0  A9 BE 69 E1 BE FC A1 D0  E9 C0 59 75 14 00 CD 70   ..i..... ..Yu...p
000003E0  8C 83 FD D0 02 54 37 27  0F C8 54 44 12 F9 E1 76   .....T7' ..TD...v
000003F0  4E D3 E6 1A 2E 4A 4C 9F  E6 DB 56 E6 56 84 7F 5A   N....JL. ..V.V.Z
00000400  46 44 3A 1E D8 71 59 22  75 2C 68 44 A8 B2 86 01   FD:..qY" u,hD....
00000410  07 FD D7 69 3C 47 0E 07  0D 79 E1 95 DF 08 AC 1D   ...i<G.. .y......
00000420  48 1E CE 4C 7D 57 7A 95  48 D1 F3 49 8D CC F6 55   H..L}Wz. H..I...U
00000430  6F 76 A0 CC DF 2B 05 6E  BF 7F 2A 23 75 26 F0 69   ov...+.n .*#u&.i
00000440  80 0E 1C 91 4E 92 88 46  99 A1 84 BF A5 EB 35 CC   ....N..F ......5.
00000450  CB EA 2A 72 FB 9A D0 18  1C 58 77 03 CC 64 DC 0A   ..*r.... .Xw..d..
00000460  BA 29 0B 05 6A 8B ED A2  F8 58 B4 53 E0 80 ED 65   .)..j... .X.S...e
00000470  22 C4 92 C1 3A B5 99 91  51 4D 68 C5 33 07 60 2B   "...:... QMh.3.`+
00000480  7C 42 41 AA C4 8C 1D 86  95 93 BD FC 9A 3B 61 87   |BA..... .....;a.
00000490  5E 2F 15 AD                                       ^/..

00000000  17 03 01 00 20 77 FA 6F  82 16 30 77 70 02 D2 3A   .... w.o ..0wp..:
00000010  7F D6 43 34 18 65 3C 45  A9 74 84 7B E2 36 E8 3F   .C4.e<E .t.{.6.?
00000020  DF 76 2C 17 0E 17 03 01  02 20 CF EF E1 CE 12 F0   .v,..... . ......
```

```
00000030  49 30 9C 95 48 D0 E1 B9  65 0A DC 03 78 4B 3F 4B   I0..H... e...xK?K
00000040  62 27 BE 08 44 63 77 8A  4D 91 98 8E 37 B0 CB 6D   b'..Dcw. M...7..m
00000050  A7 EE 15 97 49 7A ED 36  E2 03 4F 39 20 BE D0 14   ....Iz.6 ..O9 ...
00000060  30 A3 B0 20 39 C1 69 57  6C 45 C0 F7 A2 EB 3B D0   0.. 9.iW lE....;.
00000070  8B F4 68 68 02 6C 57 E6  80 1C CA 59 8C 53 E5 2A   ..hh.lW. ...Y.S.*
00000080  C1 3C 54 37 9B 1E 3F 1E  A1 CC 8D B3 5E D3 E8 AE   .<T7..?. ....^...
00000090  7D B2 6A 3E 33 B6 C9 04  8F 27 57 1B 94 BD 69 F6   }.j>3... .'W...i.
000000A0  5F 99 D8 22 58 CB 29 EC  B6 E5 78 3A 8E 3C 03 5F   _.."X.). ..x:.<._
000000B0  A0 83 7F 2B 61 DE E5 4D  ED EE D6 49 E1 28 C4 49   ..+a..M ...I.(.I
000000C0  3C 77 55 38 55 90 35 C0  FB A8 D0 8C 27 14 0B CA   <wU8U.5. ....'...
000000D0  85 9D 34 7B 89 0E D6 89  42 31 DB 03 3E 4A 68 C0   ..4{.... B1..>Jh.
000000E0  04 48 D3 4C C1 4A 2B E2  06 E3 D5 13 BD 73 83 DC   .H.L.J+. .....s..
000000F0  1F C2 C7 70 1E 8C 53 36  B5 9E 8B 6B 0A 6C F9 E4   ...p..S6 ...k.l..
00000100  EB 07 48 43 55 4B 3C 3B  12 BA 0E 3A 9A 36 60 6E   ..HCUK<; ...:.6`n
00000110  AB 6F 46 7B 5C BB 74 4F  80 97 73 CA C6 E8 E6 15   .oF{\.tO ..s.....
00000120  81 ED 54 03 57 11 92 4F  2D 6D 02 F4 5B 32 2A 73   ..T.W..O -m..[2*s
00000130  D5 BC 4B 7E 4C 43 2F 25  E6 5C 5E 2B 44 12 C6 05   ..K~LC/% .\^+D...
00000140  6A 4A E5 7A 5D 00 2C 78  92 9C 5E 14 D0 FA BE 67   jJ.z].,x ..^....g
00000150  2A 71 16 DA 5A E7 10 61  A2 64 EF 4F 90 8B EC 0C   *q..Z..a .d.O....
00000160  1B 35 28 4C 63 0A 16 1F  46 96 17 AA F0 7D 9B 2B   .5(Lc... F....}.+
00000170  9B 02 11 AE F6 B9 77 94  3C 63 C2 19 95 1D 80 72   ......w. <c.....r
00000180  75 AB 7C FA 6A 46 05 0A  1E 43 0D A8 37 37 EA 1D   u.|.jF.. .C..77..
00000190  7A 03 41 B0 34 18 31 9C  05 C5 CF 94 29 42 F2 A4   z.A.4.1. ....)B..
000001A0  E6 F9 B3 2D 2B 49 A8 AB  5E C8 77 66 C1 08 E9 30   ...-+I.. ^.wf...0
000001B0  06 57 D3 E2 FF 8A A8 1B  87 51 4E D8 3E 14 B8 A0   .W..... .QN.>...
000001C0  7F C1 61 40 B7 B4 8A 46  69 98 B7 63 EA 5E 2B 52   .a@...F i..c.^+R
000001D0  57 64 A1 A3 89 C1 47 C7  AF BE A7 52 FE 28 A8 A5   Wd....G. ...R.(..
000001E0  E9 F1 77 45 B7 7A 3E D7  CD 75 15 21 0C 5E 54 D6   ..wE.z>. .u.!.^T.
000001F0  45 70 90 B6 5E 83 0A AB  15 D5 BF 62 71 96 18 3A   Ep..^... ...bq..:
00000200  73 8B 11 59 4B 6F 07 1D  E1 4B 36 10 B7 A3 25 C2   s..YKo.. .K6...%.
00000210  28 21 85 D3 58 74 22 4D  C5 4D DF 4A 83 7E AF 81   (!..Xt"M .M.J.~..
00000220  0D BF 8A 4D C0 CB 24 91  0F DE 96 30 03 60 88 4B   ...M..$. ...0.`.K
00000230  70 AF 71 42 0C 7A 0B 32  66 EB 47 81 D2 27 FE 96   p.qB.z.2 f.G..'..
00000240  CC CF 53 A8 E5 85 E0 0D  64 C4                     ..S..... d.
```

The Tor Project
122 Scott Circle
Dedham, MA 02026 USA
https://torproject.org/

From: Roger Dingledine, Tor Project Leader
To: Kelly DeYoe, program officer, IBB
RE: contract BBGCON1807S6441
Date: February 10, 2008

This report documents progress in January 2008 on contract BBGCON1807S6441 between IBB and The Tor Project.

C.2.0 New package releases and related software.

Tor 0.2.0.18-alpha (released Jan 25) adds a sixth v3 directory authority run by CCC, fixes a big memory leak in 0.2.0.17-alpha, and adds new config options that can warn or reject connections to ports generally associated with vulnerable-plaintext protocols.
http://archives.seul.org/or/talk/Jan-2008/msg00442.html

Tor 0.2.0.16-alpha and 0.2.0.17-alpha (released Jan 17) add a fifth v3 directory authority run by Karsten Loesing, and generally clean up a lot of features and minor bugs.
http://archives.seul.org/or/talk/Jan-2008/msg00254.html

Tor 0.1.2.19 (released Jan 17) fixes a huge memory leak on exit relays, makes the default exit policy a little bit more conservative so it's safer to run an exit relay on a home system, and fixes a variety of smaller issues.
http://archives.seul.org/or/announce/Jan-2008/msg00000.html

C.2.1 *The Contractor shall continue design, development and implementation of enhancements to the existing Tor software to increase its suitability as a tool for Internet users in countries with government-sponsored Internet censorship to circumvent censorship controls, based on the existing research and documentation performed during the previous contract period (e.g. as described in the paper "Design of a blocking-resistant anonymity system").*

Continuing enhancements have been made to the Tor website Chinese translation.

We continued work on the "BridgeDB" module: major progress on January was to improve robustness of the email subsystem so it is better at detecting forged mails that claim to be from gmail but are actually from elsewhere.

Work continued toward the upcoming Torbutton 1.1.13 release (which came out Feb 1). This new release has several significant security-related fixes:
https://torbutton.torproject.org/dev/CHANGELOG

Work continued toward the upcoming Vidalia 0.1.0 release: support for launching Firefox and Polipo as supporting applications; support for learning from Tor when the first circuit is ready so it can inform the user; and many other bugfixes including a few security fixes:
http://trac.vidalia-project.net/browser/vidalia/trunk/CHANGELOG

We added a "How do I find a bridge?" link and corresponding help text to Vidalia's 'Network' settings page.

From the Tor 0.2.0.16-alpha ChangeLog:
"Do not try to download missing certificates until we have tried to check our fallback consensus." This change gets us closer to being able to bootstrap without ever needing to contact the central directory authorities.

C.2.2 *The Contractor shall submit system architecture and technical design documentation for Tor enhancements specifically related to anti-censorship improvements in C.2.1 to the Authorized Representative of the Contracting Officer (AR/CO) for review and approval before development and implementation. Significant changes to the design that are discovered during development must be documented and reviewed by the AR/CO as soon as the Contractor becomes aware of the need for these revisions.*

New proposal "Version 2 Tor connection protocol" that specifies the details of our proposed new TLS handshake and how it interacts with current clients and servers:
https://www.torproject.org/svn/trunk/doc/spec/proposals/130-v2-conn-protocol.txt

New proposal "Block Insecure Protocols by Default" in collaboration with researchers at University of Colorado to warn and/or refuse users when they try to use ports commonly associated with vulnerable-plaintext protocols:

https://www.torproject.org/svn/trunk/doc/spec/proposals/129-reject-plaintext-ports

Implemented in Tor 0.2.0.18-alpha:
"New config options WarnPlaintextPorts and RejectPlaintextPorts so Tor can warn and/or refuse connections to ports commonly used with vulnerable-plaintext protocols. Currently we warn on ports 23, 109, 110, and 143, but we don't reject any."

Started work on a roadmap of all the future features and extensions we know we need. It's still mostly in outline form at this point:
https://www.torproject.org/svn/trunk/doc/design-paper/roadmap-future.pdf

C.2.3 *The Contractor shall develop and implement the bridge relay mechanism, as designed during the previous contract period, to allow individual Tor users to easily reconfigure their Tor client to automatically relay traffic from users in countries with government-imposed Internet censorship so as to circumvent that censorship.*

From the Tor 0.2.0.18-alpha ChangeLog:
"If we've gone 12 hours since our last bandwidth check, and we estimate we have less than 50KB bandwidth capacity but we could handle more, do another bandwidth test." Bridge relays that weren't getting any use were seeing their bandwidth estimate fall to 0 after the first few days of uptime.

C.2.4 *The Contractor shall develop and implement the bridge directory authority mechanism, as designed during the previous contract period, to allow Tor clients configured as bridge relays (as described in C.2.3) to communicate their existence to the bridge directory authority, and to provide a subset of addresses of available bridge relays to Tor users in countries with government-imposed Internet censorship so that they may access the Tor network.*

From the Tor 0.2.0.16-alpha ChangeLog:
"Make bridges round reported GeoIP stats info up to the nearest multiple of 8, not down. Now we can distinguish between "0 people from this country" and "1 person from this country", without needing to collect precise statistics."
"Bridge authorities are no longer willing to serve bridge descriptors over unencrypted connections." This will discourage people from writing tools that don't bother using encrypted connections.

C.2.5 *The Contractor shall design and develop revisions to the Tor network protocols to hide the network signature of Tor traffic so it is difficult for government-sponsored Internet censors to identify Tor traffic and trivially block it.*

We continued to deploy the new design for the normalized TLS handshake. Thanks to some assistance from an OpenSSL development team member, we were able to get closer to completing a new version-2 style TLS handshake. In early February we have successfully made such a handshake: so we expect that February will be the month when this feature finally rolls out.

C.2.6 *The Contractor shall design enhancements to Tor's cell-based protocol to improve*

performance on substandard network connections including those with low bandwidth and/or high latency and/or high packet loss.

No changes.

C.2.7 *The Contractor shall continue development of enhancements to improve the scalability of the Tor network toward the goal of supporting 2 million or more concurrent end users. This requirement is only a goal for system scalability and is not a requirement on number of actual concurrent users of the Tor network.*

We set up the Tor relay "gabelmoo" (run by Karsten Loesing) and "dannenberg" (run by CCC) as the fifth and sixth v3 directory authorities.

From the Tor 0.1.2.19 ChangeLog:
"Exit policies now reject connections that are addressed to a relay's public (external) IP address too, unless ExitPolicyRejectPrivate is turned off. We do this because too many relays are running nearby to services that trust them based on network address." This change will allow more people to run relays comfortably, thus expanding the network.
"Stop thinking that 0.1.2.x directory servers can handle "begin_dir" requests. Should ease bugs 406 and 419 where 0.1.2.x relays are crashing or mis-answering these types of requests."
"Fix a memory leak on exit relays; we were leaking a cached_resolve_t on every successful resolve. Reported by Mike Perry."

From the Tor 0.2.0.16-alpha ChangeLog:
"Major performance improvement: Switch our old ring buffer implementation for one more like that used by free Unix kernels. The wasted space in a buffer with 1MB of data will now be more like 8KB than 1MB. The new implementation also avoids realloc();realloc(); patterns that can contribute to memory fragmentation."

C.2.8 *The Contractor shall work with IBB staff and other IBB contractors to identify tasks in support of this program that might be developed collaboratively with Contractor. Tasks in areas such as documentation, bug fixes, software testing, and any other areas involving specific knowledge of foreign government-sponsored Internet censorship may be especially appropriate for this purpose.*

No reports for this month.

C.2.9 *The Contractor shall communicate tasks identified for delegation to IBB in C.2.8 to the AR/CO and negotiate time frames for their completion. The Contractor shall monitor and coordinate work performed by IBB staff on delegated tasks and integrate it into Tor software releases as appropriate.*

No reports for this month.

C.2.10 *The Contractor shall promote active growth of the Tor server network and advocacy of Tor*

products to increase the performance, stability, and usability of Tor.

Roger met with law enforcement in Stuttgart on Jan 4 to talk about the upcoming data retention law in the EU which may impact willingness to run Tor relays, and more specifically to teach the officers and investigators about how Tor works and how Internet security works. They were surprisingly interested to learn how to support Tor better; a further report will be on our blog sometime in February.

Roger also met with grad students at Indiana University who are working on research that will hopefully lead to a UDP transport design for Tor. Switching to UDP would let the network scale better, would allow us to provide more robust and faster traffic through Tor, and could allow us to support a wider range of protocols through Tor including voice over IP.

While at Indiana, Roger also talked to a Google representative who may be able to help Tor get more funded students in the 2008 Google Summer of Code program.

Nick Mathewson, Mike Perry, and Jake Appelbaum met with Mozilla Corporation to talk about Firefox bugs that Mike had unearthed while working on Torbutton. They also talked about branding issues, and getting permission to use the Firefox name even on our privacy-oriented Tor Browser Bundle. We may be able to get some funding from Mozilla to continue working on Torbutton and related applications.

Roger met with researchers at the Financial Cryptography conference. These included the Nymble researchers at Dartmouth University, who are working on mechanisms to let websites like Wikipedia allow anonymous users while still being able to refuse connections from Tor users who have abused them in the past. They also included researchers from Russia and Belarus who provided feedback on the blocking-resistance design. Finally, they included researchers at the University of Minnesota who have been working on a design that will make Tor more robust against certain anonymity-breaking attacks. Roger was selected to be the Program Chair for the conference next year.

We moved the TorStatus project to Tor's official SVN repository:
https://tor-svn.freehaven.net/svn/torstatus/trunk/
https://torstatus.kgprog.com/

C.2.11 *The Contractor shall improve the ease of use of Tor for end users by continuing research and development on one or both of the following products: (1) all-in-one software bundle containing Tor and supporting applications, as well as an easy-to-use installer for Microsoft Windows operating systems, as well as option to install and run from a Universal Serial Bus (USB) flash device; (2) bootable CD-ROM image ("LiveCD") which contains a minimal operating system, Tor, and supporting applications. Both would have all appropriate applications pre-configured to use Tor out of the box with only minimal additional configuration required by the end user. If Contractor determines it is not feasible to develop both products, Contractor will provide detailed written technical analysis and explanation to the AR/CO. The Contractor shall make an initial public release of at least one implementation*

of one of these products during the term of this contract.

The Tor Browser Bundle now has its own webpage, complete with an installation guide and screenshots:
https://torbrowser.torproject.org/

The new 0.0.6 Tor Browser Bundle (released Jan 29) includes Polipo, includes a newer Tor release, and fixes a few configuration aspects to make it more secure:
https://tor-svn.freehaven.net/svn/torbrowser/branches/stable/README

A new version of the Incognito Privacy LiveCD was released on Jan 26. It includes new versions of many components, and also some bugfixes on the USB support:
http://anonymityanywhere.com/incognito/

The Tor Project
122 Scott Circle
Dedham, MA 02026 USA
https://www.torproject.org/

From: Roger Dingledine, Tor Project Leader
To: Kelly DeYoe, program officer, BBG
RE: contract BBGCON1807S6441
Date: July 10, 2008

This report documents progress in June 2008 on contract BBGCON1807S6441 between BBG and The Tor Project.

C.2.0 New package releases and related software.

Torbutton 1.2.0rc1 (released June 1), the first release candidate for the next stable series of the security-enhanced Torbutton Firefox extension, features functional support for Firefox 3. However, this support has not been extensively tested. In particular, timezone masking does not work at all. The workaround is to manually set the environment variable 'TZ' to 'UTC' before starting Firefox. This works on both Linux and Windows:
http://archives.seul.org/or/talk/Jun-2008/msg00044.html

Tor 0.2.0.27-rc (released June 3) adds a few features we left out of the earlier release candidates. In particular, we now include an IP-to-country GeoIP database, so controllers can easily look up what country a given relay is in, and so bridge relays can give us some sanitized summaries about which countries are making use of bridges. (See proposal 126-geoip-fetching.txt for details.)
http://archives.seul.org/or/talk/Jun-2008/msg00055.html

Torbutton 1.2.0rc2 (released June 8) features a fix for an annoying bug on MacOS, and adds much clamored for options to start Firefox in a specific Tor state:
http://archives.seul.org/or/talk/Jun-2008/msg00103.html

Tor 0.2.0.28-rc (released June 13) fixes an anonymity-related bug, fixes a hidden-service performance bug, and fixes a bunch of smaller bugs.
http://archives.seul.org/or/talk/Jun-2008/msg00165.html

Tor 0.2.1.1-alpha (released June 13) fixes a lot of memory fragmentation problems that were making the Tor process bloat especially on Linux; makes our TLS handshake blend in better; sends "bootstrap phase" status events to the controller, so it can keep the user informed of progress (and problems) fetching directory information and establishing circuits; and adds a variety of smaller features.

http://archives.seul.org/or/talk/Jun-2008/msg00185.html

Vidalia 0.1.4 (released June 13) adds a bootstrap progress bar, UPnP support, a new set of freely licensed GUI icons, and fixes a few bugs.
http://trac.vidalia-project.net/browser/vidalia/tags/vidalia-0.1.4/CHANGELOG

The Tor Browser Bundle 1.1.0 (released June 13) replaces startup batch script with application (RelativeLink) so there is a helpful icon, optionally installs Pidgin (for Tor IM Browser Bundle), optionally uses WinRAR to produce a self-extracting split bundle, and includes upgraded versions of Tor, Vidalia, and Torbutton.
https://svn.torproject.org/svn/torbrowser/trunk/README

Tor 0.2.1.2-alpha (released June 20) includes a new "TestingTorNetwork" config option to make it easier to set up your own private Tor network; fixes several big bugs with using more than one bridge relay; fixes a big bug with offering hidden services quickly after Tor starts; and uses a better API for reporting potential bootstrapping problems to the controller.
http://archives.seul.org/or/talk/Jun-2008/msg00247.html

Vidalia 0.1.5 (released June 21) switches Vidalia's internal string representation so it can use the new Pootle-based translation system.
http://trac.vidalia-project.net/browser/vidalia/tags/vidalia-0.1.5/CHANGELOG

Torbutton 1.2.0rc3 and 1.2.0rc4 (both released June 27) provide improved addon compatibility, better preservation of Firefox preferences that we touch, fixing issues with Tor toggle breaking for some option combos, and an improved 'Restore Defaults' button.
https://torbutton.torproject.org/dev/CHANGELOG

C.2.1 The Contractor shall continue design, development and implementation of enhancements to the existing Tor software to increase its suitability as a tool for Internet users in countries with government-sponsored Internet censorship to circumvent censorship controls, based on the existing research and documentation performed during the previous contract period (e.g. as described in the paper "Design of a blocking-resistant anonymity system").

We continued enhancements to the Chinese and Russian Tor website translations. We started working with Farsi translators to produce Vidalia, Torbutton, and website translations into Farsi; we expect those to be included in a July release.

From the Tor 0.2.1.1-alpha ChangeLog:
"When we choose to abandon a new entry guard because we think our older ones might be better, close any circuits pending on that new entry guard connection. This fix should make us recover much faster when our network is down and then comes back. Bugfix on 0.1.2.8-beta; found by lodger."

From the Tor 0.2.0.28-rc ChangeLog:
"Fix a bug where, when we were choosing the 'end stream reason' to put in our relay end cell that we send to the exit relay, Tor clients on Windows were sometimes sending the wrong 'reason'. The anonymity problem is that exit relays may be able to guess whether the client is running

Windows, thus helping partition the anonymity set. Down the road we should stop sending reasons to exit relays, or otherwise prevent future versions of this bug."

We also added bootstrap status events and bootstrap problem events, plus a variety of efficiency improvements for directory download overhead; see Section C.2.2 below.

C.2.2 The Contractor shall submit system architecture and technical design documentation for Tor enhancements specifically related to anti-censorship improvements in C.2.1 to the Authorized Representative of the Contracting Officer (AR/CO) for review and approval before development and implementation. Significant changes to the design that are discovered during development must be documented and reviewed by the AR/CO as soon as the Contractor becomes aware of the need for these revisions.

We finally got around to writing down the details of many of our architecture and technical design changes:

Proposal 137 ("Keep controllers informed as Tor bootstraps") modifies Tor so it keeps Vidalia informed of each "bootstrap phase" -- that is, progress Tor makes at learning directory information, making connections to the network, etc. Now Vidalia has a progress bar on Tor startup that explains what's going on. Further, Tor reports "bootstrap problems" when it believes it's having troubles starting up correctly, and Vidalia can now tell the user. All of this is in as of the Tor 0.2.1.2-alpha release (June 20).
https://svn.torproject.org/svn/tor/trunk/doc/spec/proposals/137-bootstrap-phases.txt

Proposal 138 ("Remove routers that are not Running from consensus documents") modifies the directory "networkstatus consensus" documents so they no longer list relays that are believed to be unusable. They used to list these relays so clients could decide for themselves, but in practice clients just ignored them. This change saves 30% to 40% in download bandwidth for consensus documents. It is included in the 0.2.1.2-alpha release (June 20).
https://svn.torproject.org/svn/tor/trunk/doc/spec/proposals/138-remove-down-routers-from-consensus.txt

Proposal 139 ("Download consensus documents only when it will be trusted") tries to make Tor clients better handle the case when new directory authorities have been added to the system, or when directory authorities have changed (for example, this could happen if we have another bug like the one in May that caused us to change keys for half the directory authorities). Now clients specify which directory authorities they trust, so the directory mirrors can give them a consensus document they'll be willing to use. This change is included in Tor 0.2.1.1-alpha, and a bugfix on it was included in Tor 0.2.1.2-alpha.
https://svn.torproject.org/svn/tor/trunk/doc/spec/proposals/139-conditional-consensus-download.txt

Proposal 140 ("Provide diffs between consensuses") is still under development, but is scheduled to be included in the Tor 0.2.1.x tree. The idea is that most parts of the consensus document don't change from one hour to the next, so we can give clients a diff on the previous one rather than a whole new document, changing the size of the document every client must download every few hours from 92KB on average to 13KB on average.
https://svn.torproject.org/svn/tor/trunk/doc/spec/proposals/140-consensus-diffs.txt

Proposal 141 ("Download server descriptors on demand") is still under discussion, and may not be ready until for inclusion until Tor 0.2.2.x. This is the more detailed version of our "grand scaling plan" first mentioned in April. The idea is to have clients download networkstatus consensus documents as they do now, but rather than preemptively fetching every relay descriptor just in case, they fetch descriptors "just in time" only when they need them. The trick is to keep the bandwidth overhead low while not introducing too many new anonymity attacks e.g. due to leaking which relays you're picking.
https://svn.torproject.org/svn/tor/trunk/doc/spec/proposals/141-jit-sd-downloads.txt

We've instrumented a Tor client to collect stats on how much bandwidth we use now for directory overhead and how much we'd save with this new approach:
http://archives.seul.org/or/dev/Jun-2008/msg00024.html

Proposals 142 ("Combine Introduction and Rendezvous Points") and 143 ("Improvements of Distributed Storage for Tor Hidden Service Descriptors") are still in the discussion phase. Their goal is to improve the experience for clients accessing Tor hidden services, both by making the handshake faster and by making hidden service reachability more reliable and more robust.
https://svn.torproject.org/svn/tor/trunk/doc/spec/proposals/142-combine-intro-and-rend-points.txt
https://svn.torproject.org/svn/tor/trunk/doc/spec/proposals/143-distributed-storage-improvements.txt

C.2.5 The Contractor shall design and develop revisions to the Tor network protocols to hide the network signature of Tor traffic so it is difficult for government-sponsored Internet censors to identify Tor traffic and trivially block it.

The "spoofing Firefox cipher suites and extensions" features are now in the Tor 0.2.1.1-alpha release, meaning they're in the Tor Browser Bundle 1.1.0 release also. From the 0.2.1.1-alpha ChangeLog:
"More work on making our TLS handshake blend in: modify the list of ciphers advertised by OpenSSL in client mode to even more closely resemble a common web browser. We cheat a little so that we can advertise ciphers that the locally installed OpenSSL doesn't know about."

We've done some initial security auditing (though there's always room for more, and we plan to do some more concrete auditing in July).

Nick also wrote some early thoughts on doing pass-through to an Apache server to improve scanning resistance:
http://archives.seul.org/or/dev/Jun-2008/msg00014.html

C.2.10 *The Contractor shall promote active growth of the Tor server network and advocacy of Tor products to increase the performance, stability, and usability of Tor.*

Roger talked to Vitaly Shmatikov (CS professor at UTexas) about the end of our SRI project making Tor more suitable for anonymizing IDS alert data; sent a summary of progress to Phil Porras, and also pitched the "Tor can be used to investigate malware sites without letting them know your project is the one investigating them" use of Tor, since Phil is working on malware analysis and collection now. A greater variety of uses will make the Tor network and development

effort more sustainable and more robust against social / legal attack.

Roger is going to be doing a Defcon talk, since Jeff Moss (the Defcon organizer) called him and asked him to do one on "Current attacks in Tor".
http://defcon.org/html/defcon-16/dc-16-speakers.html#Dingledine

Roger asked Angelos D. Keromytis for a copy of his paper doing a clogging / bandwidth measurement attack on Tor. The general consensus seems to be that the attack may well work, but the analysis section of the paper really needs more work before we can have any intuition about when the attack works and when it doesn't work -- which is needed before we can start working on ways to mitigate the issue. There's an early thread discussing it on or-talk:
http://archives.seul.org/or/talk/Jun-2008/msg00162.html
Next steps are to open a discussion with the authors, and try to help them make a more convincing evaluation section.

Roger started talking to Christian Grothoff about his upcoming Defcon talk presenting an anonymity attack on Tor. It turns out to be based on the "infinite-length circuits" DoS attack that Christian came up with back in 2005 (and which we're slowly addressing with proposal 110). Christian sent me his slides and a draft of the attack paper, and we're keeping in touch.

Roger and Andrew had lunch on Jun 3 with Rob Faris of the Berkman Center. We explained all the various ways that Tor can be useful as a building block for e.g. ONI or Berkman's "distributed app". I saw Rob again at the Global Voices Summit where he was presenting an overview of circumvention issues, so I think prepping him was (is?) a good investment.

Dmitri Vitaliev et al from Tactical Tech are getting closer to having their NGO-in-a-box ready, and they're slowly trying to write docs on using Tor safely and then translate them. I offered that we would be happy to look over and correct any docs they have. I also learned at the GV summit that they have a "by September" deadline that they are going to have a great deal of trouble meeting.

Rebecca MacKinnon contacted us about a newspaper owner in Hong Kong who wants to fund the deployment of many many Tor Browser Bundle USB keys, so customers can go into China and still be able to reach the newspaper website. I handed the topic to Andrew and Isaac so they could help pursue it further.

I helped Julius Mittenzwei (our lawyer contact at CCC in Germany) by providing "expert witness" details that in fact a particular IP address in Germany was running a Tor exit node at a given point in time (in June 2007 in this case). Sounds like he's busy behind the scenes defending Germany Tor operators, though he doesn't mention it to us much.

I started chatting with Martin Peck (a currently volunteer developer) about having him do some contract work for us on a VM-based Tor install. I'm hoping to pay him quarter-time for 3-4 months and then we can evaluate whether we should hire him full-time (and by then we will hopefully have the funding for it also).

Nick, Geoff Goodell, and Andrew had dinner with Valer Mischenko of NLnet last week. We had

an informal peer review of a few projects NLnet is considering funding. In general, we had a great discussion about the future of The Tor Project, Tor itself, and other applications where Tor might be relevant. Valer wore his Tor t-shirt for his appearances at USENIX,
http://www.usenix.org/events/usenix08/

Jacob Appelbaum gave a talk at IBM research in Zurich, and got the researchers there thinking more about anonymity as well as their usual enterprise privacy.

China has finally started blocking our website. Isaac and others started talking about mechanisms to deal with that. I helped Jacob set up an automatic script to test our mirrors and generate the mirror page. Jacob has also been working on a "gettor" email auto-responder. We need to work harder to let people know there are Tor mirrors inside the GFW too.

I went to the Global Voices Summit in Budapest (Jun 26-28), where I gave two talks -- one a 45 minute workshop talk to give the core audience a good idea of what's up in Tor and blocking-resistance, and another a 10-minute blurb on a panel that was opened up to the whole crowd. I gave a variety of interviews for various media places, met a few funders that I've passed on to Andrew, talked at length with Isaac, met some folks from Russia and Iran (among others) who may be able to help us, and convinced Joi Ito (a famous Japanese blogger) he wants to run a v3 directory authority for us.

C.2.11 *The Contractor shall improve the ease of use of Tor for end users by continuing research and development on one or both of the following products: (1) all-in-one software bundle containing Tor and supporting applications, as well as an easy-to-use installer for Microsoft Windows operating systems, as well as option to install and run from a Universal Serial Bus (USB) flash device; (2) bootable CD-ROM image ("LiveCD") which contains a minimal operating system, Tor, and supporting applications. Both would have all appropriate applications pre-configured to use Tor out of the box with only minimal additional configuration required by the end user. If Contractor determines it is not feasible to develop both products, Contractor will provide detailed written technical analysis and explanation to the AR/CO. The Contractor shall make an initial public release of at least one implementation of one of these products during the term of this contract.*

The Tor Browser Bundle 1.1.0 (released June 13) replaces startup batch script with application (RelativeLink) so there is a helpful icon, optionally installs Pidgin (for Tor IM Browser Bundle), optionally uses WinRAR to produce a self-extracting split bundle, and includes upgraded versions of Tor, Vidalia, and Torbutton.

We also looked into running two Firefoxes in parallel:
https://svn.torproject.org/svn/torbrowser/trunk/docs/two-firefox.txt
and we even hacked in some Torbutton fixes that will come out in version 1.2.0rc3 that should get us closer:
http://archives.seul.org/or/cvs/Jun-2008/msg00213.html

Speaking of which, we also hacked in another feature in Torbutton 0.1.2rc2, to add a "locked" mode so Tor Browser Bundle can start Torbutton and not fear that the user will click and disable Tor. I believe TBB 1.1.0 doesn't use this feature yet though.
http://archives.seul.org/or/cvs/Jun-2008/msg00186.html

C.2.12 *The Contractor shall continue to develop and implement improvements to the bridge relay and bridge directory authority mechanisms to improve the usability, performance and reliability of the Tor network by users in countries with government-imposed Internet censorship.*

From the Tor 0.2.1.2-alpha ChangeLog:
"If you have more than one bridge but don't know their digests, you would only learn a request for the descriptor of the first one on your list. (Tor considered launching requests for the others, but found that it already had a connection on the way for $0000...0000 so it didn't open another.) Bugfix on 0.2.0.x."
"If you have more than one bridge but don't know their digests, and the connection to one of the bridges failed, you would cancel all pending bridge connections. (After all, they all have the same digest.) Bugfix on 0.2.0.x."
"If you're using bridges, generate "bootstrap problem" warnings as soon as you run out of working bridges, rather than waiting for ten failures -- which will never happen if you have less than ten bridges."

We put up a new webpage to describe bridges, how to fetch bridge relay addresses, etc:
https://www.torproject.org/bridges

We also modified the BridgeDB database (that is, the server that runs https://bridges.torproject.org/ and answers mail to bridges@torproject.org) to autodetect if the address hitting https://bridges.torproject.org/ is currently a Tor exit relay, and if so to treat it specially -- that is, we reserve a set of bridge addresses and give those out only to folks coming in over Tor.

The updated BridgeDB version now makes sure to give out at least one bridge that's listed as Stable in the bridge authority's networkstatus document, and at least one bridge that listens on port 443. The goal here is to increase the odds that at least one of the bridges we give the user will be usable even if he's in a tightly firewalled situation.

From the Tor 0.2.0.27-rc ChangeLog:
"Include an IP-to-country GeoIP file in the tarball, so bridge relays can report sanitized summaries of the usage they're seeing."

We've started collecting "geoip-clients" lines from bridge relays, to get a handle on which countries are using bridges and how much use they're seeing. We sent an update on June 11 with some example bridge relays and the users-per-country they were reporting. We expect to start gathering more organized data points for this in the July / August timeframe.

C.2.13 *The Contractor shall research and document additional options for the scalability of the Tor* network *beyond 2 million concurrent users, including analysis of splitting the network into multiple segments, switching to datagram-based protocols, and improving the load balancing within the network.*

We finished work on a patch for OpenSSL that will make it keep less buffer space around. Currently fast Tor relays use (waste) as much as 100M of memory in OpenSSL's buffers. This

patch was accepted and included in the main OpenSSL tree in June:
http://marc.info/?l=openssl-cvs&m=121246471627426&w=2

The Vidalia 0.1.4 release has folded the UPnP library and GUI changes into the main Vidalia tree, along with a "test" button to try speaking UPnP at the local router and tell the user whether it worked; these features will be available by default in the 0.2.0.x stable release.

We've put a lot of effort into reducing Tor's memory footprint again. The main issue was a "memory fragmentation" problem in Linux's memory allocator, which was causing Tor servers on Linux to slowly grow without bound. As of Tor 0.2.1.2-alpha, the issue appears to be substantially better. Many more details are here:
http://archives.seul.org/or/dev/Jun-2008/msg00001.html

From the Tor 0.2.1.2-alpha ChangeLog:
"New TestingTorNetwork config option to allow adjustment of previously constant values that, while reasonable, could slow bootstrapping. Implements proposal 135. Patch from Karsten Loesing."
"When building a consensus, do not include routers that are down. This will cut down 30% to 40% on consensus size. Implements proposal 138."

C.2.14 The *Contractor shall continue research into the option of providing incentives for Tor users to run Tor relay servers. If further research indicates that this should be pursued, the Contractor shall develop a project plan and timeline for this work. If further research indicates this option should be abandoned, the Contractor shall document and explain in writing the reasoning behind this decision.*

[No updates]

C.2.15 *The Contractor shall develop a more reliable download mechanism for the Tor browser bundle for users on slow and/or unreliable network connections, by means of a split download of multiple smaller files, implementation of a lightweight download manager, reduction in the software bundle file size, or other method as chosen by the Contractor.*

We've added clear user-oriented instructions for the Tor Browser Bundle split-download page:
https://www.torproject.org/torbrowser/split.html.en

We're starting work on a "gettor" email auto-responder script that will let people mail gettor@torproject.org and retrieve a copy of Tor from their mailbox. More info forthcoming in July.

More generally, we have a new https://www.torproject.org/finding-tor page that describes various mechanisms such as mirrors.

In July we plan to deploy a more automated mechanism for tracking which Tor mirrors are up-to-date.

C.2.16 The *Contractor shall test the Tor browser bundle on multiple computer systems and analyze these systems afterwards for any changes to the system that may have been made inadvertently by use*

of the Tor browser bundle. The Contractor shall document any such changes found and develop a plan to reduce the footprint of Tor browser bundle use.

We started evaluating the footprints here:
https://svn.torproject.org/svn/torbrowser/trunk/docs/traces.txt

The "Windows Prefetch trace" issue appears to be pervasive and hard to fix. I've started talking to a friend who runs a security forensics company in the UK about how we can mitigate this issue, but so far there are no good ways.

It appears that Windows Vista will have enough more issues like the Prefetch trace issue; we should hope that Internet cafes don't move to Vista anytime soon.

C.2.17 *The Contractor shall develop or adapt existing open source software to implement a web-based portal to manage the translations of text into multiple languages for the user interface text of* software *of Torbutton and Vidalia and other software that may in the future be included in the Tor browser bundle. The web site must allow non-technical users the ability to contribute translations by providing text to be translated in English, as well as any needed context on the use of the text, and allowing users to enter the translation into their language from their web browser.*

We have our translation server up and online:
https://translation.torproject.org/

We have imported the strings from Vidalia, Torbutton, and Torcheck, and we currently have active translations for Spanish, French, German, Italian, Polish, Romanian, Swedish, Turkish, Finnish, Russian, Chinese, and Arabic.

We have a more useful overall translation tutorial here:
https://www.torproject.org/translation-portal

And we have internal documentation here for how to deal with the translation stuff behind the scenes:
https://svn.torproject.org/svn/tor/trunk/doc/translations.txt

In July we plan to add the strings for Vidalia's installer; the challenge is that we need to write a script to convert from the "nsh" (nullscript installer language) format to the "po" (preferred by Pootle) format and back.

In July we also expect to see the first version of our "wml to po and back" conversion tool, that will allow us to start putting our website pages into the translation server.

The Tor Project
122 Scott Circle
Dedham, MA 02026 USA
https://torproject.org/

From: Roger Dingledine, Tor Project Leader
To: Kelly DeYoe, program officer, IBB
RE: contract BBGCON1807S6441
Date: April 10, 2008

This report documents progress in March 2008 on contract BBGCON1807S6441 between IBB and The Tor Project.

C.2.0 New package releases and related software.

Tor 0.2.0.23-rc (released Mar 24) is the fourth release candidate for the 0.2.0 series. It makes bootstrapping faster if the first directory mirror you contact is down. The bundles also include the new Vidalia 0.1.2 release.
http://archives.seul.org/or/talk/Mar-2008/msg00204.html

Tor 0.2.0.22-rc (released Mar 18) is the third release candidate for the 0.2.0 series. It enables encrypted directory connections by default for non-relays, fixes some broken TLS behavior we added in 0.2.0.20-rc, and resolves many other bugs. The bundles also include Vidalia 0.1.1 and Torbutton 1.1.17.
http://archives.seul.org/or/talk/Mar-2008/msg00136.html

Tor 0.2.0.21-rc (released Mar 2) is the second release candidate for the 0.2.0 series. It makes Tor work well with Vidalia again, fixes a rare assert bug, and fixes a pair of more minor bugs. The bundles also include Vidalia 0.1.0 and Torbutton 1.1.16.
http://archives.seul.org/or/talk/Mar-2008/msg00025.html

Torbutton 1.1.16 (released Mar 3) and 1.1.17 (released Mar 15) fix many more potential privacy and identity leaks, mostly based on exploits found by Greg Fleischer, and try to start adding support for Firefox 3.
https://torbutton.torproject.org/dev/CHANGELOG

Vidalia 0.1.0 (released Mar 1), 0.1.1 (released Mar 17), and 0.1.2 (released Mar 24) changes the build process from make to cmake, starts doing encrypted geoip fetches rather than plaintext geoip fetches, checks if the user is running a dangerous or obsolete version of Tor and pops up a window warning them, waits to turn the Vidalia taskbar onion green until Tor reports that it has established a circuit, folds in the patches from Tor Browser Bundle to have Vidalia launch a

browser and/or an http proxy, and fixes many miscellaneous bugs.
http://trac.vidalia-project.net/browser/vidalia/tags/vidalia-0.1.2/CHANGELOG

C.2.1 *The Contractor shall continue design, development and implementation of enhancements to the existing Tor software to increase its suitability as a tool for Internet users in countries with government-sponsored Internet censorship to circumvent censorship controls, based on the existing research and documentation performed during the previous contract period (e.g. as described in the paper "Design of a blocking-resistant anonymity system").*

We continued enhancements to the Chinese and Russian Tor website translations.

From the Tor 0.2.0.23-rc ChangeLog:
"When a tunneled directory request is made to a directory server that's down, notice after 30 seconds rather than 120 seconds. Also, fail any begindir streams that are pending on it, so they can retry elsewhere. This was causing multi-minute delays on bootstrap."

From the Tor 0.2.0.22-rc ChangeLog:
"Enable encrypted directory connections by default for non-relays, so censor tools that block Tor directory connections based on their plaintext patterns will no longer work. This means Tor works in certain censored countries by default again."

From the Vidalia 0.1.1 ChangeLog:
"TunnelDirConns and PreferTunneledDirConns are now enabled by default as of Tor 0.2.0.22-rc. Don't check the 'My ISP blocks connections to the Tor network' box simply because TunnelDirConns is enabled. Checking the box still enables encrypted directory connections on older Tors."

From the Vidlia 0.1.0 ChangeLog:
"Listen for the DANGEROUS_VERSION general status event and warn the user if their version of Tor is no longer recommended."
"Listen for the CIRCUIT_ESTABLISHED client status event and only turn the yellow onion status icon green after Tor has successfully established a circuit."
"Add a "How do I find a bridge?" link and corresponding help text to the 'Network' settings page."
"Add a 'BrowserExecutable' configuration option to launch a Web browser when Tor has built a circuit, and exit Vidalia when the browser is closed."
"Add 'ProxyExecutable' and 'ProxyExecutableArguments' configuration options to launch a proxy application with given parameters when Vidalia starts, and close it when Vidalia exits."
"Rename the 'Relay' settings page to the 'Sharing' settings page."

We also prepared a Russian-language version of the Tor Browser Bundle, drafted a blurb for getting more IBB/BBG people involved, and hopefully moved us further down the track of deploying a beta test in Russia.

C.2.2 *The Contractor shall submit system architecture and technical design documentation for Tor*

enhancements specifically related to anti-censorship improvements in C.2.1 to the Authorized Representative of the Contracting Officer (AR/CO) for review and approval before development and implementation. Significant changes to the design that are discovered during development must be documented and reviewed by the AR/CO as soon as the Contractor becomes aware of the need for these revisions.

We've started to work on a design for allowing Tor users to verify whether they've got Tor configured correctly in their browser without hitting as many false positives or false negatives. The basic idea is to intercept the website request inside the Tor client, and provide a confirmation page back. Then also set up a real website at that same address, to give back a page explaining that there's a configuration problem and giving some tips.
https://www.torproject.org/svn/trunk/doc/spec/proposals/131-verify-tor-usage.txt

C.2.3 *The Contractor shall develop and implement the bridge relay mechanism, as designed during the previous contract period, to allow individual Tor users to easily reconfigure their Tor client to automatically relay traffic from users in countries with government-imposed Internet censorship so as to circumvent that censorship.*

From the Tor 0.2.0.21-rc ChangeLog:
"We were sometimes miscounting the number of bytes read from the network, causing our rate limiting to not be followed exactly. Bugfix on 0.2.0.16-alpha. Reported by lodger."

From the Vidalia 0.1.2 ChangeLog:
"Bridges are no longer required to have a DirPort set as of Tor 0.2.0.13-alpha, so stop forcing it on for bridges. At some point, we'll likely start forcing DirPort to be disabled for bridges, and on by default but optional for normal relays."

C.2.4 *The Contractor shall develop and implement the bridge directory authority mechanism, as designed during the previous contract period, to allow Tor clients configured as bridge relays (as described in C.2.3) to communicate their existence to the bridge directory authority, and to provide a subset of addresses of available bridge relays to Tor users in countries with government-imposed Internet censorship so that they may access the Tor network.*

No changes.

C.2.5 *The Contractor shall design and develop revisions to the Tor network protocols to hide the network signature of Tor traffic so it is difficult for government-sponsored Internet censors to identify Tor traffic and trivially block it.*

From the Tor 0.2.0.22-rc ChangeLog:
"Enable encrypted directory connections by default for non-relays, so censor tools that block Tor directory connections based on their plaintext patterns will no longer work. This means Tor works in certain censored countries by default again."
"Make sure servers always request certificates from clients during TLS renegotiation. Reported by lodger; bugfix on 0.2.0.20-rc."

C.2.6 *The Contractor shall design enhancements to Tor's cell-based protocol to improve performance on substandard network connections including those with low bandwidth and/or high latency and/or high packet loss.*

We've started to work on a design that would allow clients to fetch only the server descriptors they actually need to build their circuits. Rather than fetching every server descriptor preemptively just in case it's needed, clients should instead fetch each descriptor on demand as they're extending their circuit. Since they would fetch the descriptor from the next hop in the circuit, they're not revealing any private information (like they would be if they were fetching it on demand from a central location). There are still many anonymity concerns with this approach though; we're aiming to write a more thorough proposal in the next month or two.

C.2.7 *The Contractor shall continue development of enhancements to improve the scalability of the Tor network toward the goal of supporting 2 million or more concurrent end users. This requirement is only a goal for system scalability and is not a requirement on number of actual concurrent users of the Tor network.*

We fixed a wide variety of bugs in the "openbsd-malloc" option we added last month in Tor 0.2.0.20-rc. Now the Debian package ships with this option enabled by default, so fast Tor relays running on Debian will use much less ram by default.

C.2.8 *The Contractor shall work with IBB staff and other IBB contractors to identify tasks in support of this program that might be developed collaboratively with Contractor. Tasks in areas such as documentation, bug fixes, software testing, and any other areas involving specific knowledge of foreign government-sponsored Internet censorship may be especially appropriate for this purpose.*

No reports for this month.

C.2.9 *The Contractor shall communicate tasks identified for delegation to IBB in C.2.8 to the AR/CO and negotiate time frames for their completion. The Contractor shall monitor and coordinate work performed by IBB staff on delegated tasks and integrate it into Tor software releases as appropriate.*

No reports for this month.

C.2.10 *The Contractor shall promote active growth of the Tor server network and advocacy of Tor products to increase the performance, stability, and usability of Tor.*

Roger presented at the Source Boston conference on March 12. He also met with Frank Rieger during the conference to discuss potential grants from NLNet in the Netherlands to work on making Tor more suitable for users with low bandwidth, e.g. on modems or cell phones. We applied for two such grants, and will know in late April how that turned out.

Roger also spoke at Harvard's Center for Research in Computation and Society on March 19. The goal here was to expose the researchers there to "the wikipedia problem": how do sites like Wikipedia accept useful content from some users while keeping the jerks out -- and how does this answer change when all the users are anonymous? This discussion prepared the audience for a talk in April from the Nymble group at Dartmouth, who are working on a solution to this problem.

We continued working toward being able to hire Jake Appelbaum and Matt Edman as contractors in April or May. Jake will be starting in mid-April and will be working on a translation portal, auto update for Tor and supporting applications, a Windows buildbot, and other advocacy projects. Matt will be starting in May and working on Vidalia maintenance, bugfixes, and new features --- for example, providing a GUI interface for the above auto update feature, letting users change their preferred language in Vidalia without requiring an application restart, and providing a better GUI for showing Tor's start-up progress.

Roger finally got his blog post up about his experiences talking to Germany law enforcement in Stuttgart in early January.
https://blog.torproject.org/blog/talking-german-police-stuttgart

We were accepted into the Google Summer of Code 2008 program, in collaboration with The Electronic Frontier Foundation. We got 40 applications, and expect to get roughly 6 slots for student interns to work with us over the summer and be funded by Google.
https://blog.torproject.org/blog/tor-project-google-summer-code-2008%21

C.2.11 *The Contractor shall improve the ease of use of Tor for end users by continuing research and development on one or both of the following products: (1) all-in-one software bundle containing Tor and supporting applications, as well as an easy-to-use installer for Microsoft Windows operating systems, as well as option to install and run from a Universal Serial Bus (USB) flash device; (2) bootable CD-ROM image ("LiveCD") which contains a minimal operating system, Tor, and supporting applications. Both would have all appropriate applications pre-configured to use Tor out of the box with only minimal additional configuration required by the end user. If Contractor determines it is not feasible to develop both products, Contractor will provide detailed written technical analysis and explanation to the AR/CO. The Contractor shall make an initial public release of at least one implementation of one of these products during the term of this contract.*

Tor Browser Bundle 1.0.0 (released Mar 20) and 1.0.1 (released Mar 26) makes it work correctly with Polipo again, updates the versions of many of its components, and makes it easier to build the Bundle with custom included "jar" (plug-in) files as well as "xpi" (extension) files.
https://tor-svn.freehaven.net/svn/torbrowser/trunk/README

We moved the Tor Browser Bundle website into the main Tor website, so it can re-use our translation infrastructure. Currently its frontpage is available in English, German, Italian, Polish, and Russian.

The Tor Project
122 Scott Circle
Dedham, MA 02026 USA
https://www.torproject.org/

From: Roger Dingledine, Tor Project Leader
To: Kelly DeYoe, program officer, IBB
RE: contract BBGCON1807S6441
Date: June 10, 2008

This report documents progress in May 2008 on contract BBGCON1807S6441 between IBB and The Tor Project.

C.2.0 New package releases and related software.

Tor 0.2.0.26-rc (released May 13) fixes a major security vulnerability caused by a bug in Debian's OpenSSL packages. All users running any 0.2.0.x version should upgrade, whether they're running Debian or not.
http://archives.seul.org/or/talk/May-2008/msg00048.html

Vidalia 0.1.3 (released May 25) adds a hidden service configuration UI designed and implemented by Domenik Bork, as well as a few other bugfixes.
http://trac.vidalia-project.net/browser/vidalia/tags/vidalia-0.1.3/CHANGELOG

The Tor Browser Bundle 1.0.2 (released May 3) and 1.0.3 (released May 16) include upgraded versions of Tor, Vidalia, Torbutton, and Firefox.

We added three new part-time developers in May. We hired Matt Edman as a part-time employee at the beginning of May, to work on Vidalia maintenance, bugfixes, and new features. We also are funding Karsten Loesing to work on making hidden service rendezvous and interaction faster, and Peter Palfrader to work on lowering the overhead of directory requests, especially during bootstrap, which should directly improve the experience for Tor users on modems or cell phones.

Google has agreed to give us some funding to work on auto-update for Windows. Our plan is for Vidalia to look at the majority-signed network status consensus to decide when to update and to what version (Tor already lists what versions are considered safe, in each network status document). We should actually do the update via Tor if possible, for additional privacy, and we need to make sure to check package signatures to ensure package validity. Last, we need to give the user an interface for these updates, including letting her opt to migrate from one major Tor version to the next.

C.2.1 The Contractor shall continue design, development and implementation of enhancements to the existing Tor software to increase its suitability as a tool for Internet users in countries with government-sponsored Internet censorship to circumvent censorship controls, based on the existing research and documentation performed during the previous contract period (e.g. as described in the paper "Design of a blocking-resistant anonymity system").

We continued enhancements to the Chinese and Russian Tor website translations. Vidalia also added a Turkish translation.

From the Vidalia 0.1.3 ChangeLog:
"If we're running Tor >= 0.2.0.13-alpha, then check the descriptor annotations for each descriptor before deciding to do a geoip lookup on its IP address. If the annotations indicate it is a special purpose descriptor (e.g., bridges), then don't do the lookup at all."

"Remove the 'Run Tor as a Service' checkbox. Lots of people seem to be clicking it even though they don't really need to, and we end up leaving them in a broken state after a reboot."

"Only display the running relays in the big list of relays to the left of the network map. Listing a big pile of unavailable relays is not particularly useful, and just clutters up the list."

We worked toward a Torbutton 1.2.0rc1 release candidate, which will include support for Firefox 3 along with a huge pile of privacy-related bugfixes.

C.2.2 The Contractor shall submit system architecture and technical design documentation for Tor enhancements specifically related to anti-censorship improvements in C.2.1 to the Authorized Representative of the Contracting Officer (AR/CO) for review and approval before development and implementation. Significant changes to the design that are discovered during development must be documented and reviewed by the AR/CO as soon as the Contractor becomes aware of the need for these revisions.

We spent much of the first half of May dealing with a surprise massive security vulnerability in a crypto library that comes with Debian:
http://archives.seul.org/or/announce/May-2008/msg00000.html

You can read a more detailed explanation of the effects of the flaw here:
https://blog.torproject.org/blog/debian-openssl-flaw%3A-what-does-it-mean-tor-clients%3F

Part of dealing with the flaw meant doing some quick design work so we could let new Tor users be safe without making it so old Tor users were cut off from the network:
https://www.torproject.org/svn/trunk/doc/spec/proposals/136-legacy-keys.txt

Sometime in late June or early July we will disable this workaround, meaning all the 0.2.0.x users who haven't upgraded yet will be cut off.

C.2.5 The Contractor shall design and develop revisions to the Tor network protocols to hide the network signature of Tor traffic so it is difficult for government-sponsored Internet censors to identify Tor traffic and trivially block it.

As far as we know, nobody's put any effort into blocking our current protocol as it stands, since it no longer says "TOR" in the TLS certificates or "/tor/" in the directory fetch requests. We have some further steps we plan to do, and we hope to get a first cut at these deployed in June.

C.2.10 The Contractor shall promote active growth of the Tor server network and advocacy of Tor products to increase the performance, stability, and usability of Tor.

Roger collected another set of GeoIP-based user stats on May 30, showing that the overall user base has held roughly steady for the past month. We noticed a slight jump in users from Russia and Turkey; it's hard to know if that's just a quirk of the data though. Nick has been working on a design proposal for how to get a more accurate and comprehensive survey of Tor users, which we hope to try out in June.

Kevin Bauer and Damon McCoy from the University of Colorado have actual live data from the Tor network, that they collected from their exit relay while working on their research papers about Tor. We are starting to work with them to figure out we should do with it. Eventually the answer should be to release it, but how much should we anonymize it first, and can we release it alongside a set of guidelines for how to safely collect data like this in the first place?

Roger and Steven went to France in mid May to meet with Internews. We showed them the UPnP progress, including a simple demo you can watch:
http://freehaven.net/~arma/upnp-movie.avi
We also talked to them about the DRL proposal, and tried to convince them that their side of the funding should include writing docs and guides and tutorials for how to use various applications safely and securely with Tor. (That has been on our todo list for years, and I am coming to realize that we are not the right people to do it.) Also we need to make sure that we are more closely integrated with the trainings they do, so we can design our tools to be more directly useful.

Nick Mathewson, Andrew Lewman, and Wendy Seltzer attended the Computers, Freedom, and Privacy (CFP) conference in New Haven. They handed out 45 flyers and roughly 200 Tor stickers. Andrew gave a 5-minute talk about Tor. He's invited to give more talks about Tor to other orgs this summer, such as the United Nations, NNEDV, and Freedom House. The Charlotte Law School is also interested, but is probably more interested in Wendy going to talk to them than Andrew. The Committee to Protect Journalists, and the World Press Freedom Committee are also interested in Tor, but wanted to "vet the technology" before talking more. Their vetting process involves talking to Rebecca MacKinnon, and possibly Roger, at the Global Voices Summit.
https://blog.torproject.org/events/andrew-and-nick-cfp2008

Steven Murdoch, Paul Syverson, Jacob Appelbaum, and Ian Goldberg attended the IEEE Security and Privacy conference in Oakland, which is one of the top three annual academic conferences on computer security and privacy.
https://blog.torproject.org/events/steven%2C-paul%2C-ian%2C-others-oakland

We are preparing for the Tor gathering at the Privacy Enhancing Technologies Symposium in Leuven in July. This is looking like it will be the largest physical gathering of Tor developers ever

-- main developers attending include Roger Dingledine, Nick Mathewson, Jacob Appelbaum, Mike Perry, Matt Edman, Steven Murdoch, and Karsten Loesing; Tor researchers include Paul Syverson and Ian Goldberg; and we'll have 5 of our 7 Google Summer of Code students there as well.
https://blog.torproject.org/events/roger%2C-nick%2C-steven%2C-matt%2C-karsten%2C-paul%2C-jacob-pets
http://petsymposium.org/2008/program.php

Roger finished reviewing and crafting the "HotPETS" program, which is the third of the three days at the PETS conference, where new and interesting papers will be presented and then we'll facilitate discussion among the speakers and audience. Two of the eight hotpets papers are being presented by our Google Summer of Code students on their Tor work.

We reviewed a lot of anonymity-related academic papers this month. Roger reviewed a journal article submission that presented an improved attack on Tor's anonymity, and gave them tips for how to improve both the attack and analysis; hopefully somebody will work on how to improve the defense next! Nick, Steven, and Matt all started reviewing a lot of anonymity and privacy papers for the ACM CCS conference that will be held in DC in early November.

There's a talk at Defcon from some folks we know at Denver University promising a new attack on Tor. We're talking to them to make sure they'll tell us enough details that we can (try to) fix it before their talk. There's also a research group at Columbia University who wrote a paper attacking Tor; their paper is getting a lot of publicity because nobody knows anything more than the title. We've gotten a copy of the paper and are evaluating it.

The research group we've been working with at SRI has switched from aggregating intrusion detection system alert messages (for which they needed anonymous communication to aggregate them without anybody knowing where they came from) to studying malware. I'm in the process of explaining to them how Tor can be useful for this new situation too: if you crawl the web looking for malware, then bad sites will recognize the address you crawl from and opt to look innocent whenever you query them. Hopefully this new use will give us a story we can tell people about yet another variation of Tor user.

TorDNS Bulk Exitlist: We started working with [redacted] **a Wikipedia volunteer developer who wants to make Wikipedia handle Tor better. His plan is to let people edit through Tor if they've logged in first, and have a manual step for creating a new account (via email) if they don't already have an account. This is a plausible plan, but of course the hard part is still in gathering enough consensus in Wikipedia-land. I asked Jake to make a few tools on our end so it's easier for Wikipedia to figure out which IP addresses they should treat specially: we should export the list of Tor exit relay IP addresses and let them query it privately, rather than having them touch our TorDNSEL DNS server each time. This approach will also be more suited for IRC networks like OFTC too. The bulk exitlist is now up and running:**
https://check.torproject.org/cgi-bin/TorBulkExitList.py?ip=208.201.224.11

We worked briefly with Colin Maclay of the Berkman Center to try to hook him up with people who can testify to US Congress about what Cisco is actually up to in China. He didn't give me much warning though before the hearing, so it didn't go very far.

We resumed talking a bit to Joan Feigenbaum and Aaron Johnson at Yale about how to safely study the traffic on the Tor network. See also
http://www.imconf.net/imc-2007/papers/imc152.pdf
which is a paper written by some researchers plus a law professor on why it may be ok to sniff (among others) Tor traffic for research purposes.

C.2.11 *The Contractor shall improve the ease of use of Tor for end users by continuing research and development on one or both of the following products: (1) all-in-one software bundle containing Tor and supporting applications, as well as an easy-to-use installer for Microsoft Windows operating systems, as well as option to install and run from a Universal Serial Bus (USB) flash device; (2) bootable CD-ROM image ("LiveCD") which contains a minimal operating system, Tor, and supporting applications. Both would have all appropriate applications pre-configured to use Tor out of the box with only minimal additional configuration required by the end user. If Contractor determines it is not feasible to develop both products, Contractor will provide detailed written technical analysis and explanation to the AR/CO. The Contractor shall make an initial public release of at least one implementation of one of these products during the term of this contract.*

The upcoming TBB release in June will include optional instant messaging support via Pidgin + Off-The-Record Messaging; replace the startup batch script with an actual application (named RelativeLink), so TBB now has a helpful Tor icon rather than an ugly batch file icon; and optionally support using WinRAR to produce a self-extracting split bundle.

We now have a more thorough set of TBB build instructions:
https://svn.torproject.org/svn/torbrowser/trunk/build-scripts/INSTALL

We also documented the build and deploy process for a new TBB version:
https://svn.torproject.org/svn/torbrowser/trunk/build-scripts/DEPLOYMENT

C.2.12 *The Contractor shall continue to develop and implement improvements to the bridge relay and bridge directory authority mechanisms to improve the usability, performance and reliability of the Tor network by users in countries with government-imposed Internet censorship.*

We started exporting the "cached-extrainfo" file from our bridge authority to the BridgeDB server, so once we integrate a GeoIP database into bridge relays and they start reporting which countries their users are coming from, we can archive and analyze these reports. The first such reports started coming in in June; more information forthcoming.

C.2.13 *The Contractor shall research and document additional options for the scalability of the Tor* network *beyond 2 million concurrent users, including analysis of splitting the network into multiple segments, switching to datagram-based protocols, and improving the load balancing within the network.*

We continued work on a patch for OpenSSL that will make it keep less buffer space around. Currently fast Tor relays use (waste) as much as 100M of memory in OpenSSL's buffers. This patch was accepted and included in the main OpenSSL tree in June:
http://marc.info/?l=openssl-cvs&m=121246471627426&w=2

We finished integrating a UPnP library into Vidalia. This feature allows users who want to set up a Tor relay but don't want to muck with manual port forwarding on their router/firewall to just click a button and have Vidalia interact with their router/firewall automatically. This approach won't work in all cases, but it should work in at least some. The upcoming Vidalia 0.1.4 (scheduled for June) release has folded the UPnP library and GUI changes into the main Vidalia tree, along with a "test" button to try speaking UPnP at the local router and tell the user whether it worked; these features will be available by default in the 0.2.0.x stable release.

Steven Murdoch and Robert Watson finished the final version of their PETS 2008 paper called "Metrics for Security and Performance in Low-Latency Anonymity Systems."
http://www.cl.cam.ac.uk/~sjm217/papers/pets08metrics.pdf

Ian Goldberg's grad student is starting to work on his UDP-Tor design, with the new funding we've provided. We'll see where that goes.

C.2.14 The *Contractor shall continue research into the option of providing incentives for Tor users to run Tor relay servers. If further research indicates that this should be pursued, the Contractor shall develop a project plan and timeline for this work. If further research indicates this option should be abandoned, the Contractor shall document and explain in writing the reasoning behind this decision.*

Roger had a meeting with the other authors on the incentives paper, and we decided that the best way to move forward with publication is to open the paper up to a wider group of people for design and security review. Then we should focus on actually integrating some of the ideas into Tor, so we can have a section at the end of the paper that describes how the actual deployment went.

This path will take a lot more work before we see a publication (realistically I don't expect much of the design to go in until Tor 0.2.2.x at this rate), but it's probably the right way forward.

C.2.15 *The Contractor shall develop a more reliable download mechanism for the Tor browser bundle for users on slow and/or unreliable network connections, by means of a split download of multiple smaller files, implementation of a lightweight download manager, reduction in the software bundle file size, or other method as chosen by the Contractor.*

We gave up on finding a free software zip splitter on Windows, and decided to use WinRAR for now to create split files. Our experimental set of split files for Tor Browser Bundle 1.0.3 in Farsi is here:
https://www.torproject.org/torbrowser/dist/tor-browser-1.0.3_fa-IR-split/

We've also automated the process of building split files:
https://svn.torproject.org/svn/torbrowser/trunk/build-scripts/Makefile

Clear user-oriented instructions will come next.

Sometime in the second half of 2008 we will tackle this problem at a more fundamental level: part of our auto-update plan is to ship a very simple http client that can bootstrap the various Tor

components, fetch new versions of them, check the appropriate crypto signatures, etc. Once we have that http client working, we can use it to bootstrap the Tor Browser Bundle too, and we won't have to do this klunky split download approach.

C.2.16 The *Contractor shall test the Tor browser bundle on multiple computer systems and analyze these systems afterwards for any changes to the system that may have been made inadvertently by use of the Tor browser bundle. The Contractor shall document any such changes found and develop a plan to reduce the footprint of Tor browser bundle use.*

No work on this item yet. We're planning to get to it in June.

C.2.17 *The Contractor shall develop or adapt existing open source software to implement a web-based portal to manage the translations of text into multiple languages for the user interface text of* software *of Torbutton and Vidalia and other software that may in the future be included in the Tor browser bundle. The web site must allow non-technical users the ability to contribute translations by providing text to be translated in English, as well as any needed context on the use of the text, and allowing users to enter the translation into their language from their web browser.*

We spent May hunting for a better online translation option, since Launchpad (intended to be used for Vidalia translation) has an ugly interface and can't handle our file formats well, and Babelzilla (intended to be used for Torbutton translation) artificially limited the number of concurrent translators we could have.

In early June we hit upon Pootle, which is a translation server that we host, as opposed to a shared web service that other organizations host. We've set up a test server at http://translation.torproject.org/ and imported strings for Vidalia, Torbutton, and Torcheck. We hope to have a lot more to show here in June.

DO NOT REDISTRIBUTE

Tor modifications for China

Likely phases of attack

We need to guard against different types of attacks that the Chinese censors may mount against any anti-censorship network. However, we expect that the opposition from the Chinese censors will pass through a number of distinct phases (note that these phases do not correspond to the deployment phases in Roger's e-mail):

1. **The Chinese censors do nothing.** (This is the current state. The Chinese censors could shut down the entire Tor network by blocking people's connections to the directory servers, but so far, they haven't.) During this phase, new users can bootstrap by connecting to one of the main directory servers (or directory servers that are part of a China-only network that we create), and can be informed of the locations of nodes run by volunteers.
2. **The Chinese censors download the Tor client, see what IP addresses it initially connects to (the China-only directory servers), and block those.** At this point, our failover strategy is that:
    - (a) existing Tor clients in China should maintain connections with the clients in the "free world" that they have already learned about;
    - (b) new Tor clients should be distributed such that new bootstrap directory servers are periodically "rotated" into the downloads: For example, everybody who downloads the build on Monday gets a build that will bootstrap by connecting to machine A, everybody who downloads the build on Tuesday gets a build that will bootstrap by connecting to machine B, and so on. This is better than having a single build that randomly connects to one of the addresses on the list, since then a determined user could reverse engineer the software (or simply install it over and over on different operating system images) to find all the IP addresses that the build connects to. If we rotate a new "bootstrap" point into the build every day, then an attacker would only discover this by downloading different versions on successive days.
    - (c) **OPTIONAL:** those "free world" clients that users in China have already learned about, should not be given out to new users. (Otherwise, the Chinese censors could join the system as "new users", see what nodes their clients are programmed to connect to, and block those.) This is an optional step, depending on whether we have enough "spare nodes" in the system volunteering to let their connections by used by people in China.
3. **The Chinese censors make a concerted effort to block use of Tor, by downloading the client every day to see what IP addresses it is programmed to connect to, and blocking those IP addresses immediately.** After this happens, the only option for new users will be to "bootstrap" by asking a friend outside China to set up a node that they can connect to, have their friend send them the node location, and then manually entering that node location in their Tor client.
4. **NOT INCLUDED:** We do not take into account attacks by the Chinese censors that would require them to re-design their censorship architecture -- for example, instead of the Great Firewall, mandating that all Internet traffic in the country go through HTTP proxy servers. HTTP proxies can be instructed to block all https sites for which the SSL certificate is not signed by a well-known signing authority; this would effectively block all Tor connections. We don't believe this stage will be reached in the foreseeable future, however, so we do not take this into account.

Adapting Tor to this strategy

To use Tor for this strategy, Tor will have to behave differently from its current design in several respects. However, many of the features in the existing Tor design were based on a desire for very strong anonymity; in the case of providing the service to Chinese users, it may be acceptable to provide a weaker level of anonymity in exchange for greater ability to circumvent the Great Firewall.

- In the first phase above, when each new client connects to a directory server to find the location of a router, we

should ensure that the directory server only gives the client the minimum required number of routers needed to maintain a stable long-term connection. The "minimum" required is determined by: the number of routers needed so that if the user disconnects and then re-connects later, it is likely that at least one of the routers will still be online at the same location, so that the user can re-connect to that node and thereby learn the new locations of all the other nodes, if they've moved.

The major **open question** called out in Roger's e-mail is: How should we determine which onion routers to give to each new node? If a different list of nodes is given out with each new request, then this means that the censors can simply make as many requests as possible, until they have learned all or most of the ORs that are available for distribution.

- In the phases outlined above, especially the later phases where newly downloaded clients are set to "bootstrap" from a directory server that may have just been set up by a new volunteer, we would have to conscript so many new volunteer nodes as "directory servers" that it would be impossible to give them all the high level of trust that is accorded to directory servers in the existing model.

  Instead, we can safely assume that *most* new volunteer directory nodes will be trustworthy, if they are hosted at IP addresses in the "free world", and all we can do is ensure that an untrustworthy directory node cannot do too much damage.

- In the existing Tor network, a Tor client always attempts to make contact with its known directory servers when it comes online. For a Tor client to work in China, it should have a failover mode so that if it cannot contact its directory server, it can maintain its connection with the "free-world" nodes that it already knows about, and use its connection to those nodes to fetch banned content.

- By separating the roles of directory servers and onion routers, we can divide volunteers into two groups depending on what is more suited for them:

  1. Users who are online at a fixed IP address but cannot or do not want to donate large amounts of bandwidth, can sign up to be directory servers. This way, their IP address can be incorporated into the latest Tor install, and by the time the user obtains the installer and runs it on their computer, hopefully the volunteer's computer will still be online at the same address. Once the user has bootstrapped by connecting to that directory, the directory can refer them to a different node volunteering as an onion router; *that* node will be the node that donates the most bandwidth.

  2. Users whose machines are not online all the time or whose IP addresses change frequently, but who are willing to donate bandwidth to support Tor, can volunteer as onion routers. At any given time, the locations of a small number of onion routers are communicated to a node acting as a directory server, and that directory server gives the locations of those onion routers to nodes in China that connect and ask for them. (We would not want a directory server to give out the location of too many onion routers, because otherwise a hostile client could request the locations of a large number of them, and block them all.)

  A user who has sufficient spare bandwidth *and* a stable long-term IP address can of course volunteer to be both a directory server and an onion router.

- We should ensure that Tor connections do not include any character sequences (in the initial TLS handshake, for example) that are always present in Tor connections but that almost never appear anywhere else. Otherwise, the Chinese censors could simply add those character sequences to the filtered-word list at the router, and all Tor connections would break.

The Tor Project
122 Scott Circle
Dedham, MA 02026 USA
https://torproject.org/

From: Roger Dingledine, Tor Project Leader
To: Kelly DeYoe, program officer, IBB
RE: contract BBGCON1807S6441
Date: December 10, 2007

This report documents progress in November 2007 on contract BBGCON1807S6441 between IBB and The Tor Project.

C.2.0 New package releases and related software.

Torbutton 1.1.10 (released Nov 6) fixes two more privacy leak avenues, adds a new logging system, and lets the user configure whether to start using Tor or Non-Tor after Firefox crash.

Tor 0.2.0.10-alpha (released Nov 10) adds a third v3 directory authority run by Mike Perry, adds most of Karsten Loesing's new hidden service descriptor format, fixes a bad crash bug and new bridge bugs introduced in 0.2.0.9-alpha, fixes many bugs with the v3 directory implementation, fixes some minor memory leaks in previous 0.2.0.x snapshots, and addresses many more minor issues.

Tor 0.2.0.11-alpha (released Nov 12) fixes some build problems with the previous snapshot. It also includes a more secure-by-default exit policy for relays, fixes an enormous memory leak for exit relays, and fixes another bug where servers were falling out of the directory list.

Tor 0.2.0.12-alpha (released Nov 16) fixes some more build problems as well as a few minor bugs.

Torbutton 1.1.11 (released Nov 16) fixes another privacy leak possibility, and prevents Tor cookies from being written to disk if the user wants them cleared.

Torbutton 1.1.12 (released Nov 26) fixes three more privacy leak possibilities.

C.2.1 The Contractor shall continue design, development and implementation of enhancements to the existing Tor software to increase its suitability as a tool for Internet users in countries with government-sponsored Internet censorship to circumvent censorship controls, based on

the existing research and documentation performed during the previous contract period (e.g. as described in the paper "Design of a blocking-resistant anonymity system").

Continuing enhancements have been made to the Tor website Chinese translation.

From the Tor 0.2.0.10-alpha ChangeLog:
"Make bridge users work again -- the move to v3 directories in 0.2.0.9-alpha had introduced a number of bugs that made bridges no longer work for clients."

C.2.2 *The Contractor shall submit system architecture and technical design documentation for Tor enhancements specifically related to anti-censorship improvements in C.2.1 to the Authorized Representative of the Contracting Officer (AR/CO) for review and approval before development and implementation. Significant changes to the design that are discovered during development must be documented and reviewed by the AR/CO as soon as the Contractor becomes aware of the need for these revisions.*

We completed the documentation and specification detailing the current design and deployment status of bridges, bridge users, and bridge authorities.
https://www.torproject.org/svn/trunk/doc/spec/proposals/125-bridges.txt

We continued work on the modified-TLS handshake, based on the new insight from one of our contributors that we should be looking at "TLS renegotiation":
http://archives.seul.org/or/dev/Nov-2007/msg00008.html

We have an initial plan and design document for how to get GeoIP data and publish usage summaries from ordinary Tor relays and from bridge relays:
https://www.torproject.org/svn/trunk/doc/spec/proposals/126-geoip-reporting.txt

C.2.3 *The Contractor shall develop and implement the bridge relay mechanism, as designed during the previous contract period, to allow individual Tor users to easily reconfigure their Tor client to automatically relay traffic from users in countries with government-imposed Internet censorship so as to circumvent that censorship.*

From the Tor 0.2.0.10-alpha ChangeLog (a minor change):
"Bridges now use begin_dir to publish their server descriptor to the bridge authority, even when they haven't set TunnelDirConns."

C.2.4 *The Contractor shall develop and implement the bridge directory authority mechanism, as designed during the previous contract period, to allow Tor clients configured as bridge relays (as described in C.2.3) to communicate their existence to the bridge directory authority, and to provide a subset of addresses of available bridge relays to Tor users in countries with government-imposed Internet censorship so that they may access the Tor network.*

No changes.

C.2.5 *The Contractor shall design and develop revisions to the Tor network protocols to hide the*

network signature of Tor traffic so it is difficult for government-sponsored Internet censors to identify Tor traffic and trivially block it.

We started to deploy the new design for the normalized TLS handshake. This development and deployment will be continuing through December and probably into 2008 also.

C.2.6 *The Contractor shall design enhancements to Tor's cell-based protocol to improve performance on substandard network connections including those with low bandwidth and/or high latency and/or high packet loss.*

We set up the Tor relay named "ides" (run by Mike Perry) as the third v3 directory authority.

C.2.7 *The Contractor shall continue development of enhancements to improve the scalability of the Tor network toward the goal of supporting 2 million or more concurrent end users. This requirement is only a goal for system scalability and is not a requirement on number of actual concurrent users of the Tor network.*

From the 0.2.0.10-alpha ChangeLog:
"Directory authorities use a new formula for selecting which nodes to advertise as Guards: they must be in the top 7/8 in terms of how long we have known about them, and above the median of those nodes in terms of weighted fractional uptime."
"Raise the default BandwidthRate/BandwidthBurst to 5MB/10MB, to accommodate the growing number of servers that use the default and are reaching it."

From the 0.2.0.11-alpha ChangeLog:
"Exit policies now reject connections that are addressed to a relay's public (external) IP address too, unless ExitPolicyRejectPrivate is turned off. We do this because too many relays are running nearby to services that trust them based on network address. This change will allow more people to run relays comfortably, thus expanding the network."

C.2.8 *The Contractor shall work with IBB staff and other IBB contractors to identify tasks in support of this program that might be developed collaboratively with Contractor. Tasks in areas such as documentation, bug fixes, software testing, and any other areas involving specific knowledge of foreign government-sponsored Internet censorship may be especially appropriate for this purpose.*

No reports for this month.

C.2.9 *The Contractor shall communicate tasks identified for delegation to IBB in C.2.8 to the AR/CO and negotiate time frames for their completion. The Contractor shall monitor and coordinate work performed by IBB staff on delegated tasks and integrate it into Tor software releases as appropriate.*

No reports for this month.

C.2.10 *The Contractor shall promote active growth of the Tor server network and advocacy of Tor products to increase the performance, stability, and usability of Tor.*

Roger Dingledine met with IBB and Tom Hallewell in Washington DC, to discuss applicability of the bridge design for RFA's user base.

Roger also presented the overall Tor design to two groups at NSA. Getting a wider variety of organizations interested in Tor and aware of its security features will ultimately produce a more sustainable and more secure network.

Roger met with Jeff Blum, who is our volunteer web contributor. He has been working on revising the frontpage, the download page, and most recently a "who users Tor" page.

Mike Perry has set up a Tor blog for us at https://blog.torproject.org/. We'll start populating it with blog entries when we find a spare moment.

C.2.11 *The Contractor shall improve the ease of use of Tor for end users by continuing research and development on one or both of the following products: (1) all-in-one software bundle containing Tor and supporting applications, as well as an easy-to-use installer for Microsoft Windows operating systems, as well as option to install and run from a Universal Serial Bus (USB) flash device; (2) bootable CD-ROM image ("LiveCD") which contains a minimal operating system, Tor, and supporting applications. Both would have all appropriate applications pre-configured to use Tor out of the box with only minimal additional configuration required by the end user. If Contractor determines it is not feasible to develop both products, Contractor will provide detailed written technical analysis and explanation to the AR/CO. The Contractor shall make an initial public release of at least one implementation of one of these products during the term of this contract.*

The Incognito LiveCD project got a new maintainer this month: [redacted] **The next focus is going to be on paring down the set of default applications it comes with, so it's easier for ordinary users to figure out how to run the applications they want.**

Steven Murdoch started putting his full attention to the Tor USB image this month. We expect to land in December the first revision of a usable bundle that includes Tor, Polipo, Vidalia, and a modified Firefox.

The Tor Project
122 Scott Circle
Dedham, MA 02026 USA
https://torproject.org/

From: Roger Dingledine, Tor Project Leader
To: Kelly DeYoe, program officer, IBB
RE: contract BBGCON1807S6441
Date: November 10, 2007

This report documents progress in October 2007 on contract BBGCON1807S6441 between IBB and The Tor Project.

C.2.1 *The Contractor shall continue design, development and implementation of enhancements to the existing Tor software to increase its suitability as a tool for Internet users in countries with government-sponsored Internet censorship to circumvent censorship controls, based on the existing research and documentation performed during the previous contract period (e.g. as described in the paper "Design of a blocking-resistant anonymity system").*

Continuing enhancements have been made to the Tor website Chinese translation.

Tor 0.2.0.8-alpha (released Oct 12) fixes a crash bug that's been bothering us since February 2007, lets bridge authorities store a list of bridge descriptors they've seen, gets v3 directory voting closer to working, starts caching v3 directory consensus documents on directory mirrors, and fixes a variety of smaller issues including some minor memory leaks.

Torbutton 1.1.8 (released Oct 12) fixes six bugs that could allow private information leaks.

Tor 0.2.0.9-alpha (released Oct 24) switches clients to the new v3 directory system; allows servers to be listed in the network status even when they have the same nickname as a registered server; and fixes many other bugs including a big one that was causing some servers to disappear from the network status lists for a few hours each day.

Torbutton 1.1.9.1 (released Oct 24) blocks loading of direct clicks of plugin-handled content. Torbutton is now included in the development OS X bundles.

Vidalia 0.0.15 (released Oct 24) adds support for Tor's HTTP/HTTPS proxy and "fascist firewall" options. Vidalia now also supports Tor's bridge relay features for

those who are blocked from reaching the Tor network, or who want to set up their own bridge relay to help censored Tor users.

The upcoming Vidalia 0.0.16 release will include a translation to Arabic.

The new Tor bundles for Windows and OS X also include modified Privoxy config files to avoid some security problems that could allow a remote attacker to disable Privoxy's filtering mechanisms.

Tor 0.1.2.18 (released Oct 28) fixes many problems including crash bugs, problems with hidden service introduction that were causing huge delays, and a big bug that was causing some servers to disappear from the network status lists for a few hours each day.

C.2.2 *The Contractor shall submit system architecture and technical design documentation for Tor enhancements specifically related to anti-censorship improvements in C.2.1 to the Authorized Representative of the Contracting Officer (AR/CO) for review and approval before development and implementation. Significant changes to the design that are discovered during development must be documented and reviewed by the AR/CO as soon as the Contractor becomes aware of the need for these revisions.*

We started on a specification proposal detailing the current design and deployment status of bridges and bridge authorities. It's due to be completed in November.

We continued work on the modified-TLS handshake, also due to be completed in November.

C.2.3 *The Contractor shall develop and implement the bridge relay mechanism, as designed during the previous contract period, to allow individual Tor users to easily reconfigure their Tor client to automatically relay traffic from users in countries with government-imposed Internet censorship so as to circumvent that censorship.*

No major changes; bridge relays seem to be working for now.

We made a minor change in 0.2.0.8-alpha that improves the anonymity of bridge relay operators:
"Never report that we've used more bandwidth than we're willing to relay according to RelayBandwidthRate: it leaks how much non-relay traffic we're using. Resolves bug 516."

C.2.4 *The Contractor shall develop and implement the bridge directory authority mechanism, as designed during the previous contract period, to allow Tor clients configured as bridge relays (as described in C.2.3) to communicate their existence to the bridge directory authority, and to provide a subset of addresses of available bridge relays to Tor users in countries with government-imposed Internet censorship so that they may access the Tor network.*

From the Tor 0.2.0.8-alpha ChangeLog:

"Bridge authorities now write bridge descriptors to disk, meaning we can export them to other programs and begin distributing them to blocked users."

The next step (scheduled for November/December) is to have the current bridge authority export this file, and start distributing the bridge identifiers to blocked users.

C.2.5 *The Contractor shall design and develop revisions to the Tor network protocols to hide the network signature of Tor traffic so it is difficult for government-sponsored Internet censors to identify Tor traffic and trivially block it.*

Continued work on a draft strategy for making our TLS handshake look more normal. We'll be deploying this design in November.

C.2.6 *The Contractor shall design enhancements to Tor's cell-based protocol to improve performance on substandard network connections including those with low bandwidth and/or high latency and/or high packet loss.*

The Tor 0.2.0.9-alpha release implements the official switch to the "v3" directory voting protocol:
"Clients now download v3 consensus networkstatus documents instead of v2 networkstatus documents. Clients and caches now base their opinions about routers on these consensus documents. Clients only download router descriptors listed in the consensus."
See https://www.torproject.org/svn/trunk/doc/spec/dir-spec.txt for details on the new directory design.

C.2.7 *The Contractor shall continue development of enhancements to improve the scalability of the Tor network toward the goal of supporting 2 million or more concurrent end users. This requirement is only a goal for system scalability and is not a requirement on number of actual concurrent users of the Tor network.*

From the 0.2.0.8-alpha ChangeLog:
"Directory authorities track weighted fractional uptime as well as weighted mean-time-between failures. WFU is suitable for deciding whether a node is "usually up", while MTBF is suitable for deciding whether a node is "likely to stay up." We need both, because "usually up" is a good requirement for guards, while "likely to stay up" is a good requirement for long-lived connections."

From the 0.2.0.9-alpha ChangeLog:
"Authorities now list servers who have the same nickname as a different named server, but list them with a new flag, "Unnamed". Now we can list servers that happen to pick the same nickname as a server that registered two years ago and then disappeared. Partially implements proposal 122."

C.2.8 *The Contractor shall work with IBB staff and other IBB contractors to identify tasks in support of this program that might be developed collaboratively with Contractor. Tasks in*

areas such as documentation, bug fixes, software testing, and any other areas involving specific knowledge of foreign government-sponsored Internet censorship may be especially appropriate for this purpose.

No reports for this month.

C.2.9 *The Contractor shall communicate tasks identified for delegation to IBB in C.2.8 to the AR/CO and negotiate time frames for their completion. The Contractor shall monitor and coordinate work performed by IBB staff on delegated tasks and integrate it into Tor software releases as appropriate.*

No reports for this month.

C.2.10 *The Contractor shall promote active growth of the Tor server network and advocacy of Tor products to increase the performance, stability, and usability of Tor.*

Roger Dingledine gave a pair of invited lectures at the University of Manizales in Colombia. He also met with several IT specialists from Venezuela who brought news of current and upcoming Internet censorship in Venezuela.

Roger also attended and helped organize the Workshop on Privacy in the Electronic Society in Washington DC, where several good academic research papers on Tor were presented; and the ACM Conference of Computer and Communication Security, where several more good academic research papers on Tor were presented.

We also met with Aaron Swartz fromOpenLibrary.org, who introduced us to a group of Thai citizens who are working to circumvent government filtering in their country.

We made the official switch to the torproject.org domain. This move will let us put much more detailed documentation and guides on our website, since the pages will no longer need to be vetted by EFF folks first.

C. *The Contractor shall improve the ease of use of Tor for end users by continuing research and development on one or both of the following products: (1) all-in-one software bundle containing Tor and supporting applications, as well as an easy-to-use installer for Microsoft Windows operating systems, as well as option to install and run from a Universal Serial Bus (USB) flash device; (2) bootable CD-ROM image ("LiveCD") which contains a minimal operating system, Tor, and supporting applications. Both would have all appropriate applications pre-configured to use Tor out of the box with only minimal additional configuration required by the end user. If Contractor determines it is not feasible to develop both products, Contractor will provide detailed written technical analysis and explanation to the AR/CO. The Contractor shall make an initial public release of at least one implementation of one of these products during the term of this contract.*

The Incognito LiveCD project is switching to Polipo. They have worked out the right config options for running Polipo safely in the context of Tor. This recommended config

file will be useful for the general Tor bundles too.

Additional news:

- **October was Shava Nerad's last month as a Tor employee. She will continue to work with us on a volunteer basis for PR and other projects.**

The Tor Project
122 Scott Circle
Dedham, MA 02026 USA
https://torproject.org/

From: Roger Dingledine, Tor Project Leader
To: Kelly DeYoe, program officer, IBB
RE: contract BBGCON1807S6441
Date: October 10, 2007

This report documents progress in September 2007 on contract BBGCON1807S6441 between IBB and The Tor Project.

C.2.1 *The Contractor shall continue design, development and implementation of enhancements to the existing Tor software to increase its suitability as a tool for Internet users in countries with government-sponsored Internet censorship to circumvent censorship controls, based on the existing research and documentation performed during the previous contract period (e.g. as described in the paper "Design of a blocking-resistant anonymity system").*

Continuing enhancements have been made to the Tor website Chinese translation.

Tor 0.2.0.7-alpha (released Sep 21) makes bridges work again, makes bridge authorities work for the first time, fixes two huge performance flaws in

The Windows bundle also includes the new development Torbutton version 1.1.7 (released Sep 21), which clears cookies and disables a lot of other dangerous web behavior. A lot more stability and usability work remains on this development branch of Torbutton.

We began investigating whether to replace Privoxy with Polipo in the default Windows and OS X bundles. Preliminary results are that Polipo offers no actual performance advantages, but it offers some improvements in other respects. More research remains.

C.2.2 *The Contractor shall submit system architecture and technical design documentation for Tor enhancements specifically related to anti-censorship improvements in C.2.1 to the Authorized Representative of the Contracting Officer (AR/CO) for review and approval before development and implementation. Significant changes to the design that are discovered during development must be documented and reviewed by the AR/CO as soon as the Contractor becomes aware of the need for these revisions.*

We now have a graphical draft of a bridge interface (along with other firewall and proxy settings) in Vidalia:
http://freehaven.net/~arma/vidalia-bridge-screenshot.png

In October we plan to attach the interface to the actual code so clicking the buttons actually produces results.

C.2.3 *The Contractor shall develop and implement the bridge relay mechanism, as designed during the previous contract period, to allow individual Tor users to easily reconfigure their Tor client to automatically relay traffic from users in countries with government-imposed Internet censorship so as to circumvent that censorship.*

We fixed a major bug that was causing bridges running recent alpha versions of Tor to not function properly:

From the 0.2.0.7-alpha ChangeLog:
"Fix a bug that made servers send a "404 Not found" in response to attempts to fetch their server descriptor. This caused Tor servers to take many minutes to establish reachability for their DirPort, and it totally crippled bridges. Bugfix on 0.2.0.5-alpha."

C.2.4 *The Contractor shall develop and implement the bridge directory authority mechanism, as designed during the previous contract period, to allow Tor clients configured as bridge relays (as described in C.2.3) to communicate their existence to the bridge directory authority, and to provide a subset of addresses of available bridge relays to Tor users in countries with government-imposed Internet censorship so that they may access the Tor network.*

We implemented another step in making bridge authorities actually useful. Now Tor clients can configure themselves to bootstrap by getting bridge descriptor updates only from the bridge authority:

From the 0.2.0.7-alpha ChangeLog:
"Make "UpdateBridgesFromAuthority" torrc option work: when bridge users configure that and specify a bridge with an identity fingerprint, now they will lookup the bridge descriptor at the default bridge authority via a one-hop tunnel, but once circuits are established they will switch to a three-hop tunnel for later connections to the bridge authority. Bugfix in 0.2.0.3-alpha."

The next step (scheduled for October) is to let bridge authorities write out a list of descriptors that are annotated by "purpose", so we can distinguish bridge descriptors from ordinary Tor server descriptors. Then we can start giving out these bridge descriptors using the variety of distribution methods described in the blocking.pdf document.

C.2.5 *The Contractor shall design and develop revisions to the Tor network protocols to hide the network signature of Tor traffic so it is difficult for government-sponsored Internet censors to identify Tor traffic and trivially block it.*

Began work on a draft strategy for making our TLS handshake look more normal. Early draft at: http://www.cl.cam.ac.uk/~sjm217/volatile/guest/xxx-tls-normalization.txt

C.2.6 *The Contractor shall design enhancements to Tor's cell-based protocol to improve performance on substandard network connections including those with low bandwidth and/or high latency and/or high packet loss.*

We continued to make progress on the "v3" directory voting protocol. The Tor 0.2.0.7-alpha release sets up moria1 and tor26 as the first v3 directory authorities. See https://tor.eff.org/svn/trunk/doc/spec/dir-spec.txt for details on the new directory design.

We also completed the last step in separating the "bandwidth usage reporting" lines out of the normal router descriptor format. All Tor servers running 0.2.0.7-alpha and later will omit these bandwidth lines and only publish them in a separate "extra info" descriptor. This will shrink ordinary router descriptors by as much as 60%. https://tor.eff.org/svn/trunk/doc/spec/proposals/104-short-descriptors.txt

C.2.7 *The Contractor shall continue development of enhancements to improve the scalability of the Tor network toward the goal of supporting 2 million or more concurrent end users. This requirement is only a goal for system scalability and is not a requirement on number of actual concurrent users of the Tor network.*

Informal GeoIP-based user statistics show that China, Germany, and the United States are our top three user bases, with roughly 20% of the Tor users each. Very rough user counts show that the overall Tor user base how grown from last year's estimate of 100K-200K users to perhaps 250K users.

C.2.8 *The Contractor shall work with IBB staff and other IBB contractors to identify tasks in support of this program that might be developed collaboratively with Contractor. Tasks in areas such as documentation, bug fixes, software testing, and any other areas involving specific knowledge of foreign government-sponsored Internet censorship may be especially appropriate for this purpose.*

No reports for this month.

C.2.9 *The Contractor shall communicate tasks identified for delegation to IBB in C.2.8 to the AR/CO and negotiate time frames for their completion. The Contractor shall monitor and coordinate work performed by IBB staff on delegated tasks and integrate it into Tor software releases as appropriate.*

No reports for this month.

C.2.10 *The Contractor shall promote active growth of the Tor server network and advocacy of Tor products to increase the performance, stability, and usability of Tor.*

Roger Dingledine presented on a panel at the MIT Technology Review conference (Sept 27). We also met a variety of other interested attendees, including a business person from Intel, the CTO of Secure Computing (the company that sells Smartfilter), and a fellow who works with CIA and State Dept and is working on human rights.

We also started moving closer to switching to the torproject.org domain. This move will let us put much more detailed documentation and guides on our website, since the pages will no longer need to be vetted by EFF folks first.

C. *The Contractor shall improve the ease of use of Tor for end users by continuing research and development on one or both of the following products: (1) all-in-one software bundle containing Tor and supporting applications, as well as an easy-to-use installer for Microsoft Windows operating systems, as well as option to install and run from a Universal Serial Bus (USB) flash device; (2) bootable CD-ROM image ("LiveCD") which contains a minimal operating system, Tor, and supporting applications. Both would have all appropriate applications pre-configured to use Tor out of the box with only minimal additional configuration required by the end user. If Contractor determines it is not feasible to develop both products, Contractor will provide detailed written technical analysis and explanation to the AR/CO. The Contractor shall make an initial public release of at least one implementation of one of these products during the term of this contract.*

The new Incognito 20070824.1 LiveCD (released on Sep 5) upgrades to Tor 0.1.2.17, adds a bunch of new software plus configurations, and fixes a variety of bugs.

The new Incognito 20070824.2 LiveCD (released on Sep 29) allows Incognito to be run in a virtual PC via qemu (qemu is like vmware but free), so ordinary Windows users can launch Incognito in a virtual window. It also includes the new version of Vidalia, a variety of upgrades, and is more tolerant of old and unusual hardware.

Additional news:

- **Steven Murdoch joined us starting this month. He will be working on making Tor's TLS fingerprint not as obvious, on considering whether to switch from the Privoxy HTTP proxy to the Polipo HTTP proxy, and on LiveCD/USB recommended configurations.**

- **Also, a storm worm email spoofed that it would protect you by downloading Tor. Some security experts mused that this implies a level of trust and knowledge of Tor that it has become a "trusted brand" -- which unfortunately can lead to exploitation of trust. We used the event to do consciousness raising.**

The Tor Project
122 Scott Circle
Dedham, MA 02026 USA
https://torproject.org/

From: Roger Dingledine, Tor Project Leader
To: Kelly DeYoe, program officer, IBB
RE: contract BBGCON1807S6441
Date: October 10, 2007

This report documents progress in September 2007 on contract BBGCON1807S6441 between IBB and The Tor Project.

C.2.1 *The Contractor shall continue design, development and implementation of enhancements to the existing Tor software to increase its suitability as a tool for Internet users in countries with government-sponsored Internet censorship to circumvent censorship controls, based on the existing research and documentation performed during the previous contract period (e.g. as described in the paper "Design of a blocking-resistant anonymity system").*

Continuing enhancements have been made to the Tor website Chinese translation.

Tor 0.2.0.7-alpha (released Sep 21) makes bridges work again, makes bridge authorities work for the first time, fixes two huge performance flaws in hidden services, and fixes a variety of minor issues.

The Windows bundle also includes the new development Torbutton version 1.1.7 (released Sep 21), which clears cookies and disables a lot of other dangerous web behavior. A lot more stability and usability work remains on this development branch of Torbutton.

We began investigating whether to replace Privoxy with Polipo in the default Windows and OS X bundles. Preliminary results are that Polipo offers no actual performance advantages, but it offers some improvements in other respects. More research remains.

C.2.2 *The Contractor shall submit system architecture and technical design documentation for Tor enhancements specifically related to anti-censorship improvements in C.2.1 to the Authorized Representative of the Contracting Officer (AR/CO) for review and approval before development and implementation. Significant changes to the design that are discovered during development must be documented and reviewed by the AR/CO as soon as the Contractor becomes aware of the need for these revisions.*

We now have a graphical draft of a bridge interface (along with other firewall and proxy settings) in Vidalia:
http://freehaven.net/~arma/vidalia-bridge-screenshot.png

In October we plan to attach the interface to the actual code so clicking the buttons actually produces results.

C.2.3 *The Contractor shall develop and implement the bridge relay mechanism, as designed during the previous contract period, to allow individual Tor users to easily reconfigure their Tor client to automatically relay traffic from users in countries with government-imposed Internet censorship so as to circumvent that censorship.*

We fixed a major bug that was causing bridges running recent alpha versions of Tor to not function properly:

From the 0.2.0.7-alpha ChangeLog:
"Fix a bug that made servers send a "404 Not found" in response to attempts to fetch their server descriptor. This caused Tor servers to take many minutes to establish reachability for their DirPort, and it totally crippled bridges. Bugfix on 0.2.0.5-alpha."

C.2.4 *The Contractor shall develop and implement the bridge directory authority mechanism, as designed during the previous contract period, to allow Tor clients configured as bridge relays (as described in C.2.3) to communicate their existence to the bridge directory authority, and to provide a subset of addresses of available bridge relays to Tor users in countries with government-imposed Internet censorship so that they may access the Tor network.*

We implemented another step in making bridge authorities actually useful. Now Tor clients can configure themselves to bootstrap by getting bridge descriptor updates only from the bridge authority:

From the 0.2.0.7-alpha ChangeLog:
"Make "UpdateBridgesFromAuthority" torrc option work: when bridge users configure that and specify a bridge with an identity fingerprint, now they will lookup the bridge descriptor at the default bridge authority via a one-hop tunnel, but once circuits are established they will switch to a three-hop tunnel for later connections to the bridge authority. Bugfix in 0.2.0.3-alpha."

The next step (scheduled for October) is to let bridge authorities write out a list of descriptors that are annotated by "purpose", so we can distinguish bridge descriptors from ordinary Tor server descriptors. Then we can start giving out these bridge descriptors using the variety of distribution methods described in the blocking.pdf document.

C.2.5 *The Contractor shall design and develop revisions to the Tor network protocols to hide the network signature of Tor traffic so it is difficult for government-sponsored Internet censors to*

identify Tor traffic and trivially block it.

Began work on a draft strategy for making our TLS handshake look more normal. Early draft at: http://www.cl.cam.ac.uk/~sjm217/volatile/guest/xxx-tls-normalization.txt

C.2.6 *The Contractor shall design enhancements to Tor's cell-based protocol to improve performance on substandard network connections including those with low bandwidth and/or high latency and/or high packet loss.*

We continued to make progress on the "v3" directory voting protocol. The Tor 0.2.0.7-alpha release sets up moria1 and tor26 as the first v3 directory authorities. See https://tor.eff.org/svn/trunk/doc/spec/dir-spec.txt for details on the new directory design.

We also completed the last step in separating the "bandwidth usage reporting" lines out of the normal router descriptor format. All Tor servers running 0.2.0.7-alpha and later will omit these bandwidth lines and only publish them in a separate "extra info" descriptor. This will shrink ordinary router descriptors by as much as 60%. https://tor.eff.org/svn/trunk/doc/spec/proposals/104-short-descriptors.txt

C.2.7 *The Contractor shall continue development of enhancements to improve the scalability of the Tor network toward the goal of supporting 2 million or more concurrent end users. This requirement is only a goal for system scalability and is not a requirement on number of actual concurrent users of the Tor network.*

Informal GeoIP-based user statistics show that China, Germany, and the United States are our top three user bases, with roughly 20% of the Tor users each. Very rough user counts show that the overall Tor user base how grown from last year's estimate of 100K-200K users to perhaps 250K users.

C.2.8 *The Contractor shall work with IBB staff and other IBB contractors to identify tasks in support of this program that might be developed collaboratively with Contractor. Tasks in areas such as documentation, bug fixes, software testing, and any other areas involving specific knowledge of foreign government-sponsored Internet censorship may be especially appropriate for this purpose.*

No reports for this month.

C.2.9 *The Contractor shall communicate tasks identified for delegation to IBB in C.2.8 to the AR/CO and negotiate time frames for their completion. The Contractor shall monitor and coordinate work performed by IBB staff on delegated tasks and integrate it into Tor software releases as appropriate.*

No reports for this month.

C.2.10 *The Contractor shall promote active growth of the Tor server network and advocacy of Tor*

products to increase the performance, stability, and usability of Tor.

Roger Dingledine presented on a panel at the MIT Technology Review conference (Sept 27). We also met a variety of other interested attendees, including a business person from Intel, the CTO of Secure Computing (the company that sells Smartfilter), and a fellow who works with CIA and State Dept and is working on human rights.

We also started moving closer to switching to the torproject.org domain. This move will let us put much more detailed documentation and guides on our website, since the pages will no longer need to be vetted by EFF folks first.

C. The Contractor shall improve the ease of use of Tor for end users by continuing research and development on one or both of the following products: (1) all-in-one software bundle containing Tor and supporting applications, as well as an easy-to-use installer for Microsoft Windows operating systems, as well as option to install and run from a Universal Serial Bus (USB) flash device; (2) bootable CD-ROM image ("LiveCD") which contains a minimal operating system, Tor, and supporting applications. Both would have all appropriate applications pre-configured to use Tor out of the box with only minimal additional configuration required by the end user. If Contractor determines it is not feasible to develop both products, Contractor will provide detailed written technical analysis and explanation to the AR/CO. The Contractor shall make an initial public release of at least one implementation of one of these products during the term of this contract.

The new Incognito 20070824.1 LiveCD (released on Sep 5) upgrades to Tor 0.1.2.17, adds a bunch of new software plus configurations, and fixes a variety of bugs.

The new Incognito 20070824.2 LiveCD (released on Sep 29) allows Incognito to be run in a virtual PC via qemu (qemu is like vmware but free), so ordinary Windows users can launch Incognito in a virtual window. It also includes the new version of Vidalia, a variety of upgrades, and is more tolerant of old and unusual hardware.

Additional news:

- **Steven Murdoch joined us starting this month. He will be working on making Tor's TLS fingerprint not as obvious, on considering whether to switch from the Privoxy HTTP proxy to the Polipo HTTP proxy, and on LiveCD/USB recommended configurations.**

- **Also, a storm worm email spoofed that it would protect you by downloading Tor. Some security experts mused that this implies a level of trust and knowledge of Tor that it has become a "trusted brand" -- which unfortunately can lead to exploitation of trust. We used the event to do consciousness raising.**

A Practical Congestion Attack on Tor Using Long Paths

Anonymous
Some Where
Email: someone@example.com

Abstract

In 2005, Murdoch and Danezis demonstrated the first practical congestion attack against a deployed anonymity network. They could identify which relays were on a target Tor user's path by building paths one at a time through every Tor relay and introducing congestion. However, the original attack was performed on only 13 Tor relays on the nascent and lightly loaded Tor network.

We show that the attack from their paper is no longer practical on today's 1500-relay heavily loaded Tor network. The attack doesn't scale because a) the attacker needs a tremendous amount of bandwidth to measure enough relays in the attack window, and b) there are too many false positives now that many other users are adding congestion at the same time as the attacks.

We then strengthen the original congestion attack by combining it with a novel bandwidth amplication attack based on a flaw in the Tor protocol that lets us build long circuits that loop back on themselves. We show that this new combination attack is practical by demonstrating a working attack on today's deployed Tor network. By coming up with a model to better understand Tor's routing behavior under congestion, we further provide a statistical analysis characterizing exactly how effective our attack is in each case. Finally, we designed a defense against our new attack and are working with the Tor developers to deploy the defense.

1. Introduction

This paper presents an attack which exploits a weakness in Tor's circuit construction protocol to implement an improved variant of Murdoch and Danezis's congestion attack [22], [23]. Tor [8] is an anonymizing peer-to-peer network that provides users with the ability to establish low-latency TCP tunnels, called circuits, through a network of relays provided by the peers in the network. In 2005, Murdoch and Danezis were able to determine the path messages take through the Tor network by causing congestion in the network and then observing the changes in the traffic patterns.

While Murdoch and Danezis's work pioneered the idea proposed in [1] of an adversary perturbing traffic patterns of a low-latency network to deanonymize its users, the original attack no longer works on the modern Tor network. Murdoch and Danezis's approach gives too many false-positives for a network the size of the current Tor network. In a network with thousands of routers, too many peers share similar latency characteristics and the amount of congestion that was detectable in 2005 is no longer significant; thus, the traffic of a single normal user does not leave an easily distinguishable signature in the significantly larger volume of data routed by today's Tor network.

Our attack addresses these weaknesses by combining JavaScript injection with a selective and asymmetric denial-of-service (DoS) attack to obtain specific information about the path selected by the victim. As a result, we are able to identify the entire path for a user of today's Tor network. We also provide an improved method for evaluating the statistical significance of the obtained data that is based on the actual Tor message scheduling algorithm. As a result, we are not only able to determine which peers are a part of the circuit with high probability, we can also quantify the extent to which the attack succeeds. This paper presents the attack and experimental results obtained from the actual Tor network, and proposes some non-trivial modifications to the current Tor protocol and implementation which would make the presented attack impractical.

Just as Murdoch and Danezis's work applied to other systems such as MorphMix [20] or Tarzan [32], our improved attack and the solution can also be generalized to other networks using onion routing. Also, in contrast to previously proposed solutions to congestion attacks [14], [18]–[20], [24], [27], [31], [32], the solution presented in this paper does not impact the performance of the anonymizing network.

2. Related Work

Chaum's mixes [3] are a common method for achieving anonymity. Multiple encrypted messages are sent to a mix from different sources and each is forwarded by the mix to its respective destination. Combinations of artificial delays, changes in message order, message batching, uniform message formats (after encryption), and chaining of multiple mixes are used to further mask the correspondence between input and output flows in various variations of the design [5]–[7], [13], [17], [21], [28], [29]. Onion routing [12] is essentially the process of using an initiator-selected chain of low-latency mixes for the transmission of encrypted streams of messages in such a way that each mix only knows the previous and the next mix in the chain, thus providing initiator-anonymity even if some of the mixes are controlled by the adversary.

2.1. Tor

Tor [8] is a distributed anonymizing network that uses onion routing to provide anonymity for its users. Most Tor users access the Tor network via a local proxy program such as Privoxy [16] to tunnel the HTTP requests of their browser through the Tor network. The goal is to make it difficult for web servers to ascertain the IP address of the browsing user. Tor provides anonymity by utilizing a large number of distributed volunteer-run relays (or routers). The Tor client software retrieves a list of participating relays, chooses some number of them at random, and creates a circuit (a chain of relays) through the network. The circuit setup involves establishing a session key with each router in the circuit, so that data sent can be encrypted in multiple layers that are peeled off as the data travels through the network. The client encrypts the data once for each relay, and then sends it to the first relay in the circuit; each relay successively peels off one encryption layer and forwards the traffic to the next link in the chain until it reaches the final node, the exit router of the circuit, which sends the traffic out to the destination on the Internet.

Data that passes through the Tor network is packaged into fixed sized cells, which are queued upon receipt for processing and forwarding. For each circuit that a Tor router is a part of, the router maintains a separate queue and processes these queues in a round-robin fashion. If a queue for a circuit is empty it is skipped. Other than using this fairness scheme, Tor does not introduce any latency when forwarding cells.

The Tor threat model differs from the usual model for anonymity schemes [8]. The traditional threat model is that of a global passive adversary: one that can observe all traffic on the network between any two links. In contrast, Tor assumes a non-global adversary which can only observe some subset of the connections and can control only a subset of Tor nodes. Well-known attack strategies such as blending attacks [30] require more powerful attackers than those permitted by Tor's attacker model. Tor's model is still valuable, as the resulting design achieves a level of anonymity that is sufficient for many users while providing reasonable performance. Unlike the aforementioned strategies, the adversary used in this paper operates within the limits set by Tor's attacker model. Specifically, our adversary is simply able to run a Tor exit node and access the Tor network with resources similar to those of a normal Tor user.

2.2. Attacks on Tor and other Mixes

Many different attacks on low latency mix networks and other anonymization schemes exist, and a fair amount of these are specifically aimed at the Tor network. These attacks can be broadly categorized into three categories: path selection attacks, passive attacks, and congestion attacks. Path selection attacks attempt to invalidate the assumption that selecting relays at random will usually result in a safe circuit. Passive attacks are those where the adversary in large part simply observes the network in order to reduce the anonymity of users. Active attacks are those where the adversary uses its resources to modify the behavior of the network; we'll focus here on a class of active attacks known as congestion or interference attacks.

2.2.1. Path Selection Attacks. Path selection is crucial for the security of Tor users; in order to retain anonymity, the initiator needs to choose a path such that the first and last relay in the circuit won't collude. By selecting relays at random during circuit creation, it could be assumed that the probability of finding at least one non-malicious relay would increase with longer paths. However, this reasoning ignores the possibility that malicious Tor routers might choose only to facilitate connections with other adversary-controlled relays and discard all other connections [2]; thus the initiator either constructs a fully-malicious circuit upon randomly selecting a malicious node, or fails that circuit and tries again. This type of attack suggests that longer circuits do not guarantee stronger anonymity.

A variant of this attack called "packet spinning" [27] attempts to force users to select malicious routers by

causing legitimate routers to time out. Here the attacker builds circular paths throughout the Tor network and transmits large amounts of data through those paths in order to keep legitimate relays busy. The attacker then runs another set of (malicious) servers which would eventually be selected by users because of the attacker-generated load on all legitimate mixes. The attack is successful if, as a result, the initiator chooses only malicious servers for its circuit, making deanonymization trivial.

2.2.2. Passive Attacks. Several passive attacks on mix systems were proposed by Back et al [1]. The first of these attacks is a "packet counting" attack, where a global passive adversary simply monitors the initiator's output to discover the number of packets sent to the first mix, then observes the first mix to watch for the same number of packets going to some other destination. In this way, a global passive adversary could correlate traffic to a specific user. As described by Levine et al [19], the main method of defeating such attacks is to pad the links between mixes with cover traffic. This defense is costly and may not solve the problem when faced with an active attacker with significant resources; an adversary with enough bandwidth can deal with cover traffic by using up as much of the allotted traffic between two nodes as possible with adversary-generated traffic [4]. As a result, no remaining bandwidth is available for legitimate cover traffic and the adversary can still deduce the amount of legitimate traffic that is being processed by the mix. This attack (as well as others described in this context) requires the adversary to have significant bandwidth. It should be noted that in contrast, the adversary described by our attack requires only the resources of an average mix operator.

Low-latency anonymity systems are also vulnerable to more active timing analysis variations. The attack presented in [19] is based on an adversary's ability to track specific data through the network by making minor modifications to it. The attack assumes that the adversary controls the first and last nodes in the path through the network, with the goal of discovering which destination the initiator is communicating with. The authors discuss both correlating traffic "as is" as well as altering the traffic pattern at the first node in order to make correlation easier at the last node. For this second correlation attack, they describe a packet dropping technique which creates holes in the traffic; these holes then percolate through the network to the last router in the path. The analysis showed that without cover traffic (as employed in Tarzan [10], [11]) or defensive dropping [19], it is relatively easy to correlate communications through mix networks. Even with "normal" cover traffic where all packets between nodes look the same, Shmatikov and Wang show that the traffic analysis attacks are still viable [31]. A proposed solution is to add cover traffic that mimics traffic flows from the initiator's application.

A major limitation of all of the attacks described so far is that while they work well for small networks, they do not scale and may fail to produce reliable results for larger anonymizing networks. For example, Back's active latency measuring attack [1] describes measuring the latencies of circuits and then trying to determine the nodes that were being utilized from the latency of a specific circuit. As the number of nodes grows, this attack becomes more difficult (due to an increased number of possible circuits), especially as more and more circuits have similar latencies.

2.2.3. Congestion Attacks. A more powerful relative of the described timing attacks is the clogging or congestion attack. In a clogging attack, the adversary not only monitors the connection between two nodes but also creates paths through other nodes and tries to use all of their available capacity [1]; if one of the nodes in the target path is clogged by the attacker, the observed speed of the victim's connection should change.

In 2005, Murdoch and Danezis described an attack on Tor [23] in which they could reveal all of the routers involved in a Tor circuit. They achieved this result using a combination of a circuit clogging attack and timing analysis. By measuring the load of each node in the network and then subsequently congesting nodes, they were able to discover which nodes were participating in a particular circuit. This result is significant, as it reduces Tor's security during a successful attack to that of a collection of one hop proxies. This particular attack worked well on the fledgling Tor network with approximately fifty nodes; the authors experienced a high success rate and no false positives. However, their clogging attack no longer produces a signal that stands out on the current Tor network with thousands of nodes. Because today's Tor network is more heavily used, circuits are created and destroyed more frequently, so the addition of a single clogging circuit has less impact. Also, the increased traffic transmitted through the routers may lead to false positives or false negatives due to normal network fluctuations. We provide details about our attempt to reproduce Murdoch and Danezis's work in the appendix.

McLachlan and Hopper [20] propose a similar circuit clogging attack against MorphMix [29], disprov-

ing claims made in [32] that Morphmix was invulnerable to such an attack. Because all MorphMix users are *required* to also be mix servers, McLachlan and Hopper achieve a stronger result than Murdoch and Danezis: they can identify not only the circuit, but the user as well.

Hopper et al [15] build on the original clogging attack idea to construct a network latency attack to guess the location of Tor users. Their attack is two-phase: first use a congestion attack to identify the relays in the circuit, and then build a parallel circuit through those relays to estimate the latency between the victim and the first relay. One major contribution from this work is a more mathematical approach that quantifies the amount of information leaked in bits over time. We also note that without a working congestion attack, the practicality of their overall approach is limited.

3. Our Attack

Three features of Tor's design are crucial for our attack. First of all, Tor routers do not introduce any artificial delays when routing requests. As a result, it is easy for an adversary to observe changes in request latency. Second, the addresses of all Tor routers are publicly known and easily obtained from the directory servers. Tor developers are working on extensions to Tor (called bridge nodes) that would invalidate this assumption, but this service was not widely used at the time of this writing. Finally, the implementation that we attacked (Tor 0.2.0.29-rc) uses a small fixed path length (specifically three) but does not restrict users from establishing paths of arbitrary length.

The attack consists of three main steps. First, the adversary needs to ensure that the initiator repeatedly performs requests at known intervals. Second, the adversary observes the pattern in arrival times of these requests. Finally, the adversary changes the pattern by selectively performing a novel clogging attack on Tor routers to determine the entry node. We will now describe each of these steps in more detail.

3.1. JavaScript Injection

Our attack assumes that the adversary controls an exit node which is used by the victim to access an HTTP server. The attacker uses the Tor exit node to inject a small piece of JavaScript code (shown in Fig. 1) into an HTML response. The JavaScript code causes the browser to perform an HTTP request every second, and in response to each request, the adversary uses the exit node to return an empty response, which

```
<script language="javascript">
  var count = 0;
  var timer = 0;
  var xmlhttp = 0;
  function runonce() {
    xmlhttp = new XMLHttpRequest();
  }
  function start() {
    xmlhttp.abort();
    xmlhttp = new XMLHttpRequest();
    count++;
    if (timer)
      clearTimeout(timer);
    timer = setTimeout("start()",
                       1000);
    myDate = new Date();
    xmlhttp.open("GET",
      "/reportIn.html?num="+count+
      "&time=" + myDate.getTime(),
      true);
    xmlhttp.send("");
  }
</script>
```

Figure 1. JavaScript code injected by the adversary's exit node. Note that other techniques, such as HTML refresh, could also be used to cause the browser to perform periodic requests.

is thrown away by the browser. Since the JavaScript code may not be able to issue requests precisely every second, it also transmits the local system time (in milliseconds) as part of the request. This allows the adversary to determine the time difference between requests performed by the browser with sufficient precision. (Clock skew on the systems of the adversary and the victim is usually insignificant for the duration of the attack.)

The adversary then captures the arrival times of the periodic requests performed by the browser. Since the requests are small, an idle Tor network would result in the differences in arrival times being roughly the same as the departure time differences – these are known because they were added by the JavaScript as parameters to the requests. Our experiments suggest that this is often true for the real network, as most routers are not seriously congested most of the time. This is most likely in part due to TCP's flow control and Tor's built-in load balancing features. Specifically, the variance in latency between the periodic HTTP requests without an active congestion attack is typically in the range of 0–5s.

However, the current Tor network is usually not entirely idle and making the assumption that the victim's circuit is idle is thus not acceptable. Observing congestion on a circuit is not enough to establish that the node under the congestion attack by the adversary is part of the circuit; the circuit may be congested for other reasons. Hence, the adversary needs to also establish a baseline for the congestion of the circuit without an active congestion attack. Establishing measurements for the baseline is done before and after causing congestion in order to ensure that observed changes during the attack are caused by the congestion attack and not due to unrelated changes in network characteristics.

The attacker can repeatedly perform interleaved measurements of both the baseline congestion of the circuit and the congestion of the circuit while attacking a node presumed to be on the circuit. The attacker can continue the measurements until either the victim stops using the circuit or until the mathematical analysis yields a sufficiently high probability that the node under congestion attack is part of the circuit. Before we can describe details of the mathematical analysis, however, we have to discuss how congestion is expected to impact the latency measurements.

3.2. Impact of Congestion on Arrival Times

In order to understand how the congestion attack is expected to impact latency measurements, we first need to take a closer look at how Tor schedules data for routing. Tor makes routing decisions on the level of fixed-size *cells*, each containing 512 bytes of data. Each Tor node routes cells by going round-robin through the list of all circuits, transmitting one packet from each circuit with pending data (see Fig. 2). Usually the number of (active) circuits is small, resulting in little to no delay. If the number of busy circuits is large, messages may start to experience significant delays as the Tor router iterates over the list (see Fig. 3).

Since the HTTP requests transmitted by the injected JavaScript code are small (~250 bytes, depending on count and time), more than one request can fit into a single Tor cell. As a result multiple of these requests will be transmitted at the same time if there is congestion at a router. A possible improvement to our attack would be to use a lower level API to send the packets, as the XMLHttpRequest object inserts unnecessary headers into the request/response objects.

We will now characterize the network's behavior under congestion with respect to request arrival times. Assuming that the browser transmits requests at a perfectly steady rate of one request per second, a congested router introducing a delay of (at most) n seconds would cause groups of n HTTP requests to arrive with delays of approximately $0, 1, \ldots, n-1$ seconds respectively: the first cell is delayed by $n-1$ seconds, the cell arriving a second later by $n-2$ seconds, and the n-th cell arrives just before the round-robin scheduler processes the circuit and sends all n requests in one batch. This characterization is of course a slight idealization in that it assumes that n is small enough to allow all of the HTTP requests to be grouped into one Tor cell and that there are no other significant fluctuations. Furthermore, it assumes that the amount of congestion caused by the attacker is perfectly steady for the duration of the time measurements, which may not be the case.

Since we ideally expect to see delays in message arrival times for a congested circuit follow a roughly flat distribution between zero and n, it makes sense to compute a histogram of the delays in message arrival times. If the congestion attack is targeting a node on the circuit, we would expect to see a roughly equal number of messages in each interval of the histogram. We will call the shape of the resulting histogram *horizontal*. If the circuit is not congested, we expect to see most messages arrive without significant delay which would place them in the bucket for the lowest latency. We will call the shape of the resulting histogram *vertical*. Note that the clock difference between the victim's system and the adversary as well as the minimal network delay are easily eliminated by normalizing the observed time differences. As a result, the latency histograms should use the increases in latency over the smallest observed latency, not absolute latencies.

We can numerically characterize how vertical or how horizontal a histogram is by computing the angle of a least squares best-fit linear regression function through the origin of the coordinate system and the weighted points of the histogram. For the best-fit, a point representing k measurements in a particular time interval is given weight k. As discussed, based on Tor's cell scheduling algorithm (Fig. 2) and the small message size of the requests generated by the JavaScript code (Fig. 1), we would, under ideal circumstances, expect an angle near zero if the node under congestion attack is part of the circuit and, given suitably large latency intervals, a steep linear approximation function for the baseline histograms (as well as for the case of the congestion attack targeting the wrong node).

Naturally, the specific numerical angle of the linear approximation function for these histograms is meaningless — the x-axis of the histogram is time and the y-axis is the number of data points; thus, the

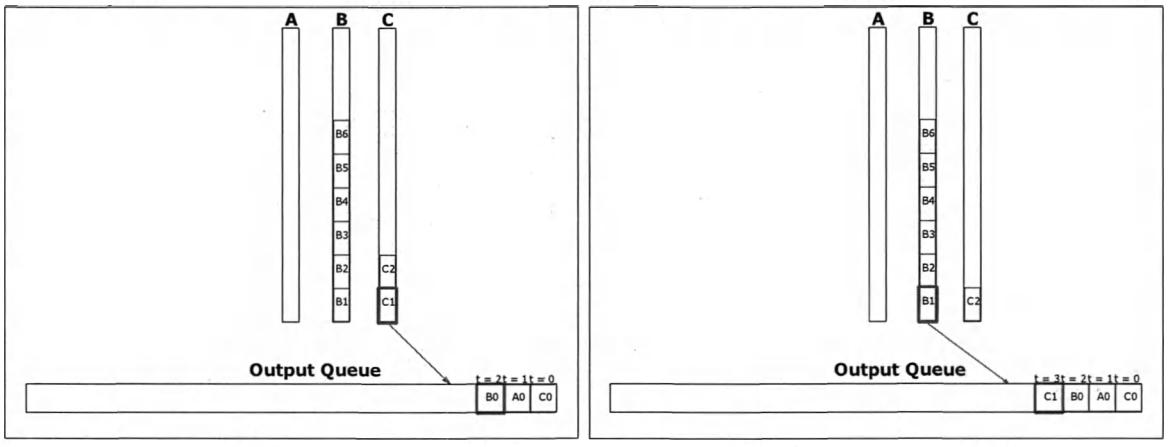


Figure 2. This example illustrates a Tor router which currently is handling three circuits at two points in time ($t = 3$ and $t = 4$). Circuits (A, B and C) have queues; cells are processed one at a time in a round-robin fashion. As the number of circuits increases, the time to iterate over the queues increases. The left figure shows the circuit queues and output queue before selection of cell `C1` for output and the right figure shows the queues after queueing `C1` for output. The thicker bottom box of queue C (left) and queue B (right) shows the current position of the round-robin queue iterator. At time $t = 1$ the last cell from queue A was processed leaving the queue A empty. As a result, queue A is skipped after processing queue C.

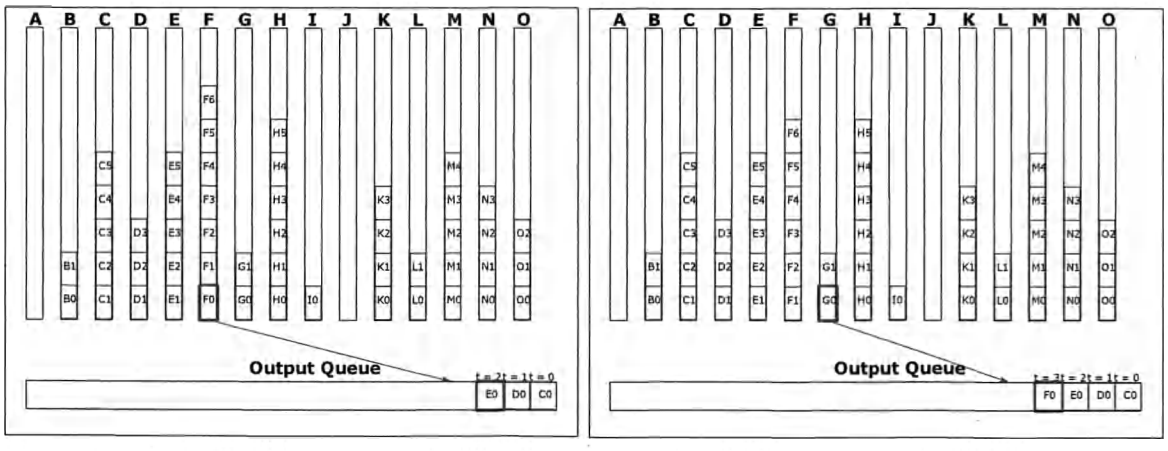


Figure 3. This example illustrates a Tor router under congestion attack handling 15 circuit queues. Note that if a circuit includes a node multiple times, the node assigns the circuit multiple circuit queues. In this example, not all of the circuit queues are busy — this may be because the circuits are not in use or because other routers on the circuit are congested. As in Fig. 2, the left and right figures show the state of the mix before and after queueing a cell, in this case `F0`.

absolute values cannot even be compared. However, it is possible to establish an expected range for the angles for an uncongested (or vertical) histogram. If the adversary is then able to selectively congest a particular node in the network and obtain a latency histogram for the victim's circuit with a linear approximation that has an angle outside of the expected range for vertical histograms, then the congested node is likely to be part of the circuit. Specifically, if the angle of the linear approximation is outside of the $p\%$ confidence interval for uncongested "vertical" histograms, then the probability is $p\%$ that the congested node is part of the circuit. Depending on the stage of the attack, the adversary may preferentially choose to congest a larger set of nodes at the same time. In that case, $p\%$ is the probability that one of the congested nodes is part of the circuit.

3.3. Congestion Attack

Now we focus on how the attacker controlling the exit node of the circuit will cause significant congestion at nodes that are suspected to be part of the circuit. In general, we will assume that all Tor routers are suspects and that in the simplest case, the attacker will iterate over all known Tor routers with the goal of finding which of these routers is the entry point of the circuit.

For each router X, the attacker constructs a long circuit that repeatedly includes X on the path. In order to ensure that Tor nodes cannot observe the circular nature of the circuit, two (or more) other (preferably high-bandwidth) Tor routers must be used before looping back to X. Note that the attacker could choose two different (involuntary) helper nodes in each loop involving X. Since X does not know that the circuit has looped back to X, Tor will treat the long attack circuit as many different circuits when it comes to packet scheduling (Fig. 2).

Once the circuit is sufficiently long (we typically found 21 hops to be effective, but in general this depends on the amount of congestion established during the baseline measurements), the attacker uses the circuit to transmit data. Note that a circuit of length m would allow an attacker with p bandwidth to consume $m \cdot p$ bandwidth on the Tor network, with X routing as much as $\frac{m \cdot p}{3}$ bandwidth. Since X now has to iterate over an additional $\frac{m}{3}$ circuits, this allows the attacker to introduce large delays at this specific router. The main limitation for the attacker here is time. The larger the desired delay d and the smaller the available attacker bandwidth k the longer it will take to construct an attack circuit of sufficient length m (since $d \sim k \cdot m$).

If the router X is independent of the victim circuit, the measured delays should not change significantly when the attack is running. If X is the entry node, the attacker should observe a delay pattern that matches the power of the attack – resulting in a horizontal latency variance histogram as described in Section 3.2. The attacker can vary the strength of the attack (or just switch the long attack circuit between idle and busy a few times) to confirm that the victim's circuit latency changes correlate with the attack.

3.4. Optimizations

The adversary can establish many long circuits to be used for attacks before trying to deanonymize a particular victim. Since idle circuits would not have any impact on measuring the baseline (or the impact of using another attack circuit), this technique allows an adversary to eliminate the time needed to establish circuits. As users can only be expected to run their browser for a few minutes, eliminating this delay may be important in practice.

In order to further speed up the process, an adversary can try to perform a binary search for X by initially running attacks on half of the routers in the Tor network. With pre-built attack circuits adding an almost unbounded multiplier to the adversary's resources, it is conceivable that a sophisticated attacker could probe a network of size s in $\log_2 s$ rounds of attacks.

In practice, pre-building a single circuit that would cause congestion for half the network is not feasible; the Tor network is not stable enough to sustain circuits that are thousands of hops long. Furthermore, the differences in available bandwidth between the routers complicates the path selection process. In practice, an adversary would most likely pre-build many circuits of moderate size, forgoing some theoretical bandwidth and attack duration reductions for circuits that are more reliable.

Furthermore, the adversary may be able to exclude certain Tor routers from the set of candidates for the first hop based on the overall round-trip latency of the victim's circuit. The Tor network allows the adversary to measure the latency between any two Tor routers [15], [23]; if the overall latency of the victim's circuit is smaller than the latency between the known second router on the path and another router Y, then Y is most likely not a candidate for the entry point.

Finally, the adversary needs to take into consideration that by default, a Tor user switches circuits every 10 minutes. This further limits the window of opportunity for the attacker. However, depending on the browser, the adversary may be able to cause the

browser to pipeline HTTP requests which would not allow Tor to switch circuits (since the HTTP session would not end). Tor's circuit switching also has advantages for the adversary: every 10 minutes there is a new chance that the adversary-controlled exit node is chosen by a particular victim. Since users only use a small number of nodes for the first node on a circuit (these nodes are called guard nodes [26]), the adversary has a reasonable chance over time to determine these guard nodes. Compromising one of the guard nodes would then allow full deanonymization of the target user.

4. Experimental Results

The results for this paper were obtained by attacking Tor routers on the real, deployed Tor network (during the Spring and Summer of 2008). However, in order to confirm the accuracy of our experiments and avoid ethical problems, we did not attempt to deanonymize real users. Instead, we established our own client circuits through the Tor network to our malicious exit node and then confirmed that our statistical analysis was able to determine the entry node used by our own client. Both the entry nodes and the second nodes on the circuits were normal nodes in the Tor network outside of our control. We did not receive any complaints about the short-lived malicious attacks that the Tor network was subjected to for the experiments.

The various roles associated with the adversary (exit node, malicious circuit client and malicious circuit webserver) as well as the "deanonymized" victim were distributed across different machines in order to minimize interference between the attacking systems and the targeted systems. For the measurements we had the simulated victim running a browser requesting and executing the malicious JavaScript code as well as a machine running the listening server to which the client transmits the "ping" signal approximately every second. The browser always connected to the same unmodified Tor client running on a separate machine via Privoxy [16]. The Tor client used the standard configuration except that we configured it to use our malicious exit node for its circuits. The other two nodes in the circuit were chosen at random by Tor. Our malicious exit node participated as a normal Tor router in the Tor network for the duration of the study (approximately six weeks).

For the congestion attack, we used three different machines, again in order to reduce interference; this separation may not be necessary for an attacker in practice. One of these machines ran the "client" which simply downloaded data from the Tor network. A modified Tor client was run on the second machine to access the Tor network. The Tor client was modified to allow us to choose two routers with high bandwidth and a specific target Tor node and build a long circuit involving these three nodes. The circuit would eventually be terminated by connecting from some exit node to our HTTP server running on the third system of the attacker. We used a high-bandwidth exit router in the Tor network to connect to our HTTP server which was running on the third attacker system and simply responded to requests with random data as fast as the network could process the data. As a result, the attacker systems maximize the utilization of the Tor circuit between them with throughput depending only on the bandwidth and performance of the individual Tor routers.

In order to cause congestion, we simply started the malicious client Tor process with the three chosen Tor routers and route length as parameters and then attempted to connect via `wget` [25] to the respective malicious server process. The amount of data received was recorded in order to determine bandwidth consumed during the tests. In order to further increase the load on the Tor network the experiments presented actually used two identical attacker setups with a total of six machines duplicating the three machine setup described in the previous paragraph. The overall strength of the attack was measured by the sum of the number of bytes routed through the Tor network by both attacker setups. For each trial, we waited to receive six hundred responses from the "victim"; since the browser transmitted requests to Tor at roughly one request per second, a trial typically took approximately ten minutes.

In addition to measuring the variance in packet arrival time while congesting a particular Tor router, each trial also included baseline measurements of the "uncongested" network in order to discover the normal variance in packet arrival time for a particular circuit. As discussed earlier, these baseline measurements are crucial in order to determine the significance of the effect that the congestion attack has had on the target circuit.

Fig. 4 illustrates how running the attack on the first hop of a circuit changes the latency of the received HTTP requests generated by the JavaScript code. The figure uses the same style chosen by Murdoch and Danezis [23], except that an additional line was added to indicate the strength of the attack (as measured by the amount of traffic caused by the congestion attack). For comparison, the first half of each of the figures shows the node latency variance when it is *not* under active congestion attack (or at least not by us).

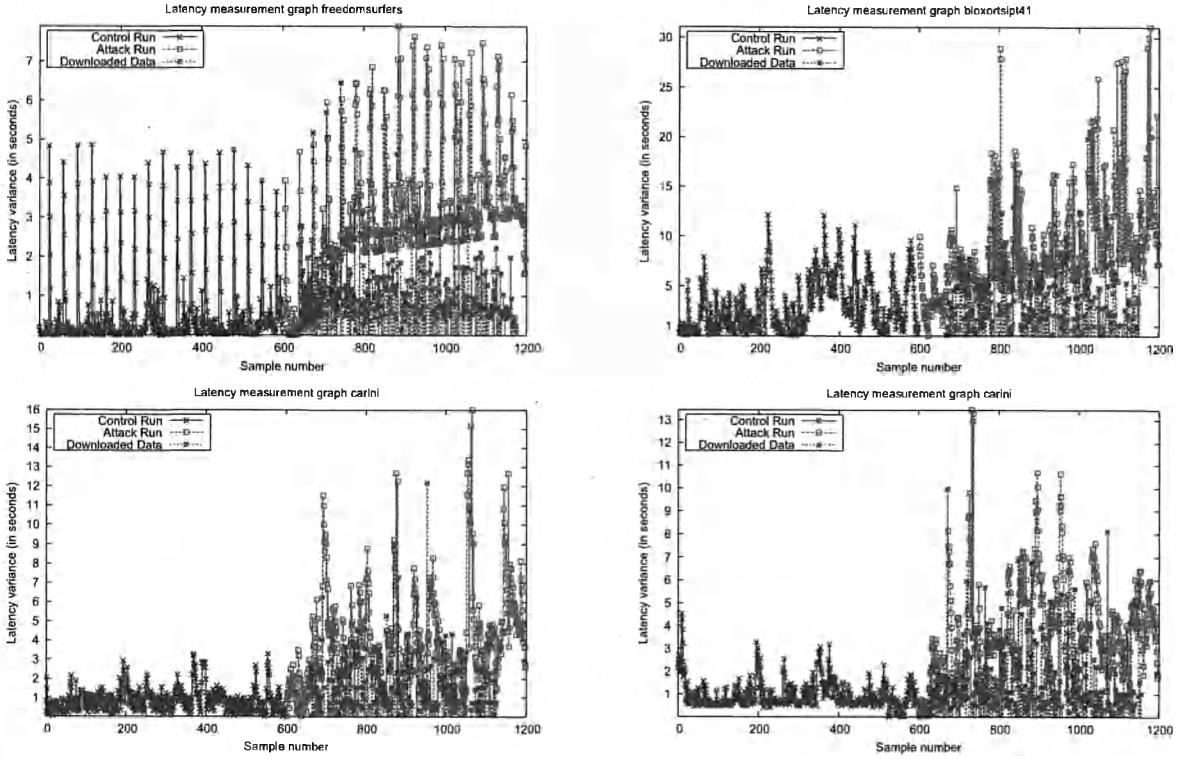


Figure 4. These figures show the results of perturbation of circuits in Tor and the resulting effects on latency. The X-axes show sample numbers (one per second), and the Y-axes are latency variance observed on the circuits in seconds. The attack on the first router of each circuit starts at time 600; the third line shows the amount of data (scaled) that transferred through the attack circuit. These are individual trials, each show a single control run and a single attack run. Statistical analyses are shown in Fig. 6 and Table 1. Note that throughout all of the figures, the same routers are used (i.e. the first image in Fig. 4 represents the same router as the first image in Fig. 5 and so on)

While the plots in Fig. 4 visualize the impact of the congestion attack in a simple manner, histograms showing the variance in latency are more suitable to demonstrate the significance of the statistical difference in the traffic patterns. Fig. 5 shows the artificial delay experienced by requests traveling through the Tor network as observed by the adversary. Since Tor is a low-latency anonymization service, the requests group around a low value for a circuit that is not under attack. As expected, if the entry node is under attack, the delay distribution changes from a steep vertical peak to a mostly horizontal distribution.

Fig. 5 also includes the best-fit linear approximation functions for the latency histograms which we will use to characterize how vertical or how horizontal the histogram is as described in Section 3.2. We repeated the baseline measurements to construct an expected range of angles for the approximation function. Fig. 6 shows the average angle of the latency distribution that is expected if the circuit's nodes are not under attack, the expected interval (in standard deviations) and the angle of the same circuit under attack. Remember that if the angle changes significantly, the attacker can be confident that the attacked node is on the circuit.

When doing this calculation, the size of the time intervals chosen clearly matters for getting reasonable results. If the time intervals are too small, virtually all intervals will have zero or one event, resulting in always near-horizontal approximation functions. Similarly, if the time interval is far too large, almost all data points will be in the first interval with no chance to even detect the introduced delays. Hence, the adversary should test different interval sizes (usually on the order of expected network delays) and

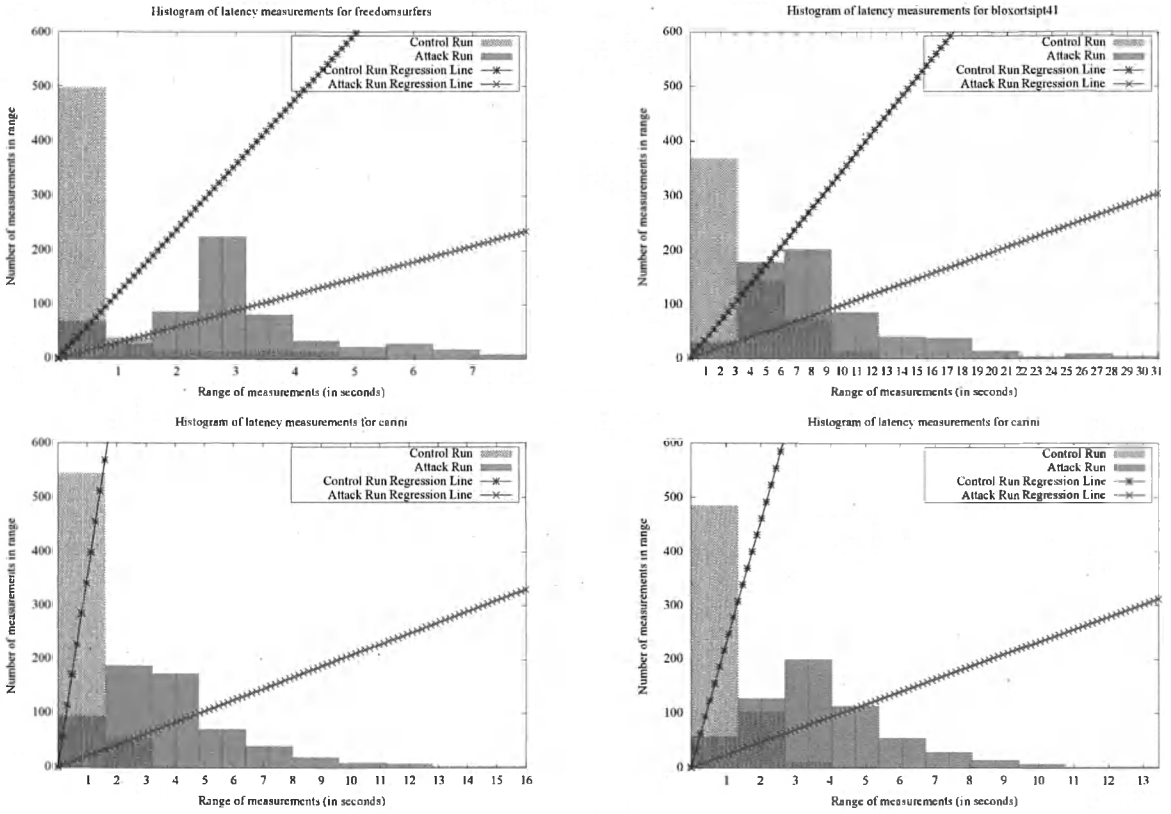


Figure 5. These figures show the results of four independent runs of our latency variance altering attack. The figures are histograms where the X-axis groups ranges of latency variance values together and the Y-axis represents the number of readings received in that range. The hash marked histogram is the graph of the unperturbed measurements on a circuit and the overlapping histogram show measurements from the same circuit during the attack. The effects of the attack are clear in the distribution of the latency variance values. The fir st and second lines are linear least squares fit approximations for the unperturbed and perturbed trials, respectively. As in Fig. 4, these data show the difference between a single control/attack run and are not averages of many runs as in Fig. 6 and Table 1.

determine experimentally which quantization produces good results (as indicated by small confidence intervals for the baseline measurement and significant deviations when the presumed entry node is under attack).

Table 1 lists the confidence levels that we were able to achieve in our experiments for the different circuits and their respective entry-nodes on the Tor network. A confidence level of c means that for the particular circuit, the probability that the given node is on the circuit is at least c, because the probability that the circuit would show the same latency changes when the adversary attacks nodes that are not on the circuit is less than $1-c$. Table 2 contrasts the standard deviation (of the histogram angle from Section 3.2) obtained while attacking the first hop with standard deviations observed while attacking other Tor routers. The data shows that our attack can be used to distinguish the first hop from other routers.

5. Proposed Solutions

An immediate workaround that would address the presented attack would be disabling of JavaScript by the end-user. However, JavaScript is not the only means by which an attacker could obtain timing information. For example, redirects embedded in the HTML header could also be used (they would, however, be more visible to the end-user). Links to images, frames and other features of HTML could also conceivably be used to generate repeated requests. Disabling all of

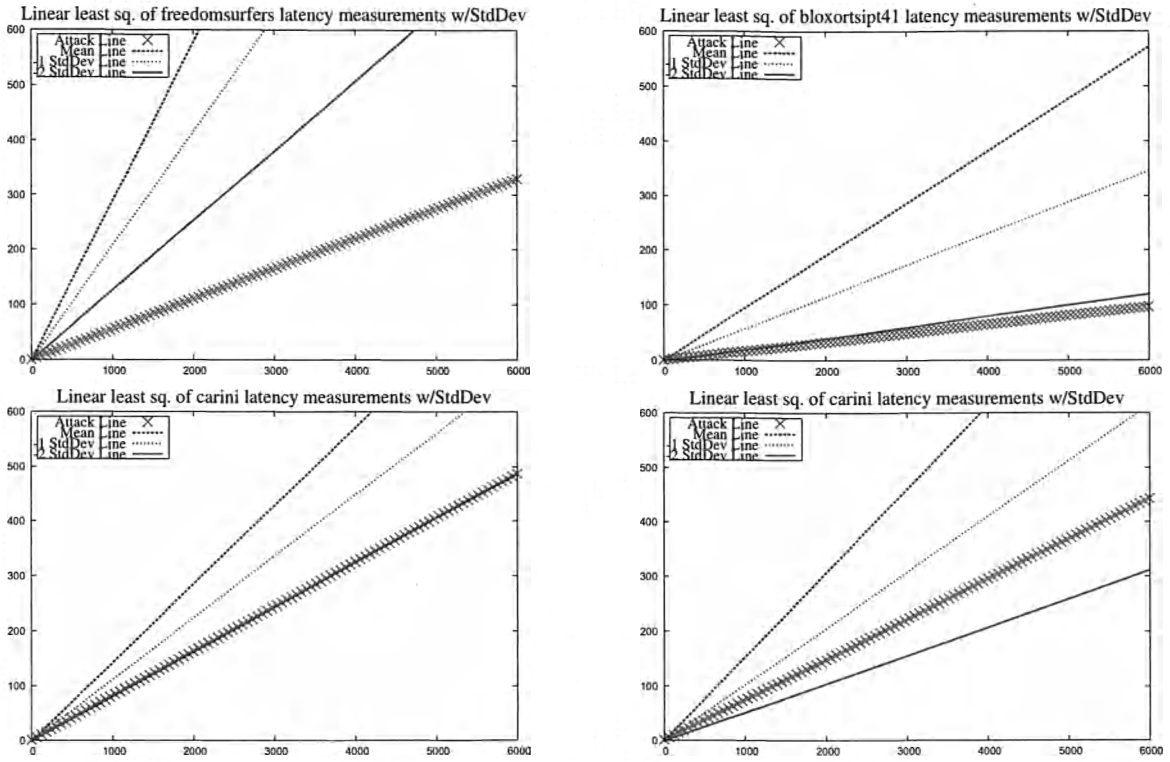


Figure 6. These figures depict the statistical analysis of the histograms created from the measurements obtained on three routers in the Tor network. The mean line of the angles obtained by doing a weighted least squares regression on the control histograms as well as two standard deviations out are shown along with the regression line of the histogram measurements taken during the attack on the Tor router. The subjective results in this light are clear, the attack regression line is typically significantly outside the range of likely values, as objectively shown in Table 1.

Router	Geographic Location	Peak BW	Configured BW	Avg. Attack Cost	Confidence	Attack Duration
`freedomsurfers`	Zurich, Switzerland	173.9 kB/s	153.6 kB/s	28.2 kB/s	0.9938	10m
`bloxortsipt41`	Florida, USA	54.1 kB/s	51.0 kB/s	4.5 kB/s	0.9826	10m
`carini` (attack 1)	Virginia, USA	98.2 kB/s	61.4 kB/s	3.0 kB/s	0.9772	10m
`carini` (attack 2)	Virginia, USA	98.2 kB/s	61.4 kB/s	5.4 kB/s	0.8944	10m
`carini` (combined)	Virginia, USA	98.2 kB/s	61.4 kB/s	4.2 kB/s	0.9950	20m

Table 1. This table shows the confidence levels established by our analysis for three circuits and the respective first router of each circuit. The stated confidence that the entry node belongs to the circuit is determined by how far outside the expected range the recorded data under congestion attack was. The table also lists basic properties of the entry node and the duration of the congestion attack. The geographic locations were determined using the hostip.info website and may not be entirely accurate.

Router	Std. Dev.	Peak BW	Configured BW
carini	2.57	98 kB/s	61 kB/s
bettyboop	1.59	2,000 kB/s	102,000 kB/s
1john2	1.15	122 kB/s	20 kB/s
NSAFortMeade	-0.41	202 kB/s	150 kB/s
zedz	-0.44	3,000 kB/s	100,000 kB/s

Table 2. This table lists the standard deviations observed in the angle of the linear approximations of latency histograms obtained for a circuit with carini as the first hop while congesting various Tor routers (including carini). The peak bandwidth is the maximum amount of traffic routed by the respective router in a 10s interval over the past day. The configured bandwidth is the bandwidth cap specified by the user in the Tor configuration.

these features has the disadvantage that the end-user's browsing experience would suffer.

A better solution would be to thwart the denial-of-service attack inherent in the Tor protocol. Attackers with limited bandwidth would then no longer be able to significantly impact Tor's performance. Without the ability to selectively increase the latency of a particular Tor router, the resulting timing measurements would most likely give too many false-positives. We have extended the Tor protocol to limit the length of a path. The details are described in [9]; we will detail the key points here.

In the modified design, Tor routers now must keep track of how often each circuit has been extended and refuse to route messages that would extend the circuit beyond a given threshold. This can be done by tagging messages that *may* extend the circuit with a special flag that is not part of the encrypted stream. The easiest way to do this is to introduce a new Tor cell type that is used to flag cells that may extend the circuit. Routers then count the number of messages with the special flag and refuse to route more than a given small number (at the moment, eight) of those messages. Routers that receive a circuit-extension request would check that the circuit-extension message was contained in a cell of the appropriate type. Note that these additional checks do not change the performance characteristics of the Tor network.

While this change prevents an attacker from constructing a circuit of arbitrary length, it does not fully prevent the attacker from constructing a path of arbitrary length. The remaining problem is that the attacker could establish a circuit and then from the exit node reconnect to the Tor network again as a client. So in order to make the construction of long paths impossible, Tor relays would need to be configured to refuse incoming connections from exit nodes on their client port. Since all Tor exit nodes are publicly known this could easily be done. Note that the solution proposed in [27] — limiting circuit construction to trees — does not address this issue; furthermore, it increases overheads and implementation complexity far beyond the change proposed here and (contrary to the claims in [27]) may also have an impact on anonymity, since it requires Tor to fundamentally change the way circuits are constructed.

Finally, given that strong adversaries may be able to mount latency altering attacks without Tor's "help", Tor users might consider using a longer path length than the minimalistic default of three. This would involve changes to Tor, as currently the only way for a user to change the default path length would be to edit and recompile the code (probably out of scope for a "normal" user). While the presented attack can be made to work for longer paths, the number of false-positives and the time required for a successful path discovery increase significantly with each extra hop. Using a random path length between four and six would furthermore require the adversary to confirm that the first hop was actually found (by determining that none of the other Tor routers could be a predecessor). Naturally, increasing the path length from three to six would double latency and bandwidth requirements.

6. Conclusion

The possibility of constructing circuits of arbitrary length was previously seen as a minor problem that could lead to a DoS attack on Tor. This work shows that the problem is more serious, in that an adversary could use such circuits to improve methods for determining the path packets take through the Tor network. Furthermore, the minimalistic default choice of circuits of length three is questionable, given that an adversary controlling an exit node would only need to recover a tiny amount of information to learn the entire circuit. Minimal changes to the Tor protocol with hardly any noticeable performance impact have been deployed to address these problems.

References

[1] A. Back, U. Möller, and A. Stiglic, "Traffic analysis attacks and trade-offs in anonymity providing systems," in *Proceedings of Information Hiding Workshop (IH 2001)*, I. S. Moskowitz, Ed. Springer-Verlag, LNCS 2137, April 2001, pp. 245–257.

[2] N. Borisov, G. Danezis, P. Mittal, and P. Tabriz, "Denial of service or denial of security? How attacks on reliability can compromise anonymity," in *CCS '07: Proceedings of the 14th ACM conference on Computer and communications security*. New York, NY, USA: ACM, October 2007, pp. 92–102.

[3] D. L. Chaum, "Untraceable electronic mail, return addresses, and digital pseudonyms," *Commun. ACM*, vol. 24, no. 2, pp. 84–90, February 1981.

[4] W. Dai, "Two attacks against freedom." 2000. [Online]. Available: http://www.weidai.com/freedom-attacks.txt

[5] G. Danezis, R. Dingledine, and N. Mathewson, "Mixminion: Design of a Type III Anonymous Remailer Protocol," in *Proceedings of the 2003 IEEE Symposium on Security and Privacy*, May 2003, pp. 2–15.

[6] Y. Desmedt and K. Kurosawa, "How to break a practical MIX and design a new one," in *Advances in Cryptology — Eurocrypt 2000, Proceedings*. Springer-Verlag, LNCS 1807, 2000, pp. 557–572.

[7] C. Diaz and A. Serjantov, "Generalising mixes," in *Proceedings of Privacy Enhancing Technologies workshop (PET 2003)*, R. Dingledine, Ed. Springer-Verlag, LNCS 2760, March 2003, pp. 18–31.

[8] R. Dingledine, N. Mathewson, and P. Syverson, "Tor: The second-generation onion router," in *Proceedings of the 13th USENIX Security Symposium*, August 2004.

[9] R. A. for review, "Tor proposal 110: Avoiding infinite length circuits," March 2007. [Online]. Available: https://svn.torproject.org/svn/tor/trunk/doc/spec/proposals/110-avoid-infinite-circuits.txt

[10] M. J. Freedman and R. Morris, "Tarzan: a peer-to-peer anonymizing network layer," in *CCS '02: Proceedings of the 9th ACM conference on Computer and communications security*. New York, NY, USA: ACM, November 2002, pp. 193–206.

[11] M. J. Freedman, E. Sit, J. Cates, and R. Morris, "Introducing tarzan, a peer-to-peer anonymizing network layer," in *IPTPS '01: Revised Papers from the First International Workshop on Peer-to-Peer Systems*. London, UK: Springer-Verlag, 2002, pp. 121–129.

[12] D. M. Goldschlag, M. G. Reed, and P. F. Syverson, "Hiding Routing Information," in *Proceedings of Information Hiding: First International Workshop*, R. Anderson, Ed. Springer-Verlag, LNCS 1174, May 1996, pp. 137–150.

[13] C. Gülcü and G. Tsudik, "Mixing E-mail with Babel," in *Proceedings of the Network and Distributed Security Symposium - NDSS '96*. IEEE, February 1996, pp. 2–16.

[14] J. Han and Y. Liu, "Rumor riding: Anonymizing unstructured peer-to-peer systems," in *ICNP '06: Proceedings of the Proceedings of the 2006 IEEE International Conference on Network Protocols*. Washington, DC, USA: IEEE Computer Society, Nov 2006, pp. 22–31.

[15] N. Hopper, E. Y. Vasserman, and E. Chan-Tin, "How much anonymity does network latency leak?" in *CCS '07: Proceedings of the 14th ACM conference on Computer and communications security*. New York, NY, USA: ACM, October 2007, pp. 82–91.

[16] F. Keil, D. Schmidt *et al.*, "Privoxy - a privacy enhancing web proxy." [Online]. Available: http://www.privoxy.org/

[17] D. Kesdogan, J. Egner, and R. Büschkes, "Stop-and-go MIXes: Providing probabilistic anonymity in an open system," in *Proceedings of the Second International Workshop on Information Hiding*. London, UK: Springer-Verlag, LNCS 1525, 1998, pp. 83–98.

[18] O. Landsiedel, A. Pimenidis, K. Wehrle, H. Niedermayer, and G. Carle, "Dynamic multipath onion routing in anonymous peer-to-peer overlay networks," *Global Telecommunications Conference, 2007. GLOBECOM '07. IEEE*, pp. 64–69, Nov. 2007.

[19] B. N. Levine, M. K. Reiter, C. Wang, and M. K. Wright, "Timing attacks in low-latency mix-based systems," in *Proceedings of Financial Cryptography (FC '04)*, A. Juels, Ed. Springer-Verlag, LNCS 3110, February 2004, pp. 251–265.

[20] J. McLachlan and N. Hopper, "Don't clog the queue! circuit clogging and mitigation in p2p anonymity schemes," in *Financial Cryptography*, ser. Lecture Notes in Computer Science, G. Tsudik, Ed., vol. 5143. Springer, 2008, pp. 31–46.

[21] U. Möller, L. Cottrell, P. Palfrader, and L. Sassaman, "Mixmaster Protocol — Version 2," IETF Internet Draft, December 2004.

[22] S. J. Murdoch, "Covert channel vulnerabilities in anonymity systems," Ph.D. dissertation, University of Cambridge, December 2007.

[23] S. J. Murdoch and G. Danezis, "Low-cost traffic analysis of Tor," in *SP '05: Proceedings of the 2005 IEEE Symposium on Security and Privacy*. Washington, DC, USA: IEEE Computer Society, May 2005, pp. 183–195.

[24] A. Nambiar and M. Wright, "Salsa: a structured approach to large-scale anonymity," in *CCS '06: Proceedings of the 13th ACM conference on Computer and communications security*. New York, NY, USA: ACM, October 2006, pp. 17–26.

[25] H. Nikšic', "GNU wget," 2008, maintained by Micah Cowan. [Online]. Available: http://www.gnu.org/software/wget/

[26] L. Øverlier and P. Syverson, "Locating hidden servers," in *SP '06: Proceedings of the 2006 IEEE Symposium on Security and Privacy*. Washington, DC, USA: IEEE Computer Society, May 2006, pp. 100–114.

[27] V. Pappas, E. Athanasopoulos, S. Ioannidis, and E. P. Markatos, "Compromising anonymity using packet spinning," in *Proceedings of the 11th Information Security Conference (ISC 2008)*, ser. Lecture Notes in Computer Science, T.-C. Wu, C.-L. Lei, V. Rijmen, and D.-T. Lee, Eds., vol. 5222. Springer, 2008, pp. 161–174.

[28] A. Pfitzmann, B. Pfitzmann, and M. Waidner, "ISDN-mixes: Untraceable communication with very small bandwidth overhead," in *Proceedings of the GI/ITG Conference on Communication in Distributed Systems*, February 1991, pp. 451–463.

[29] M. Rennhard and B. Plattner, "Introducing MorphMix: Peer-to-Peer based Anonymous Internet Usage with Collusion Detection," in *WPES '02: Proceedings of the 2002 ACM workshop on Privacy in the Electronic Society*. New York, NY, USA: ACM, November 2002, pp. 91–102.

[30] A. Serjantov, R. Dingledine, and P. Syverson, "From a trickle to a flood: Active attacks on several mix types," in *IH '02: Revised Papers from the 5th International Workshop on Information Hiding*, F. Petitcolas, Ed. London, UK: Springer-Verlag, LNCS 2578, 2003, pp. 36–52.

[31] V. Shmatikov and M.-H. Wang, "Timing analysis in low-latency mix networks: Attacks and defenses," in *Proceedings of the 11th European Symposium on Research in Computer Security (ESORICS)*, September 2006, pp. 236–252.

[32] R. Wiangsripanawan, W. Susilo, and R. Safavi-Naini, "Design principles for low latency anonymous network systems secure against timing attacks," in *Proceedings of the fifth Australasian symposium on ACSW frontiers (ACSW '07)*. Darlinghurst, Australia, Australia: Australian Computer Society, Inc, 2007, pp. 183–191.

Appendix

We attempted to reproduce Murdoch and Danezis's work [23] on the Tor network of 2008. Murdoch provided us with their code and statistical analysis framework which performs their congestion attack while measuring the latency of the circuit. Their analysis also determines the average latency and uses normalized latencies as the strength of the signal.

As in this paper, the adversary implemented by Murdoch and Danezis repeatedly switches the congestion attack on and off; a high correlation between the presence of high latency values and the congestion attack being active is used to determine that a particular router is on the circuit. If such a correlation is absent for the correct router, the attack produces false-negatives and fails. If a strong correlation is present between high latency values and random time periods (without an active attack) then the attack produces false-positives and also fails.

Fig 7 shows two runs of the method used in [23], one with the congestion attack being active and one without. The figure plots the observed latency of a router over time. Blue bars are used to indicate when the congestion attack was active. Note that in the second graph, the congestion attack was run against a Tor router unrelated to the circuit and thus inactive for the circuit that was measured. Any correlation observed in this case implies that Murdoch and Danezis's attack produces false-positives. At times where the attack is active (including, depending on the figure, activity on an unrelated Tor router), red lines are drawn to latency values above average to mark latencies that correlate with the attack.

Due to the large amount of traffic on the Tor network Murdoch and Danezis's analysis is unable to differenciate between normal congestion and congestion caused by the attacker: the small amount of congestion caused by Murdoch and Danezis is lost in the noise of the network. Table 3 shows some representative correlation values that were computed using the statistical analysis from [23] when performed on the modern Tor network. Note that the correlation values are high regardless of whether or not the congestion attack was actually performed on the respective router during the recordings. Ideally, high correlation values would only appear when the attack was on. This data shows that Murdoch and Danezis's attack no longer works on today's Tor network.

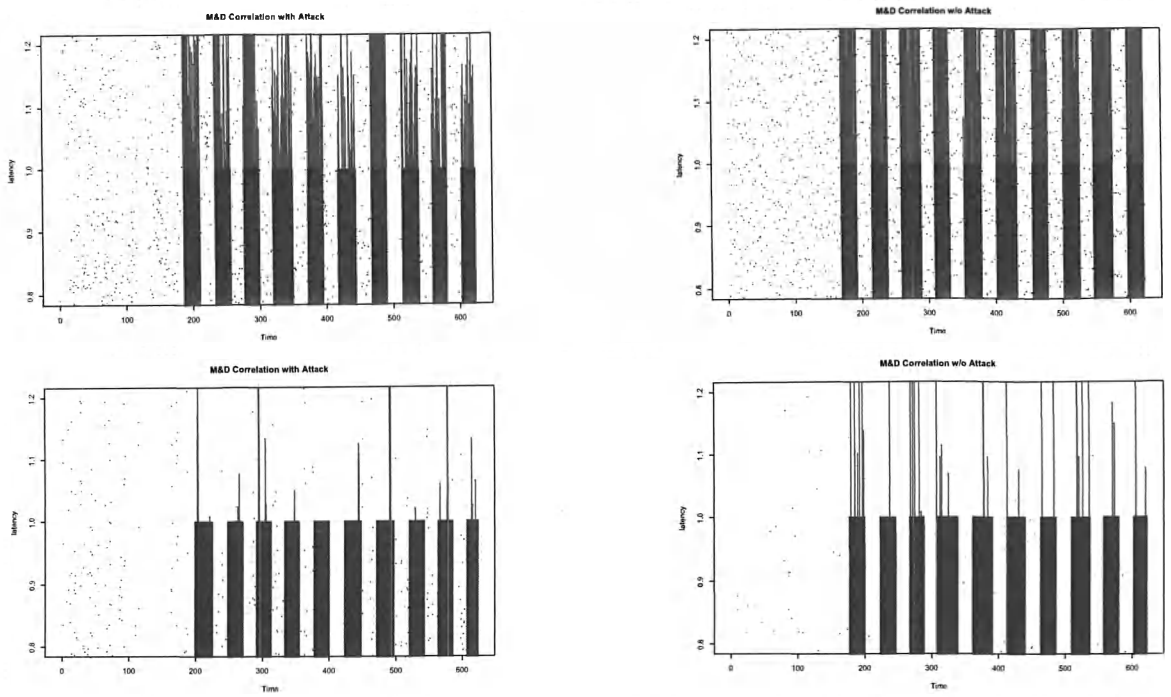


Figure 7. The graphs on the left illustrate the correlation between increased latency and the congestion attack performed by Murdoch and Danezis. However, the graphs on the right suggests a similar correlation pattern even when the attack was "off" (or targeting unrelated Tor routers). This is due to the high volume of traffic on today's Tor network causing baseline-congestion which makes their analysis too indiscriminate.

Router	Correlation	Attacked?	Peak BW	Configured BW
morphium...	1.43	Yes	222 kB/s	201 kB/s
ccc23	1.34	No	5414 kB/s	5120 kB/s
humanist...	1.18	No	5195 kB/s	6000 kB/s
mikezhan...	1.07	No	1848 kB/s	2000 kB/s
hummingb...	1.03	No	710 kB/s	600 kB/s
ccc42	1.00	Yes	1704 kB/s	5120 kB/s
degaussY...	1.00	No	4013 kB/s	4096 kB/s
ephemera	0.91	Yes	445 kB/s	150 kB/s
fissefja...	0.99	Yes	382 kB/s	50 kB/s
zymurgy	0.86	Yes	230 kB/s	100 kB/s
charlesb...	0.53	Yes	2604 kB/s	1300 kB/s

Table 3. This table shows the correlation values calculated using the Murdoch and Danezis's attack on the Tor network in Spring of 2008. False positives and false negatives are both abundant; many congested routers are marked as part of the circuit when they are not.