

From: [Diane Sturgis](#)
To: [Andrew Lewman](#) (b) (6)
Cc: [Kelly DeYoe](#)
Subject: Task 5
Date: Monday, June 18, 2012 3:24:36 PM
Attachments: [BBG50-1-12-0508.pdf](#)

Andrew,

Attached is copy of Task Order for your records.

Diane

From: [Ken Berman](#)
To: [Roger Dingleline](#)
Cc: [Kelly DeYoe](#)
Subject: Telex at UMI
Date: Friday, December 09, 2011 2:58:45 PM

Roger – do you have a connection with Haldeman at the University of Michigan as regards the Telex project? Any idea where he is on this?

Ken

From: [Jeff Gillis](#)
To: [Roger Dingleline](#); (b) (6); [Mike Perry](#); (b) (6)
Cc: [Damian Menscher](#); [Kelly DeVoe](#); [Minnie Ingersoll](#); [Trisha Weir](#); [Nelis Provos](#); [Chris DiBona](#); [Jason](#); [Dorothy Chou](#)
Subject: Thanks
Date: Thursday, April 21, 2011 3:59:59 AM

Roger, Mike, Andrew,

Thank you very much for making time to visit and call in to talk with us today, and then with Damian afterwards. It was very interesting, and we're going to vet the collaboration ideas that came up. Thanks also to Kelly for referring Minnie to Tor and getting the ball rolling. Let's stay in touch, and feel free to reach out to any of us at any time if you think of other issues or ways to work together.

--Jeff on behalf of the Google team

From: [Roger Dingleline](#)
To: [Demetria Anderson](#)
Cc: [Kelly DeYoe](#)
Subject: Third invoice for Moria Research Labs, BBGCON1806S6149
Date: Wednesday, October 25, 2006 9:50:29 PM
Attachments: [mrl-ibb3.pdf](#)

Hi Demetria,

Attached is my third invoice for contract BBGCON1806S6149. Please let me know if I've left out any necessary information.

Thanks!
--Roger

From: [Roger Dingledine](#)
To: [Kelly DeYoe](#)
Cc: [Ken Berman](#); [Hiu Ho](#)
Subject: Tor / China plan
Date: Friday, June 16, 2006 5:15:57 AM

On Mon, Jun 05, 2006 at 04:37:09PM -0400, Roger Dingledine wrote:
> An outline for a design doc coming soon...

Hi folks. Here is The Plan, broken down into three phases, which we can pursue in parallel. Phase one is mostly me, phase two is where Bennett comes in, and then we all team up to take phase two to phase three.

Phase one: Core Tor development to make it easier to become a relay. Generally improve usability of Tor and supporting programs.

[Phase one-prime: Fix the fact that Windows XP networking doesn't handle being a Tor server as well as we'd like. Mike Chiussi at UToronto is making a start at this. We can skip this step by telling Windows volunteers to come back later -- but that might impact the number of volunteer relays we end up with.]

Phase two: Come up with ways to communicate some bootstrap relays to dissidents. Try to make China not notice despite all the media who want to write about us. Iterate.

Phase three: Code for a separate "unlisted" Tor network, handling the easy/promising cases from phase two.

- a) Change directory authority code to enable a separate, parallel Tor network that doesn't broadcast the addresses of all its participants.
- b) Add interfaces to Vidalia to allow a user to sign up to be one of these secondary relays; document and test.
- c) Add interfaces to Vidalia for the people who are blocked; document and test.

To start phase two, we need to enumerate all the schemes we can imagine. Then we can list their pros and cons, prioritize them, and predict what sort of interfaces will be helpful for each in phase three.

Here are a few starts that need to be fleshed out and need more thought.

- The dir server just gives you a random IP address if you ask. This is great until the dir server gets censored, or until the adversary starts collecting IP addresses too.
- To solve the blocked-dirserver problem, we could
 - Have a way to manually enter a relay address learned out-of-band (e.g. via social network).
 - Encourage users to use open proxies or other proxies to reach the dir server.
- To solve the adversary-collecting-addresses problem, we could
 - Add a captcha.
 - Require a valid email address at e.g. yahoo -- leveraging somebody else's captcha system.
 - Give out a single address to all queries for a given hour, so you need to ask every single hour for many weeks in order to learn the entire list.
 - Give accounts to users, let them earn trust, try to detect when the addresses they use get blocked, and reward when not.
- ...

- Send lots of spam to bootstrap relay IPs.
- Have users scan the Internet for relay IPs (won't work while we're small, but we're trying to be complete here).
- We could sneak a big pile of relay addresses into popular software, and have them each suddenly enabled one day.
- ...

From: [Roger Dingledine](#)
To: [Ken Berman](#); [Kelly DeYoe](#); [Sho Ho](#)
Cc: [REDACTED]
Subject: Tor "net installer"
Date: Tuesday, January 20, 2009 10:27:27 PM

Hi folks,

Here's a heads up on a new development. We've been working lately on a "secure updater" named Thandy so a) Tor users can learn when there's a new version they should get, and b) they can automatically fetch it, and check package signatures and so on, optionally doing the fetch via Tor in case their local firewall blocks the Tor website:

<https://svn.torproject.org/svn/updater/trunk/specs/thandy-spec.txt>

That part shouldn't be too surprising to you, since I've been including it in the monthly reports. The interesting part is that now that we have our secure updater working, people can actually use it to bootstrap their Tor in the first place: we just ship Thandy with a tiny wrapper script, and it pulls down the packages it needs and then launches Vidalia. You can see more about this "net install" approach here:

<https://data.peertech.org/files/demo/updater/netinstall.html>

We've been getting features requests for a year or so now from folks in Iran, Saudi Arabia, etc who have modems and only Internet Explorer. They can't download the Tor bundles because their modem hangs up periodically and IE doesn't have any "resume download" features. So this "fetch a small core program which auto-fetches the correct versions of the software you should have and checks all the signatures" idea might be just the thing for them.

It isn't ready for prime-time yet, since we need to keep mucking with the internals to become more compatible with how ordinary Windows apps are supposed to behave (e.g. to take out the assumption that every user has Windows development libs installed). But once we've done said mucking, we could conceivably get the "small core program" down to 1-2MB.

--Roger

From: Roger Dingledine
To: (b) (6); (b) (6); Kelly DeYoe
Cc: Ken Berman; (b) (6)
Subject: Tor + IBB: moving forward
Date: Monday, February 13, 2006 10:52:19 AM

Hi Bennett, Hiu, Kelly,

Here's a mail that has been queueing until I learned more details about our plan. I'd still like to hear from Ken what vision and goals he's hoping for (maybe Kelly can learn this and include it in our upcoming conference call?), but in the meantime, here's an overview of some of the tasks that need to be tackled.

My goal here is to see if any of these paragraphs catch your eye. My experience is that people do their best work when they're excited about it, so -- does any of this excite you in particular? We have a lot of different tasks to work on, so whatever you're most interested in is clearly the right thing to work on. Or are there other related things that you think need attention too? I'm open to suggestions.

My focus for the next little while is to get the Tor 0.1.1.x release candidate ready. This new Tor version includes a more scalable and secure directory system, and we'll need it in order for the Tor network to grow much larger.

After that, my plan is to start focusing on server usability -- how to make all the internals of Tor work correctly if we have a button to sign yourself up as a relay for our alternate Tor network. In addition, we could really use some good simple documentation for how to forward a port through some typical home routers, how to set Tor up, etc. We could also use some help on the Tor GUI that lets people choose to become servers. It's pretty far from having the 'help China' button on it. (In fact, we have no front-end at all for OS X, Linux, etc.)

We also need to think about a strategy for how to spin this move in terms of Tor's overall direction. I would guess that we don't want to loudly declare war on China, since this only harms our goals? But we also don't want to hide the existence of funding from IBB, since "they're getting paid off by the feds and they didn't tell anyone" sounds like a bad Slashdot title for a security project. Is it sufficient just to always talk about Iran, or is that not subtle enough?

Somewhere in this, we need to keep processing volunteer mail such as the nice people who just translated the Tor site into Chinese and Russian, and keep trying to support server operator questions, and keep trying to find somebody to help with Windows stability, docs, faqs, user support, and so forth.

In parallel to this, we're going to need somebody to design a GUI controller for the people who want to be Tor clients but can't make it to the main Tor network directly. The actual back-end talking to the Tor client is pretty easy, but it's probably not good enough to have it in English, and there are a lot of design issues to work out too:

We need to enumerate some ways for clients to bootstrap relay IP addresses -- a couple of default addresses just in case they work, a way to manually

enter them, the instant-messaging account that Bennett was talking about, receiving them in the mass-mailing spams, and so forth. Once the Tor client knows a few relay IP addresses, it can automatically build the connection and reach the main directory server.

...Which also needs to be figured out. I had originally envisioned this as just a little cgi script on a web page somewhere, but now that I think about it more, we probably want some features like being able to check whether a relay is actually working right now. All of that is already working if we use a normal Tor directory server, and we can modify it to not answer requests for the whole directory, and to answer with our special algorithm.

...Which brings us to the algorithm for disbursing backup relay addresses. This could vary widely from Bennett's "if they already know the public key then they can lookup the current server descriptor for that key but nothing more" to "we'll give them a couple of random addresses every time they ask for some" to "everybody who asks this hour gets this IP address, and next hour we'll switch to giving out a new one" to "ask them to register pseudonymous accounts and try to build a system to detect which accounts defect". We need to enumerate these options and make a concise list of pros and cons. Remember that we're not just targetting one country, so it may be reasonable to deploy different strategies simultaneously.

We'll want to build a plan for bootstrapping the whole thing (if we make it too locked down originally, then we'll end up with no users, and there will be no point). I think it's fine to assume that when we first start out nobody will care, but we need to consider and anticipate some of the transition problems, for example so we avoid letting them enumerate the whole set of relays and destroy the progress we've made once they do start to care.

There's clearly more to plan and more to lay out, but hopefully this will get us moving in the right direction. Another topic for the conference call is integrating this discussion into the normal Tor mailing lists so we can do more design out in the open (or at least with a broader set of developers).

Thanks,
--Roger

From: Roger Dingledine
To: Ali Alvami
Cc: Kelly DeYoe; Ken Berman; [REDACTED]
Subject: Tor and Saudi Arabia
Date: Tuesday, September 23, 2008 9:36:12 PM

Hi Ali,

Great to talk to you last week.

I mentioned Ethan Zuckerman's "anonymous blogging" tutorial in the phone call: <http://advocacy.globalvoicesonline.org/projects/guide/>
It's a few years out of date, but still mostly right.

The most important change since that tutorial came out is the introduction of the Tor Browser Bundle: <https://www.torproject.org/torbrowser/>
Note that we do have an Arabic version of the Tor Browser Bundle, though I think the components are only about half translated.

There are people in Saudi Arabia using Tor right now. Tor provides anonymity as well as circumvention, so it's hard to count them very accurately :), but I think the number of people there using Tor at this moment is between 100 and 1000.

So:

A) Can you forward this mail to Jeremiah, and introduce me? I'm planning to be in La Jolla for some subset of Oct 21-24.

B) We could use help finishing the Arabic translation, and verifying the current translation. You can read all about how to do this here: <https://www.torproject.org/translation-overview.html.en>
I fear you still need to be a bit technically adept to navigate the translation system; and translators will probably want to understand the basics of Tor to be best at translating. Please let us know if you have any questions or want more specific help.

C) We really need to learn more about how Saudi Arabia does its blocking, and what it's likely to block next, before we can plan too much there. Do you know anybody good who can answer questions like that?

D) You may also be interested in a new feature we've been working on, called 'bridges'. They are designed for the next step in the arms race, once a government firewall decides to target Tor specifically and try to block connections to it. <https://www.torproject.org/bridges>

Thanks,
--Roger

From: [Roger Dingledine](#)
To: [Ken Berman](#); [Kelly DeYoe](#); [Sho Ho](#)
Subject: Tor blocking resistance: likely attacks and defenses
Date: Friday, February 20, 2009 1:53:29 AM

Hi Ken, Kelly, Sho,

Below is our first stab at analyzing weaknesses in Tor's current TLS footprint. We're not planning to do anything with the results quite yet (see the two conflicting conclusions at the end), but we figured it was a good move to get it started early so we could keep it in mind during further development, and shape it as needed.

Let me know if you have any questions or thoughts (or answers ;). Please check with me before sharing it with anyone -- we're big fans of transparency in general, but this is one of the exceptions.

Thanks,
--Roger

TOR BLOCKING RESISTANCE: LIKELY ATTACKS AND DEFENSES

Analysis by Nick Mathewson, based largely on analysis by Steven Murdoch.
Current as of February 2009.

(Copyright 2009 The Tor Project. This is sensitive strategic analysis;
do not circulate it.)

Assumptions

~~~~~

We assume a censor who wants to block all or most Tor connections, but who wants to maintain an otherwise usable Internet. We assume that the censor is willing to accept some false positive rate (that is, to block some non-Tor connections), but not a huge one.

We assume that the censor's overall computational resources are limited: that they cannot devote much additional RAM or CPU to each TCP stream or packet if they want to route packets conventionally.

We ignore bridge enumeration attacks since they're out-of-scope; down the road we'll write a second analysis document that discusses them.

#### Notation

~~~~~

For each attack below, we give estimated difficulty for attackers and defenders. A "Low" difficulty represents a simple extension of existing technology. A "Medium" difficulty represents a non-trivial development effort, but with a reasonably high probability of success. A "High" difficulty represents an effort that would require significant research and development, with some risk of false starts, blind alleys, and failed attempts.

Some of the ways that attackers can distinguish Tor connections from Firefox-talking-to-Apache connections will also label many other connection types as Tor-like. The severity of this effect in each case is documented below as the false positive rate for the attack technique.

Category I: Single-packet tests

Current firewalls perform most efficiently with rules that apply to a single IP packet: the firewall does not need to remember older packets, meaning it can consider each record individually.

The easiest way to block older versions of Tor, and current versions of many tools, is by rules of this kind. Newer versions of the Tor protocol take pains to mimic a "Firefox 2 to Apache" SSL connection at the byte level, with a goal of making it hard to distinguish our packets from those made in a typical secure browsing session.

OpenSSL-style empty application records

~~~~~

OpenSSL, unlike the NSS SSL library that's used by Firefox, generates empty TLS application records periodically in its stream. (OpenSSL does this in order to avoid a relatively obscure plaintext confirmation attack in CBC-based ciphers. For more information, see this post on the `openssl-users` list: <http://marc.info/?l=openssl-users&m=115654275717293&w=2> )

Tor is probably not the only application that behaves in this way, and killing streams that contain these blank records would not only block Tor, but nearly every other OpenSSL-based application, including a wide variety of commonly used infrastructure tools, like Subversion, Curl, Wget, Irssi, RPM, and PHP. Blocking these tools would shut down a fairly large number of networks and websites.

Nonetheless, a future version of Tor should disable these records anyway, since the attack they're meant to guard against doesn't apply to us.

Attack difficulty:: Low  
False-positive rate:: High  
Defense difficulty:: Very Low

#### No session ID in server hello

~~~~~

The Tor server does not respond to the client with a TLS session ID, whereas most HTTPS servers do support this. This feature does not identify connections as specifically being Tor, but it makes our connections distinguishable from most HTTPS traffic.

We could fix this by including a session identifier whether we support resumption or not. (It might be a good idea to support session resumption for other reasons; see "connection longevity" below.)

Attack difficulty:: Low
False-positive rate:: High
Defense difficulty:: Low

Server certificates

~~~~~

Our current ersatz server certificates have an unusual DN pattern: they include a random hostname as a common name, with no Organization or country field.

There are few enough self-signed certificates of this form in the world that we should add a random organization and a plausible country to our certificates' DNs.

An attacker might also note that the hostname in the certificate does not in fact resolve to the server. While this is not so unusual on the internet, it might be a good idea to try harder to pick a hostname that resolves to the server, if one exists.

Attack difficulty:: Low (Medium for checking for DNS matches occasionally)  
False-positive rate:: High (Medium for checking DNS)  
Defense difficulty:: Low (Medium-Low for making DNS match)

#### Short certificate lifetimes

~~~~~

Tor rotates short-duration TLS certificates far more often than a regular HTTPS server. Though seeing a certificate that will expire in the next 2-24 hours does not prove that a connection is Tor, blocking all connections whose certificates are set to expire soon would not hurt many non-Tor services, and would hurt Tor a lot.

The defense is simple: generate certificates with longer liveness intervals than we actually intend to use. Our frequent TLS key rotation does not actually require that certificate intervals match the TLS key lifespan.

Attack difficulty:: Medium-Low
False-positive rate:: Low
Defense difficulty:: Low

Category II: Per-stream tests

These attacks require the censor to look at more than one packet on a stream, and to remember information between packets. They tend to fit less well into most firewall rule sets.

Renegotiation step

~~~~~

The new Tor handshake's SSL renegotiation phase is visible \_as\_ an SSL negotiation. Many real HTTPS connections do this, but most do not. To get around this, we could switch to a cell-based re-authentication step in a future version of the Tor link protocol.

Attack difficulty:: Medium  
False-positive rate:: High  
Defense difficulty:: Medium

#### Empty application records with Firefox ciphersuites

~~~~~

Although empty application records like those referred to above only

indicate that an application is using OpenSSL, those empty records when on the same stream as the Firefox ciphersuite lists indicate that you're using Tor.

This will be solved trivially when we turn off the OpenSSL empty-records feature.

Attack difficulty:: Medium
False-positive rate:: High
Defense difficulty:: Low

Application record length

~~~~~

Tor tends to form TLS application records shorter than those generally generated by Firefox-Apache connections. Other than the empty records above, the difference is not profound enough to tell Tor from HTTPS by a single record, but in aggregate the difference is not hard to tell, especially on a fast connection where data does not bunch up and create larger records.

We should probably try harder to batch more cells into each TLS record, for efficiency as well as fingerprinting-resistance.

Attack difficulty:: Medium-High  
False-positive rate:: High  
Defense difficulty:: Medium

#### Circuit setup pattern

~~~~~

When a Tor client first builds a circuit, it often follows a fairly predictable pattern of sending a set of CREATE or EXTEND cells and getting CREATED/EXTENDED cells back. These cells are fixed-sized data structures, and are (on uncomplicated connections) sent and received in a pretty fixed pattern.

Tracking these patterns will become trickier over time due to upcoming unrelated changes in Tor's circuit establishment protocols. We could make these patterns even harder to detect by having clients establish non-urgent circuits more slowly, interspersed with small amounts of padding cells.

Attack difficulty:: Medium-High
False-positive rate:: Low
Defense difficulty:: Medium-High

Cell quantization

~~~~~

Once a Tor circuit is established, all of its data is sent in cells of 512 bytes each. If the censor observes a stream over time and realizes that the amount of data sent in a number of application records is usually a multiple of 512, he can conclude that the connection is likely to be Tor.

We can solve this with variable-length cells, or possibly with some amount of connection padding (either at the TLS level or the OR protocol level).

Attack difficulty:: Medium-High  
False-positive rate:: Low  
Defense difficulty:: Medium-High

#### Upload/download ratio and timing

~~~~~

Most HTTP traffic is made up of bursts of relatively short requests from the client, followed by a burst of longer response objects from the server, followed by a delay in which the user reads and reacts to whatever web page they just retrieved. Typical HTTP-over-Tor traffic on an established circuit, however, has a short begin cell, followed by a short connected cell, followed by a short request from the client, followed by response objects from the server. In other words, the begin/connected cycle, in addition to slowing down the connection, also may be a detectable timing pattern to indicate Tor traffic. When the user sends non-HTTP traffic over a Tor connection, the timing pattern will be even more apparent.

For performance reasons, we should already try to remove the state in our protocol where the client is waiting for a CONNECTED cell.

To try to make non-HTTP protocols resemble HTTP, we would need to follow some data padding approach. In the literature so far, these approaches have generally wound up in an arms race: the defender needs to normalize traffic according to all the metrics the attacker can use, whereas the attacker need only find one metric that the defender hasn't normalized. It might be smarter instead to try to use data profiles (like VPN or SSH) that are not expected to be so predictable as HTTP.

Attack difficulty:: Medium-High
False-positive rate:: Low
Defense difficulty:: Medium-High

Connection longevity

~~~~~

A typical browser TLS connection does not last nearly so long as a typical Tor TLS connection. Browsers tend to close their TLS connections fairly quickly, and use session resumption if they want to talk to the same server again.

This attack would have a pretty high false positive rate, since there are other applications that use long-lived TLS connections besides Tor. Nonetheless, we could address it by having Tor clients close TLS connections promptly, and use TLS session resumption when they have something more to say to the same server.

Attack difficulty:: Medium  
False-positive rate:: High  
Defense difficulty:: Medium

#### Broader volume and timing statistics

~~~~~

An attacker with high resources could probably identify --whether in a principled manner, or through a machine-learning process-- patterns of data timing and volume that indicate a Tor connection. This might be

feasible to do on a random sample of traffic, but is too resource-intensive to use for large-scale blocking.

Resisting attacks like this remains an open problem.

Attack difficulty:: High
False-positive rate:: Low-Medium
Defense difficulty:: Very High

Category III: Attack multipliers

Probing

~~~~~

Most of the above detection techniques can be mitigated to the point where their false positive rate is unacceptably high: using them to block Tor would also block much desirable traffic. But if instead of blocking all connections that match these patterns, the censor uses them to identify possible bridges, we'll be in for a bit more work. Once a possible bridge is identified, it's currently pretty easy to confirm whether it is a bridge or not: the censor can just connect to it and see whether it speaks the Tor protocol.

Resisting probing attacks is a separate problem in need of more design work. Generally, our goal would be to make bridges behave like regular HTTPS servers unless the client presents a secret bridge-specific key. The tricky bit would be imitating a HTTPS service that the client would have a plausible reason for connecting to.

Attack difficulty:: Medium-High  
False-positive rate:: Very Low  
Defense difficulty:: High

#### Sampling

~~~~~

Another way to improve the above attacks is to only perform them against a sample of TCP connections in order to identify suspected clients and bridges. Once enough suspect connections had been identified to or from a particular address, the censor could put more computational resources into the traffic of that particular address.

An automated setup like this would take a significant piece of infrastructure investment, but one well within the resources of a technologically sophisticated nation.

To handle this approach, we would have to seek high bridge churn (to have many bridges unblocked at any given time).

Traffic shaping techniques seem our best bet here, but they will result in the kind of arms race described in "broader volume and timing statistics" above, unless there is a major advance in the field of traffic fingerprinting resistance.

Attack difficulty:: Potentially makes any attack above slightly easier.
False-positive rate:: n/a
Defense difficulty:: Medium-high

Category IV: Other

Forced SSL proxies

~~~~~

Some censors (like Burma) sometimes block all SSL traffic that does not pass through a censor-controlled proxy. Tor will not accept such a proxy's certificates as valid, since doing so would render our TLS encryption useless.

Attack difficulty:: Medium in sophistication; High in resources

False-positive rate:: High

Defense difficulty:: High

Conclusion I: The good future (by Nick)

-----

If our observations about detection and fingerprinting of the last version of the Tor protocol are right, the censorious ISPs do not seem to be in the forefront of blocking technology. Instead, the protocol signatures that were used to identify the last protocol seem first to have been isolated by the user community of the "Snort" protocol analysis tool, and then picked up by some commercial vendors, and only much later included by the censors' technology suppliers. If this continues in the future, then we'll have an early warning about good Tor-blocking techniques, so long as the open-source protocol analysis world continues to be interested in Tor detection.

So long as the current pattern continues, it may in fact not be to our advantage to accelerate the censorship/anticensorship arms race by fixing the "low-hanging fruit" in our current detection profile right away. While the censors rely on commercial providers for their R&D, and the commercial providers rely on the broader network protocol analysis community for theirs, we have the dual advantages of advance notice of future censorship techniques (so long as the censors and their providers take their lead from the protocol analysis community), and of being able to deploy defenses with much less overhead than the censors can deploy attacks. If these trends continue, then it may be to our advantage to hold off on deploying easy fixes for the techniques that are easy for attackers to use until it seems that the censors are likely to use them. Since the censors' development cycle is so slow, if we wait for them, they spend time developing attacks that we can easily obsolete. But if we were to deploy the easy fixes ahead of time, the censors would be spared the effort of designing and deploying the corresponding attacks.

Thus, so long as censors continue their current strategy, the best strategy for us may be to begin writing fixes for some of the detectable features, but not release them until the censors' supply chain has spent significant time in producing the corresponding attacks.

Conclusion II: The bad future (by Roger)

-----

One of the primary reasons the filtering tools aren't any good at blocking Tor connections is because the customers of these tools don't really care much about Tor. The modern gizmos from Cisco can be remotely upgraded

to the latest filter-set many times a day, so in fact those "lumbering supply chains" from Conclusion I can provide quite quick turnaround if they care enough.

Worse, even if a quick fix on our side doesn't take much development effort, the upgrade effort from users who can't reach the Tor website might be much higher. The more investment our user community makes in burning TBB to USB keys, making software and instructional DVDs like NGO-in-a-box, etc, the more damage even a trivial filter patch can do.

And while our "secure updater" plan promises to automatically check signatures and fetch upgrades, if it can't reach the update repositories, and its Tor client is blocked from reaching the Tor network, then users will be forced to go through more manual and risky means for upgrade. That means each successful filter patch also introduces a new opportunity to trick users into getting the wrong software. (Or we figure out a good usable way for users to manually feed packages into Thandy for verification -- that seems like it's going to be a good feature to have in any case.)

So that would argue for always keeping a few moves ahead of the filters, since every small step in the arms race creates another opportunity to lose users who don't think it's worthwhile to figure out how to manually upgrade yet again.

Ultimately, winning this arms race will mean staying out of sight of the primary customers of these filtering tools. To understand that more, consider that there are actually two arms races going on right now. The first is between Western corporations and their employees, to keep the workers focused on their jobs, and to prevent them from accidentally visiting malware-infected sites. The second is between censoring countries and their citizens, to keep them from getting out of hand.

Smartfilter and Websense have both changed their marketing pitch in the past few years, to focus more on quickly detecting and blocking sites with malware. Their sales people talk about the infected Superbowl website a few years ago that served malware to millions of computers. No doubt they have a different story when presenting in Saudi Arabia, but my sense is that they've found the "keep our employees from getting infected" feature to be a lot more profitable overall.

So in an ideal world, we'd figure out how to separate these two arms races: if we aren't much of a bother to the Western corporations, which are primarily funding the filter tool development, then the filter companies won't put as much energy into blocking us as they could.

Of course, all of this analysis ignores the fact that China builds and deploys its own filtering tools too. But so far that hasn't really entered into the equations, so there's no reason to worry too much about it quite yet.

**From:** Roger Dingledine  
**To:** [REDACTED]; Sho Ho  
**Subject:** Tor blocking resistance: likely attacks and defenses  
**Date:** Friday, February 20, 2009 1:53:29 AM

---

Hi Ken, Kelly, Sho,

Below is our first stab at analyzing weaknesses in Tor's current TLS footprint. We're not planning to do anything with the results quite yet (see the two conflicting conclusions at the end), but we figured it was a good move to get it started early so we could keep it in mind during further development, and shape it as needed.

Let me know if you have any questions or thoughts (or answers ;). Please check with me before sharing it with anyone -- we're big fans of transparency in general, but this is one of the exceptions.

Thanks,  
--Roger

#### TOR BLOCKING RESISTANCE: LIKELY ATTACKS AND DEFENSES

-----

Analysis by Nick Mathewson, based largely on analysis by Steven Murdoch.  
Current as of February 2009.

(Copyright 2009 The Tor Project. This is sensitive strategic analysis;  
do not circulate it.)

#### Assumptions

~~~~~

We assume a censor who wants to block all or most Tor connections, but who wants to maintain an otherwise usable Internet. We assume that the censor is willing to accept some false positive rate (that is, to block some non-Tor connections), but not a huge one.

We assume that the censor's overall computational resources are limited: that they cannot devote much additional RAM or CPU to each TCP stream or packet if they want to route packets conventionally.

We ignore bridge enumeration attacks since they're out-of-scope; down the road we'll write a second analysis document that discusses them.

Notation

~~~~~

For each attack below, we give estimated difficulty for attackers and defenders. A "Low" difficulty represents a simple extension of existing technology. A "Medium" difficulty represents a non-trivial development effort, but with a reasonably high probability of success. A "High" difficulty represents an effort that would require significant research and development, with some risk of false starts, blind alleys, and failed attempts.

Some of the ways that attackers can distinguish Tor connections from Firefox-talking-to-Apache connections will also label many other connection types as Tor-like. The severity of this effect in each case is documented below as the false positive rate for the attack technique.

#### Category I: Single-packet tests

-----

Current firewalls perform most efficiently with rules that apply to a single IP packet: the firewall does not need to remember older packets, meaning it can consider each record individually.

The easiest way to block older versions of Tor, and current versions of many tools, is by rules of this kind. Newer versions of the Tor protocol take pains to mimic a "Firefox 2 to Apache" SSL connection at the byte level, with a goal of making it hard to distinguish our packets from those made in a typical secure browsing session.

#### OpenSSL-style empty application records

~~~~~

OpenSSL, unlike the NSS SSL library that's used by Firefox, generates empty TLS application records periodically in its stream. (OpenSSL does this in order to avoid a relatively obscure plaintext confirmation attack in CBC-based ciphers. For more information, see this post on the `openssl-users` list:
<http://marc.info/?l=openssl-users&m=115654275717293&w=2>)

Tor is probably not the only application that behaves in this way, and killing streams that contain these blank records would not only block Tor, but nearly every other OpenSSL-based application, including a wide variety of commonly used infrastructure tools, like Subversion, Curl, Wget, Irssi, RPM, and PHP. Blocking these tools would shut down a fairly large number of networks and websites.

Nonetheless, a future version of Tor should disable these records anyway, since the attack they're meant to guard against doesn't apply to us.

Attack difficulty:: Low
False-positive rate:: High
Defense difficulty:: Very Low

No session ID in server hello

~~~~~

The Tor server does not respond to the client with a TLS session ID, whereas most HTTPS servers do support this. This feature does not identify connections as specifically being Tor, but it makes our connections distinguishable from most HTTPS traffic.

We could fix this by including a session identifier whether we support resumption or not. (It might be a good idea to support session resumption for other reasons; see "connection longevity" below.)

Attack difficulty:: Low  
False-positive rate:: High  
Defense difficulty:: Low

#### Server certificates

~~~~~

Our current ersatz server certificates have an unusual DN pattern: they include a random hostname as a common name, with no Organization or country field.

There are few enough self-signed certificates of this form in the world that we should add a random organization and a plausible country to our certificates' DNs.

An attacker might also note that the hostname in the certificate does not in fact resolve to the server. While this is not so unusual on the internet, it might be a good idea to try harder to pick a hostname that resolves to the server, if one exists.

Attack difficulty:: Low (Medium for checking for DNS matches occasionally)
False-positive rate:: High (Medium for checking DNS)
Defense difficulty:: Low (Medium-Low for making DNS match)

Short certificate lifetimes
~~~~~

Tor rotates short-duration TLS certificates far more often than a regular HTTPS server. Though seeing a certificate that will expire in the next 2-24 hours does not prove that a connection is Tor, blocking all connections whose certificates are set to expire soon would not hurt many non-Tor services, and would hurt Tor a lot.

The defense is simple: generate certificates with longer liveness intervals than we actually intend to use. Our frequent TLS key rotation does not actually require that certificate intervals match the TLS key lifespan.

Attack difficulty:: Medium-Low  
False-positive rate:: Low  
Defense difficulty:: Low

Category II: Per-stream tests  
-----

These attacks require the censor to look at more than one packet on a stream, and to remember information between packets. They tend to fit less well into most firewall rule sets.

Renegotiation step  
~~~~~

The new Tor handshake's SSL renegotiation phase is visible `_as_` an SSL negotiation. Many real HTTPS connections do this, but most do not. To get around this, we could switch to a cell-based re-authentication step in a future version of the Tor link protocol.

Attack difficulty:: Medium
False-positive rate:: High
Defense difficulty:: Medium

Empty application records with Firefox ciphersuites
~~~~~

Although empty application records like those referred to above only

indicate that an application is using OpenSSL, those empty records when on the same stream as the Firefox ciphersuite lists indicate that you're using Tor.

This will be solved trivially when we turn off the OpenSSL empty-records feature.

Attack difficulty:: Medium  
False-positive rate:: High  
Defense difficulty:: Low

#### Application record length ~~~~~

Tor tends to form TLS application records shorter than those generally generated by Firefox-Apache connections. Other than the empty records above, the difference is not profound enough to tell Tor from HTTPS by a single record, but in aggregate the difference is not hard to tell, especially on a fast connection where data does not bunch up and create larger records.

We should probably try harder to batch more cells into each TLS record, for efficiency as well as fingerprinting-resistance.

Attack difficulty:: Medium-High  
False-positive rate:: High  
Defense difficulty:: Medium

#### Circuit setup pattern ~~~~~

When a Tor client first builds a circuit, it often follows a fairly predictable pattern of sending a set of CREATE or EXTEND cells and getting CREATED/EXTENDED cells back. These cells are fixed-sized data structures, and are (on uncomplicated connections) sent and received in a pretty fixed pattern.

Tracking these patterns will become trickier over time due to upcoming unrelated changes in Tor's circuit establishment protocols. We could make these patterns even harder to detect by having clients establish non-urgent circuits more slowly, interspersed with small amounts of padding cells.

Attack difficulty:: Medium-High  
False-positive rate:: Low  
Defense difficulty:: Medium-High

#### Cell quantization ~~~~~

Once a Tor circuit is established, all of its data is sent in cells of 512 bytes each. If the censor observes a stream over time and realizes that the amount of data sent in a number of application records is usually a multiple of 512, he can conclude that the connection is likely to be Tor.

We can solve this with variable-length cells, or possibly with some amount of connection padding (either at the TLS level or the OR protocol level).

Attack difficulty:: Medium-High  
False-positive rate:: Low  
Defense difficulty:: Medium-High

#### Upload/download ratio and timing

~~~~~

Most HTTP traffic is made up of bursts of relatively short requests from the client, followed by a burst of longer response objects from the server, followed by a delay in which the user reads and reacts to whatever web page they just retrieved. Typical HTTP-over-Tor traffic on an established circuit, however, has a short begin cell, followed by a short connected cell, followed by a short request from the client, followed by response objects from the server. In other words, the begin/connected cycle, in addition to slowing down the connection, also may be a detectable timing pattern to indicate Tor traffic. When the user sends non-HTTP traffic over a Tor connection, the timing pattern will be even more apparent.

For performance reasons, we should already try to remove the state in our protocol where the client is waiting for a CONNECTED cell.

To try to make non-HTTP protocols resemble HTTP, we would need to follow some data padding approach. In the literature so far, these approaches have generally wound up in an arms race: the defender needs to normalize traffic according to all the metrics the attacker can use, whereas the attacker need only find one metric that the defender hasn't normalized. It might be smarter instead to try to use data profiles (like VPN or SSH) that are not expected to be so predictable as HTTP.

Attack difficulty:: Medium-High
False-positive rate:: Low
Defense difficulty:: Medium-High

Connection longevity

~~~~~

A typical browser TLS connection does not last nearly so long as a typical Tor TLS connection. Browsers tend to close their TLS connections fairly quickly, and use session resumption if they want to talk to the same server again.

This attack would have a pretty high false positive rate, since there are other applications that use long-lived TLS connections besides Tor. Nonetheless, we could address it by having Tor clients close TLS connections promptly, and use TLS session resumption when they have something more to say to the same server.

Attack difficulty:: Medium  
False-positive rate:: High  
Defense difficulty:: Medium

#### Broader volume and timing statistics

~~~~~

An attacker with high resources could probably identify --whether in a principled manner, or through a machine-learning process-- patterns of data timing and volume that indicate a Tor connection. This might be

feasible to do on a random sample of traffic, but is too resource-intensive to use for large-scale blocking.

Resisting attacks like this remains an open problem.

Attack difficulty:: High
False-positive rate:: Low-Medium
Defense difficulty:: Very High

Category III: Attack multipliers

Probing

~~~~~

Most of the above detection techniques can be mitigated to the point where their false positive rate is unacceptably high: using them to block Tor would also block much desirable traffic. But if instead of blocking all connections that match these patterns, the censor uses them to identify possible bridges, we'll be in for a bit more work. Once a possible bridge is identified, it's currently pretty easy to confirm whether it is a bridge or not: the censor can just connect to it and see whether it speaks the Tor protocol.

Resisting probing attacks is a separate problem in need of more design work. Generally, our goal would be to make bridges behave like regular HTTPS servers unless the client presents a secret bridge-specific key. The tricky bit would be imitating a HTTPS service that the client would have a plausible reason for connecting to.

Attack difficulty:: Medium-High  
False-positive rate:: Very Low  
Defense difficulty:: High

#### Sampling

~~~~~

Another way to improve the above attacks is to only perform them against a sample of TCP connections in order to identify suspected clients and bridges. Once enough suspect connections had been identified to or from a particular address, the censor could put more computational resources into the traffic of that particular address.

An automated setup like this would take a significant piece of infrastructure investment, but one well within the resources of a technologically sophisticated nation.

To handle this approach, we would have to seek high bridge churn (to have many bridges unblocked at any given time).

Traffic shaping techniques seem our best bet here, but they will result in the kind of arms race described in "broader volume and timing statistics" above, unless there is a major advance in the field of traffic fingerprinting resistance.

Attack difficulty:: Potentially makes any attack above slightly easier.
False-positive rate:: n/a
Defense difficulty:: Medium-high

Category IV: Other

Forced SSL proxies

~~~~~

Some censors (like Burma) sometimes block all SSL traffic that does not pass through a censor-controlled proxy. Tor will not accept such a proxy's certificates as valid, since doing so would render our TLS encryption useless.

Attack difficulty:: Medium in sophistication; High in resources

False-positive rate:: High

Defense difficulty:: High

Conclusion I: The good future (by Nick)

-----

If our observations about detection and fingerprinting of the last version of the Tor protocol are right, the censorious ISPs do not seem to be in the forefront of blocking technology. Instead, the protocol signatures that were used to identify the last protocol seem first to have been isolated by the user community of the "Snort" protocol analysis tool, and then picked up by some commercial vendors, and only much later included by the censors' technology suppliers. If this continues in the future, then we'll have an early warning about good Tor-blocking techniques, so long as the open-source protocol analysis world continues to be interested in Tor detection.

So long as the current pattern continues, it may in fact not be to our advantage to accelerate the censorship/anticensorship arms race by fixing the "low-hanging fruit" in our current detection profile right away. While the censors rely on commercial providers for their R&D, and the commercial providers rely on the broader network protocol analysis community for theirs, we have the dual advantages of advance notice of future censorship techniques (so long as the censors and their providers take their lead from the protocol analysis community), and of being able to deploy defenses with much less overhead than the censors can deploy attacks. If these trends continue, then it may be to our advantage to hold off on deploying easy fixes for the techniques that are easy for attackers to use until it seems that the censors are likely to use them. Since the censors' development cycle is so slow, if we wait for them, they spend time developing attacks that we can easily obsolete. But if we were to deploy the easy fixes ahead of time, the censors would be spared the effort of designing and deploying the corresponding attacks.

Thus, so long as censors continue their current strategy, the best strategy for us may be to begin writing fixes for some of the detectable features, but not release them until the censors' supply chain has spent significant time in producing the corresponding attacks.

Conclusion II: The bad future (by Roger)

-----

One of the primary reasons the filtering tools aren't any good at blocking Tor connections is because the customers of these tools don't really care much about Tor. The modern gizmos from Cisco can be remotely upgraded

to the latest filter-set many times a day, so in fact those "lumbering supply chains" from Conclusion I can provide quite quick turnaround if they care enough.

Worse, even if a quick fix on our side doesn't take much development effort, the upgrade effort from users who can't reach the Tor website might be much higher. The more investment our user community makes in burning TBB to USB keys, making software and instructional DVDs like NGO-in-a-box, etc, the more damage even a trivial filter patch can do.

And while our "secure updater" plan promises to automatically check signatures and fetch upgrades, if it can't reach the update repositories, and its Tor client is blocked from reaching the Tor network, then users will be forced to go through more manual and risky means for upgrade. That means each successful filter patch also introduces a new opportunity to trick users into getting the wrong software. (Or we figure out a good usable way for users to manually feed packages into Thandy for verification -- that seems like it's going to be a good feature to have in any case.)

So that would argue for always keeping a few moves ahead of the filters, since every small step in the arms race creates another opportunity to lose users who don't think it's worthwhile to figure out how to manually upgrade yet again.

Ultimately, winning this arms race will mean staying out of sight of the primary customers of these filtering tools. To understand that more, consider that there are actually two arms races going on right now. The first is between Western corporations and their employees, to keep the workers focused on their jobs, and to prevent them from accidentally visiting malware-infected sites. The second is between censoring countries and their citizens, to keep them from getting out of hand.

Smartfilter and Websense have both changed their marketing pitch in the past few years, to focus more on quickly detecting and blocking sites with malware. Their sales people talk about the infected Superbowl website a few years ago that served malware to millions of computers. No doubt they have a different story when presenting in Saudi Arabia, but my sense is that they've found the "keep our employees from getting infected" feature to be a lot more profitable overall.

So in an ideal world, we'd figure out how to separate these two arms races: if we aren't much of a bother to the Western corporations, which are primarily funding the filter tool development, then the filter companies won't put as much energy into blocking us as they could.

Of course, all of this analysis ignores the fact that China builds and deploys its own filtering tools too. But so far that hasn't really entered into the equations, so there's no reason to worry too much about it quite yet.

**From:** Roger Dingledine  
**To:** Kelly DeYoe  
**Cc:** Ken Berman  
**Subject:** Tor bridge user testimonials  
**Date:** Wednesday, February 13, 2008 11:21:48 PM

---

Hi Kelly,

I was talking to Chris Walker about "success stories", and I realized that you might also be interested to know that people show up to Tor IRC every week or so saying "help, I'm blocked", and now we have an answer that makes them happy.

Below is a transcript of one such user from mid-January.

--Roger

08-01-17 21:12:09 -!- recluse [~none@dxb-as90008.alshamil.net.ae] has joined #tor  
08-01-17 21:14:16 <recluse> hello  
08-01-17 21:14:42 <DJHasis> yo  
08-01-17 21:15:41 <recluse> can someone help me with my Tor connection problem?  
08-01-17 21:16:04 <DJHasis> tell us what is your problem and maybe someone here could try to do something at it  
08-01-17 21:16:47 <recluse> am currently in UAE  
08-01-17 21:16:59 <DJHasis> cool  
08-01-17 21:17:11 <DJHasis> I've been in Dubai a few months ago  
08-01-17 21:17:20 <recluse> gr8  
08-01-17 21:17:33 <DJHasis> so tell us what is bugging you  
08-01-17 21:17:34 <recluse> so u must have guesses what my problem is  
08-01-17 21:17:45 <recluse> my isp has blocked tor  
08-01-17 21:18:11 <DJHasis> the new version of tor alpha and vidalia have the right fix to that problem  
08-01-17 21:18:29 <recluse> i have installed the latest  
08-01-17 21:18:56 <recluse> and going with the default tor configuration file its not working  
08-01-17 21:18:59 <DJHasis> it gives you a possibility of using a someother tor-node to connect to the tor-network as a bridge  
08-01-17 21:19:07 <DJHasis> yeah, I know  
08-01-17 21:19:29 <recluse> i dont know how to edit it to make it work  
08-01-17 21:19:38 <recluse> if i send u the log  
08-01-17 21:19:47 <recluse> would u b able to help me out?  
08-01-17 21:19:48 <mwenge> are you using vidalia?  
08-01-17 21:19:52 <recluse> yup  
08-01-17 21:20:08 <mwenge> in the config dialog do you see mention of bridges?  
08-01-17 21:20:58 <recluse> torrc?  
08-01-17 21:21:02 <DJHasis> recluse, <http://trac.vidalia-project.net/browser/vidalia/trunk/src/vidalia/help/content/en/bridges.html?format=raw>  
08-01-17 21:21:19 <DJHasis> you need to use a bridge  
08-01-17 21:21:51 <mwenge> you can leave the torrc alone and just use vidalia's config dialog  
08-01-17 21:22:00 <DJHasis> and here are the instructions how to conf vidalia to use a bridge <http://trac.vidalia-project.net/browser/vidalia/trunk/src/vidalia/help/content/en/config.html?format=raw>  
08-01-17 21:22:36 <arma> recluse is in UAE? you don't need bridges. if you have vidalia, click the button in the settings window that says "My ISP blocks connections to the Tor network."  
08-01-17 21:22:39 <arma> you'll need the 0.2.0.15 bundle.  
08-01-17 21:23:43 <recluse> vidalia 0.0.14  
08-01-17 21:23:53 <recluse> oh  
08-01-17 21:24:16 <arma> <https://www.torproject.org/download#Dev>  
08-01-17 21:24:28 <recluse> 0.1.2.18a this is the bundle i have installed  
08-01-17 21:24:44 <arma> recluse: the dev one has the feature you want

08-01-17 21:25:40 <recluse> am using windows  
08-01-17 21:27:08 <recluse> i dont have to compile right?  
08-01-17 21:27:24 <arma> recluse: no, just fetch the windows bundle  
08-01-17 21:28:10 <DJHasis> <http://www.torproject.org/dist/vidalia-bundles/vidalia-bundle-0.2.0.15-alpha-0.0.16.exe>  
08-01-17 21:31:28 <recluse> hoow do i add a bridge  
08-01-17 21:31:32 <arma> you don't need a bridge  
08-01-17 21:31:41 <arma> just go to settings -> network  
08-01-17 21:31:47 <arma> and click 'my isp blocks connections to the tor network'  
08-01-17 21:31:48 <arma> then 'ok'  
08-01-17 21:31:52 <recluse> ok  
08-01-17 21:31:59 <arma> and see what happens. :)  
08-01-17 21:32:50 <recluse> Jan 18 00:30:51.327 [Warning] Received http status code 404 ("Not found") from server '72.165.204.88:9030' while fetching "/tor/keys/fp/OD95B91896E6089AB9A3C6CB56E724CAF898C43F".  
08-01-17 21:33:09 <arma> try stopping tor and then starting it again  
08-01-17 21:33:21 <recluse> ok  
08-01-17 21:33:32 <arma> (through vidalia's interface)  
08-01-17 21:34:34 <recluse> do i have to shutdown privoxy and restart the whole thing?  
08-01-17 21:34:49 <arma> recluse: no, i don't think so  
08-01-17 21:36:02 <recluse> seems like its up  
08-01-17 21:36:07 <arma> recluse: yay.  
08-01-17 21:36:20 <recluse> Jan 18 00:35:04.989 [Notice] Tor has successfully opened a circuit. Looks like client functionality is working.  
08-01-17 21:36:23 <arma> tell your friends (if you feel comfortable doing so)  
08-01-17 21:36:30 <recluse> i will  
08-01-17 21:37:05 <recluse> everyone here is frantically seraching for some solution  
08-01-17 21:38:30 <DJHasis> recluse, where in UAE are you in?  
08-01-17 21:38:34 <DJHasis> Abu Dhabi?  
08-01-17 21:38:52 <recluse> dubai  
08-01-17 21:38:53 <arma> recluse: great. we've been working on a solution for that over the past year. we also are prepared for the next few steps, if they start to crack down more.  
08-01-17 21:39:14 <arma> recluse: but -- i don't think uae has taken any steps to block tor directly. rather, they just contract to an american company called smartfilter, and do whatever smartfilter does.  
08-01-17 21:39:21 <recluse> we dont have access to orkut, flickr over here  
08-01-17 21:39:31 <arma> recluse: you do now. :)  
08-01-17 21:40:23 <DJHasis> that could be possible arma cos when I was in dubai last september, some people did run different kinds of servers on their home pc's  
08-01-17 21:41:26 <recluse> am using firefox with torbutton  
08-01-17 21:41:38 <arma> recluse: the new torbutton that you got with the alpha bundle is much more advanced  
08-01-17 21:41:47 <arma> right-click on it and go to preferences  
08-01-17 21:41:54 <arma> it does all sorts of application-level privacy things for you now too.  
08-01-17 21:42:04 <arma> assuming the installer managed to install it. :)  
08-01-17 21:42:19 <recluse> its installed  
08-01-17 21:42:29 <recluse> then?  
08-01-17 21:42:53 <arma> recluse: then look at the bottom of the window it gives you  
08-01-17 21:43:12 <recluse> The proxy server is refusing connections  
08-01-17 21:43:12 <recluse> Firefox is configured to use a proxy server that is refusing connections.  
08-01-17 21:43:12 <recluse> \* Check the proxy settings to make sure that they are correct.  
08-01-17 21:43:12 <recluse> \* Contact your network administrator to make sure the proxy server is working.  
08-01-17 21:43:12 <recluse> working.  
08-01-17 21:43:44 <recluse> am i disturbing u guys ?  
08-01-17 21:43:58 <arma> recluse: perhaps your privoxy is not running?  
08-01-17 21:44:05 <arma> recluse: did you use tor in uae before, or is this your first time?  
08-01-17 21:44:10 <recluse> yeah  
08-01-17 21:47:25 <recluse> guys , this is gr8  
08-01-17 21:47:33 <recluse> its working perfectly  
08-01-17 21:47:52 <arma> recluse: yay. tell your friends. and figure out how we can tell the right

people without making it too obvious to the wrong people.  
08-01-17 21:48:06 <arma> recluse: please stick around and help us solve this problem for your country, over the next few months. :)  
08-01-17 21:48:15 <recluse> i will  
08-01-17 21:48:24 <recluse> what do u want me to do?  
08-01-17 21:48:50 <arma> are there 'freedom of speech' user groups in your area? :)  
08-01-17 21:49:51 <recluse> how can i support u guys?  
08-01-17 21:51:11 <recluse> hehhehe  
08-01-17 21:51:13 <recluse> ok  
08-01-17 21:51:52 <recluse> i was using hotspot shield before this  
08-01-17 21:53:03 <recluse> thats the weapon of choice for most of the people here. :)  
08-01-17 22:01:47 <recluse> skype is blocked over here  
08-01-17 22:02:21 <lttu> recluse: Where are you?  
08-01-17 22:03:07 <recluse> dubai  
08-01-17 22:03:10 <recluse> UAE  
08-01-17 22:04:06 <lttu> Interesting. What else is blocked?  
08-01-17 22:04:21 <recluse> all voip is blocked  
08-01-17 22:05:10 <recluse> yahoo messenger's callout works though  
[snip]  
08-01-17 22:30:09 <recluse> guys thanks a lot for helping me out  
08-01-17 22:30:32 <recluse> i will join you all later  
08-01-17 22:35:33 -!- recluse [~none@dxb-as90008.alshamil.net.ae] has left #tor []

**From:** [Ken Berman](#)  
**To:** [Roger Dingleline](#); [Kelly DeYoe](#)  
**Subject:** Tor call?  
**Date:** Tuesday, August 22, 2006 8:31:02 AM

---

Do we have a Tor call scheduled for this week? Ken

**From:** Kelly DeYoe  
**To:** ~~Boam, Dinah~~; Bennett, Harrison; ~~McLennan~~; Hiu, Ho; Betty Pruitt  
**Subject:** TOR conference call tomorrow, Thursday 7/6 3 pm EDT / 12 noon PDT  
**Date:** Wednesday, July 05, 2006 6:33:05 PM

---

After Roger initially asking to switch to another time, he indicated his schedule has changed again, and so we are back on at the original time for our conference call tomorrow, Thursday 7/6 at 3pm EDT / 12 noon PDT.

Roger and Bennett, I assume you both will be available at your usual telephone numbers, if either of you needs us to call you at a different number, please let me know.

Although we can discuss any and all relevant topics relating to TOR development, I'd like us to especially focus on the work we've outlined for Bennett, to make sure he has a clear idea of what we need from him to help direct the later anti-censorship development efforts.

-k

**From:** [Kelly DeYoe](#)  
**To:** [Roger Dingledine](#); [Ken Berman](#); [Hui Ho](#)  
**Subject:** TOR contract kickoff meeting, Monday June 5th, 3pm EDT  
**Date:** Friday, June 02, 2006 7:02:37 PM

---

On Monday, June 5th at 3pm EDT we'll have a conference call to get things going with our new contract for TOR development for anti-censorship purposes. Roger, we'll call you from Ken's office at that time.

We had pretty extensive discussions about goals and objectives just to define the scope of work for the contract, but it has been awhile, so let's just plan to review everything and lay out both short-term and long-term goals.

-k



**From:** [Andrew Lewman](#)  
**To:** [Ken Berman](#); [Kelly DeYoe](#)  
**Subject:** Tor Contract  
**Date:** Monday, October 17, 2011 5:01:31 PM

---

Hello Ken and Kelly,

I hope you had a good meeting with Roger and Jake on Friday. Our current contract expires today. I haven't heard back from Diane Sturgis yet, so I thought I'd ask you two directly about the proposal and new contract status.

If there is anything I need to do, I'm all ears.

Thanks.

--

Andrew  
pgp 0x74ED336B

**From:** [Roger Dingleline](#)  
**To:** [Ken Berman](#)  
**Cc:** [Andrew Lewman](#); [Kelly DeYoe](#); [Sho Ho](#); [Jill Moss](#)  
**Subject:** Tor draft statement-of-work  
**Date:** Tuesday, September 06, 2011 7:56:47 AM  
**Attachments:** [2011-09-06-sow-proposed.txt](#)  
[2011-09-06-sow-worksheet.txt](#)

---

Hi Ken,

Attached is a draft statement-of-work. I found it tricky to describe what we will do without knowing how much funding there will be, but hopefully I struck a workable balance between listing things we'd like to do and making it clear that the specific items in each category are "depending on level of funding".

Let me know if you have any feedback and I'm happy to rearrange it as needed.

I also included the worksheet that I used internally to summarize and categorize items from our proposal. I remember from the solicitation that you don't want to hear costs for specific tasks, so feel free to interpret the numbers as rough level-of-effort for each task. The more important part is that I ordered the items within each category based on priority, and put an asterisk by items that I think are most critical to making sure Tor (as an organization and as software) is useful to you this coming year.

--Roger

**From:** Roger Dingledine  
**To:** Kelly DeYoe  
**Cc:** [REDACTED]  
**Subject:** Tor exit relay / bridge questions  
**Date:** Thursday, July 26, 2012 4:22:15 AM

---

Hi Kelly,

We're continuing the process of setting up the exit relays and bridges BBG asked for. You can follow along with the discussions at <https://lists.torproject.org/pipermail/tor-relays/2012-July/001433.html>

The Tor network doesn't have many 100mbit exit relays as it is, and it turns out many of the ones currently running are not in particularly stable situations. So we're starting out focusing on strengthening the hosting situations for the current exit relay operators. That will take us to about 25 exits -- I'll let you know as we develop a plan for the other 100. :)

I have two questions for you:

- The contract says "To ensure diversity of IP addresses, no more than 2 servers may reside in the same /24 IP subset". This constraint totally makes sense for bridges, and in general it's a good idea for public relays because it is related to various diversity metrics, but it turns out there are some cases where it makes less sense. For background, several of our large exit relay operators have found that they can handle abuse much better when they get the ISP to SWIP the netblock to them -- meaning it shows up as theirs on a whois query. The DFRI group in Sweden (a nonprofit set up to run Swedish exit relays) pulled some strings to get a /24 block SWIPed to them, and they're planning to rig things on the backend so they have servers in different cities (and different data centers) yet all the addresses come from their /24, so they can handle abuse complaints themselves. If we want to have them running four exits, how important is it that they go outside their current /24?

- Do you want us to distribute the 75 bridges automatically via our bridgedb service (via https, gmail, etc) or just tell you their addresses privately? There are tradeoffs with each approach (and I'm happy to help you decide), but we should figure it out before we set more of them up. The simple way to decide is: do you have plans to give out their addresses yourself?

(More generally, the scarce resource for bridges is address space, not bandwidth. Most places in the world don't need bridges yet, and in the one place that does (China), I expect seventy-five static fast bridges will get blocked after a while. So I think the longer-term strategy should be to investigate borrowing whole netblocks and redirecting them into bridges en masse. But rather than trying to rework our contract terms in the next few weeks, I figure the easiest approach is to meet the contract terms; then when we've got the exit relay question under control we can start experimenting with more useful bridge solutions.)

Thanks!  
--Roger

**From:** [Roger Dingledine](#)  
**To:** [Kelly DeYoe](#)  
**Cc:** [Ken Berman](#); [Sho Ho](#);  
**Subject:** Tor "exit scanning" up and working (mostly)  
**Date:** Friday, February 20, 2009 6:34:48 AM

---

Hi Kelly,

We've got SoaT (our exit relay scanner, named "Snakes on a Tor" after a movie Mike liked a few years back) a bit more stable now. Mike wrote a README here:

<https://svn.torproject.org/svn/torflow/trunk/NetworkScanners/README.ExitScanning>

You will need to know the super-sekrit url:

[https://ides.fscked.org/transient/soat\\_config.shhhh](https://ides.fscked.org/transient/soat_config.shhhh)

to fetch the config file for it if you want to give it a spin.

The theory for keeping the config file secret is that a malicious exit node could just run the tests against himself and keep tweaking things until the tests don't notice him -- rather like the major antivirus companies become less effective because virus writers test on them before releasing. Sometime in the coming months Mike is going to come up with a "generic" config, so people without the secret config can still try out scanning, and then we can add cute tricks to uncover weird behavior as the arms race demands them. Most of the config file doesn't need to be kept secret after all. We could imagine a future where people can write "scan plugins" that look for some particular behavior, and then share them around; but let's not get ahead of ourselves. :)

It still has quite a few components, so it's non-trivial to set up. I figure once we have a generic config file we can make some simpler step-by-step instructions, and there isn't that much point until then.

My hope for now is to get it pretty quick and stable at finding accidentally misconfigured relays, and we can leave hunting maliciously misconfigured relays for a future date. It looks like false positives are going to be a problem on anything beyond the most rudimentary comparisons, what with the modern trend toward customized active web content.

Mike also started the discussion for how the scanner should interact with the directory authorities to report its results. One day it will automatically tell the directory authority which relays to vote 'BadExit' on, and then when a threshold of voting directories agree, clients will automatically avoid those relays.

<http://archives.seul.org/or/dev/Feb-2009/msg00005.html>

Though getting it automated and reliable enough to let it loose on a directory authority's config is still a long way away.

--Roger

**From:** Roger Dingledine  
**To:** Kelly DeYoe  
**Cc:** Ken Berman; Sho Ho; [REDACTED] (b) (6)  
**Subject:** Tor "exit scanning" up and working (mostly)  
**Date:** Friday, February 20, 2009 6:34:48 AM

---

Hi Kelly,

We've got SoaT (our exit relay scanner, named "Snakes on a Tor" after a movie Mike liked a few years back) a bit more stable now. Mike wrote a README here:

<https://svn.torproject.org/svn/torflow/trunk/NetworkScanners/README.ExitScanning>

You will need to know the super-sekrit url:

[https://ides.fscked.org/transient/soat\\_config.shhhh](https://ides.fscked.org/transient/soat_config.shhhh)

to fetch the config file for it if you want to give it a spin.

The theory for keeping the config file secret is that a malicious exit node could just run the tests against himself and keep tweaking things until the tests don't notice him -- rather like the major antivirus companies become less effective because virus writers test on them before releasing. Sometime in the coming months Mike is going to come up with a "generic" config, so people without the secret config can still try out scanning, and then we can add cute tricks to uncover weird behavior as the arms race demands them. Most of the config file doesn't need to be kept secret after all. We could imagine a future where people can write "scan plugins" that look for some particular behavior, and then share them around; but let's not get ahead of ourselves. :)

It still has quite a few components, so it's non-trivial to set up. I figure once we have a generic config file we can make some simpler step-by-step instructions, and there isn't that much point until then.

My hope for now is to get it pretty quick and stable at finding accidentally misconfigured relays, and we can leave hunting maliciously misconfigured relays for a future date. It looks like false positives are going to be a problem on anything beyond the most rudimentary comparisons, what with the modern trend toward customized active web content.

Mike also started the discussion for how the scanner should interact with the directory authorities to report its results. One day it will automatically tell the directory authority which relays to vote 'BadExit' on, and then when a threshold of voting directories agree, clients will automatically avoid those relays.

<http://archives.seul.org/or/dev/Feb-2009/msg00005.html>

Though getting it automated and reliable enough to let it loose on a directory authority's config is still a long way away.

--Roger

**From:** Roger Dingledine  
**To:** Kelly DeYoe  
**Cc:** Ken Berman; Sho Ho; [REDACTED]  
**Subject:** Tor "exit scanning" up and working (mostly)  
**Date:** Friday, February 20, 2009 6:34:48 AM

---

Hi Kelly,

We've got SoaT (our exit relay scanner, named "Snakes on a Tor" after a movie Mike liked a few years back) a bit more stable now. Mike wrote a README here:

<https://svn.torproject.org/svn/torflow/trunk/NetworkScanners/README.ExitScanning>

You will need to know the super-sekrit url:

[https://ides.fucked.org/transient/soat\\_config.shhhh](https://ides.fucked.org/transient/soat_config.shhhh)

to fetch the config file for it if you want to give it a spin.

The theory for keeping the config file secret is that a malicious exit node could just run the tests against himself and keep tweaking things until the tests don't notice him -- rather like the major antivirus companies become less effective because virus writers test on them before releasing. Sometime in the coming months Mike is going to come up with a "generic" config, so people without the secret config can still try out scanning, and then we can add cute tricks to uncover weird behavior as the arms race demands them. Most of the config file doesn't need to be kept secret after all. We could imagine a future where people can write "scan plugins" that look for some particular behavior, and then share them around; but let's not get ahead of ourselves. :)

It still has quite a few components, so it's non-trivial to set up. I figure once we have a generic config file we can make some simpler step-by-step instructions, and there isn't that much point until then.

My hope for now is to get it pretty quick and stable at finding accidentally misconfigured relays, and we can leave hunting maliciously misconfigured relays for a future date. It looks like false positives are going to be a problem on anything beyond the most rudimentary comparisons, what with the modern trend toward customized active web content.

Mike also started the discussion for how the scanner should interact with the directory authorities to report its results. One day it will automatically tell the directory authority which relays to vote 'BadExit' on, and then when a threshold of voting directories agree, clients will automatically avoid those relays.

<http://archives.seul.org/or/dev/Feb-2009/msg00005.html>

Though getting it automated and reliable enough to let it loose on a directory authority's config is still a long way away.

--Roger

**From:** [Andrew Lewman](#)  
**To:** [Ken Berman](#); [Kelly DeYoe](#)  
**Cc:** [Dingledine, Roger](#)  
**Subject:** Tor February 2009 Report  
**Date:** Tuesday, March 10, 2009 9:01:58 AM  
**Attachments:** [tor-bbg\\_feb09-report.doc](#)



---

Hello Ken and Kelly,

Attached is our February 2009 report. As always, feel free to ask questions.

--

Andrew Lewman  
The Tor Project

  
pgp 0x31B0974B  


Website: <https://torproject.org/>  
Blog: <https://blog.torproject.org/>  
Identicat/Twitter: torproject

**From:** [Andrew Lewman](#)  
**To:** [Ken Berman](#); [Kelly DeYoe](#)  
**Cc:** [Dingledine, Roger](#)  
**Subject:** Tor February 2009 Report  
**Date:** Tuesday, March 10, 2009 9:01:58 AM  
**Attachments:** [tor-bbg-feb09-report.doc](#)



---

Hello Ken and Kelly,

Attached is our February 2009 report. As always, feel free to ask questions.

--

Andrew Lewman  
The Tor Project

  
pgp 0x31B0974B  


Website: <https://torproject.org/>  
Blog: <https://blog.torproject.org/>  
Identica/Twitter: torproject



**From:** [Roger Dingleline](#)  
**To:** [Kelly DeYoe](#)  
**Cc:** [Ken Berman](#)  
**Subject:** Tor IBB update for Dec coming soon  
**Date:** Thursday, January 10, 2008 10:31:58 PM

---

Hi folks,

I've been spending today working on getting the 0.2.0.x releases closer to becoming an actual release candidate. So I have some notes for the Dec report, but I figure it'll be better if I send it tomorrow. Hopefully that will be soon enough. :)

Shall we schedule our monthly conf call sometime? Next week I'm in Indiana but my cell phone should work just as well there. The week after I'm available most days too.

Thanks,  
--Roger

**From:** [Roger Dingleline](#)  
**To:** [Ken Berman](#); [Kelly DeYoe](#)  
**Cc:** [REDACTED] (b) (6)  
**Subject:** Tor IBB update for Dec  
**Date:** Sunday, January 13, 2008 1:36:49 AM  
**Attachments:** [tor-dec07.doc](#)

---

Hi folks,

Attached is the Dec update for our progress. It should reflect most of the things we talked about in our mid-Dec meeting, plus some other progress beyond that, e.g. the new Torbutton-dev design docs.

Thanks,  
--Roger

**From:** [Roger Dingledine](#)  
**To:** [Ken Berman](#); [Kelly DeYoe](#)  
**Cc:** [REDACTED]  
**Subject:** Tor IBB update for Feb  
**Date:** Tuesday, March 11, 2008 4:19:17 AM  
**Attachments:** [tor-feb08.doc](#)

---

Hi folks,

Attached is the Feb update for our progress.

In newer news, there's a new set of Tor Browser Bundle instructions with clearer explanations:

<https://www.torproject.org/torbrowser/>

and it also comes in German, Italian, Polish, and Russian:

<https://www.torproject.org/torbrowser/index.html.ru>

We also had a strategy meeting recently, and we decided that we weren't gaining anything by leaving our "encrypted directory request" feature off by default now that our recent release has enabled our "TLS normalization" features. Indeed, most of the folks in our target countries were figuring that Tor had simply stopped working, and few of them were coming to us for help, so they weren't able to hear about the "My ISP blocks connections to the Tor network" option in Vidalia. By turning this feature on by default, we're taking the next step in the arms race (and encouraging it to require further steps), but that's better than remaining one step behind.

This decision means we should put out another release candidate to see what breaks when we turn encrypted directory fetches on by default. (Hopefully nothing will break, of course.) We'll have that release candidate out in the next few days. Then we'll make a new Tor Browser Bundle snapshot with all the new versions.

Then we can point your in-house Russian translators to the Tor Browser Bundle Russian page, and see what happens from there. How much of a blurb should we prepare for them separately from that page? I can imagine such topics as "this is why IBB cares about this", "this is who we're planning to show it to next", "this is the sort of feedback we'd like", "here's what Tor is good for and here's what it isn't so good for", etc.

And lastly, shall we organize a conference call again sometime? This week is looking pretty rough for me, but next week is mostly empty.

Thanks!  
--Roger

**From:** Roger Dingleline  
**To:** Ken Berman; Kelly DeYoe  
**Cc:** [REDACTED]  
**Subject:** Tor IBB update for Jan  
**Date:** Tuesday, February 12, 2008 4:07:11 AM  
**Attachments:** [tor-ian08.doc](#)

---

Hi folks,

Attached is the Jan update for our progress.

Now on to the other items on that task list I sent earlier today. :)

Thanks!  
--Roger

**From:** [Roger Dingleline](#)  
**To:** [Ken Berman](#); [Kelly DeYoe](#)  
**Cc:** [REDACTED]  
**Subject:** Tor IBB update for March  
**Date:** Friday, April 11, 2008 7:06:25 AM  
**Attachments:** [tor-mar08.doc](#)

---

Hi folks,

Attached is the March update for our progress.

I'm curious to hear if anything further has happened with Vlad or the other Russian folks?

Thanks,  
--Roger

**From:** [Roger Dingleline](#)  
**To:** [Kelly DeYoe](#); [Ken Berman](#)  
**Cc:** (b) (6)  
**Subject:** Tor IBB update for Nov  
**Date:** Tuesday, December 11, 2007 12:02:28 AM  
**Attachments:** [tor-nov07.doc](#)

---

Hi folks,

Attached is the Nov update for our progress. It's basically more set-up for the December push, which I summarized in my email last week.

Should we schedule a conference call for this month, or should we just plow ahead and we'll get the summaries at the end of the month? I'm ok either way. :)

Thanks!  
--Roger

**From:** [Roger Dingledine](#)  
**To:** [Kelly DeYoe](#); [Ken Berman](#)  
**Cc:** [REDACTED]  
**Subject:** Tor IBB update for Oct  
**Date:** Monday, November 12, 2007 3:23:25 AM  
**Attachments:** [tor-oct07.doc](#)

---

Hi folks,

Attached is the Oct update for our progress. Progress is basically what you saw last week. In the next week or so we're hoping to deploy the new (more subtle) TLS handshake, and then we can focus on the public bridge distribution strategies and on a safe USB Windows Tor image.

Should we schedule a conference call for this month, or did we take care of that last week?

Thanks!  
--Roger

**From:** [Roger Dingledine](#)  
**To:** [Kelly DeYoe](#); [Ken Berman](#)  
**Cc:** [redacted] (b) (6)  
**Subject:** Tor IBB update for Sept  
**Date:** Thursday, October 11, 2007 3:32:29 AM  
**Attachments:** [tor-sep07.doc](#)

---

Hi folks,

Attached is the Sept update for our progress. Main updates are that we got bridges and bridge authorities more stable again (turns out we introduced some bugs while adding other features); we've started a document brainstorming details on how to get our TLS handshake to not stand out; and we've made a lot of progress on the new Torbutton version that promises to help ordinary users stay secure from application-level attacks while using Tor.

I'm up for a conference call pretty much whenever (especially if it's after noon rather than in the morning. :)

Also, my day-for-dropping-by-IBB is looking like it should be Nov 2. Does that still work for you two? Do you prefer morning or afternoon (or heck, both)? With luck I'll show you our snazzy Vidalia bridge interface actually working at that point.

--Roger



**From:** Ken Berman  
**To:** Roger Dingledine; Andrew Lewman; Jacob Appelbaum; (b) (6)  
**Cc:** Eric Howard; Kelly DeYoe; Kyle Noori  
**Subject:** Tor info  
**Date:** Wednesday, November 23, 2011 10:26:18 AM

---

Y'all know about this? Ken

U.S. CERT

#### Recent Incidents Highlight Threats to Tor Proxy System

November 22, 2011 12:13:38 PM, Network Conflict, Intel-496857, Version: 1  
Key Points

- \* Two recent incidents highlight methods for compromising the information of users of The Onion Router (Tor) proxy system.
- \* Anonymous-affiliated actors described how they disseminated a "honeypotted" TorButton and encouraged visitors of child pornography sites to download it. In a separate event, French security researcher Eric Filiol described an attack involving infecting existing Tor nodes with "cryptographic backdoors."
- \* These events could inspire repressive regimes and other entities to carry out similar operations against the Tor network, which political dissidents and other individuals seeking anonymity often use.

#### Overview

Two recent developments highlight methods for compromising the information of users of The Onion Router (Tor), a popular proxy network used by political dissidents, hacktivists and other actors. Actors claiming an affiliation with the Anonymous community targeted consumers of online child pornography by convincing them to download a fraudulent "TorButton" that recorded their traffic and IP addresses. Additionally, French security researcher Eric Filiol gave a presentation at a late October 2011 security conference describing a large-scale attack involving the compromise of existing Tor nodes with "cryptographic backdoors." Both developments could prove instructional for repressive regimes and other entities seeking to identify political dissidents or other actors using Tor.

#### Threat Detail

\*Operation Darknet\*

"Operation Darknet" logo  
(iSIGHT Partners)

Beginning in late October 2011, actors claiming to be affiliated with the Anonymous hacktivism community began an operation targeting Lolita City and The Hidden Wiki, two websites associated with child pornography. Along with denial-of-service (DoS) attacks against the two sites, they also engaged in an operation they called "Paw Printing,"

which is designed to identify visitors to the two child pornography sites. According to the attackers, the operation entailed:

1. Monitoring two IRC channels used by Tor developers to determine when a new security update was scheduled (it ended up being scheduled for Oct. 27, 2011).
2. Creating "The Honey Pawt," a Trojaned TorButton (i.e., a Firefox plugin used to access Tor), and enlisting an insider at The Mozilla Network to authorize a developer signer certificate for the plugin.
3. Placing a link to download "The Honey Pawt" at the "HARD CANDY" section (the section dedicated to child pornography) of The Hidden Wiki, along with text claiming that it was a security upgrade for TorButton. "The Honey Pawt" would replace users' existing TorButtons.
4. Once installed, "The Honey Pawt" would funnel originating traffic to the attackers' forensic logger (dubbed "Whiny da Pedo"), which would log the victimized users' IPs and destinations before re-routing the traffic through the attackers' local Tor Bridge.

The hacktivists claim to have carried out the operation over 24 hours, from Oct. 27-28, 2011, before resuming DoS attacks against the two sites. As a result, they claimed to have identified 190 unique IPs and users, publicly posting the results online along with a Google map mashup showing their locations.

The idea of exploiting Tor to expose traders in child pornography could have been inspired by a 2007 proposal by HD Moore, who runs the Metasploit Project, to use Tor to reveal the physical location of dealers in child pornography. However, Moore's plan was different, as it involved using a specially patched server to "listen" for child pornography-related keywords, and then injecting HTML code into the response that would link to a decloaking engine that would reveal users' personal information.

\*Upcoming Conference Describes Tor Attacks\*

Another, large-scale method for compromising Tor users was described by French security researcher Lt. Col. Eric Filiol at the Oct. 29, 2010, H2HC conference in São Paulo, Brazil. Filiol's method involves:

1. Infecting Tor nodes with a "dynamic cryptographic backdoor" that would covertly compromise their cryptographic functions and
2. DDoSing the other Tor nodes to force users to visit the compromised nodes.

Although this attack would necessarily involve compromising individual machines, Filiol claimed that 30 percent of computers running Tor were vulnerable to known Windows vulnerabilities. Filiol stated that he successfully tested the attack against a network architecture simulating the Tor network and claimed that he plans to release a finalized paper on the subject in late November 2011. iSIGHT Partners has written on Lt. Col. Filiol's other research (for more information, see iSIGHT Partners. /ThreatScape Network Conflict/, "Presentation on 'Office Documents as New Weapons of Cyberwarfare' Likely Gives Insight into Longstanding French CNO Capabilities and Intent." Intel-310548. Nov. 29, 2010).

**\*Outlook\***

For several years, Western free-speech organizations have advocated that political dissidents in countries with repressive regimes (particularly Iran, Burma, Syria and India) use Tor to mask their activities from their governments. However, the aforementioned two incidents demonstrate how Tor is by no means a foolproof solution and is potentially vulnerable to large-scale and targeted attacks.

Filiol's presentation explicitly posits "a non-democratic country that wants to monitor all its political opponents (inside and outside the country)." However, an attack similar to "Operation Darknet" (i.e., creating and disseminating a backdoored Tor client) would likely be easier to perform, as it relies largely on social engineering rather than identifying and compromising individual machines. Although it would not be as large scale as the attack described by Filiol, it could still compromise a sufficient number of individuals to identify a dissident network, especially when combined with examination of social networks and other data-mining activities.

Tags: Cryptography

<https://portal.isightpartners.com/tfc/portal/search/tags/ti:Cryptography>,

English

<https://portal.isightpartners.com/tfc/portal/search/tags/ti:English>,

Fraud Investigations (3)

[>](https://portal.isightpartners.com/tfc/portal/search/tags/ti:Fraud%20Investigations%20(3)),

>

Intel Staff (4)

[>](https://portal.isightpartners.com/tfc/portal/search/tags/ti:Intel%20Staff%20(4)),

Non-state Hacktivist

[>](https://portal.isightpartners.com/tfc/portal/search/tags/ti:Non-state%20Hacktivist),

Security Operations (2)

[>](https://portal.isightpartners.com/tfc/portal/search/tags/ti:Security%20Operations%20(2))

>

**Technical Tags**

THREAT CATEGORIES: Cryptography;Non-state Hacktivist

**OTHER TAGS**

Language: English

Intended Audience: Security Operations (2);Fraud Investigations

(3);Intel Staff (4)

© Copyright 2011 iSIGHT Partners All rights reserved.

Rate This Report

[>](https://portal.isightpartners.com/tfc/portal/reports/docidviewer/Intel-496857)

| Contact an Analyst

[>](mailto: (b) (6) subject=Regarding Intel-496857)

(Recent Incidents Highlight Threats to Tor Proxy System)>

(back to table of contents...) <#TOC>

-----