angr

```
>>> import angr
>>> proj = angr.Project('./fauxware-amd64')
>>> cfg = proj.analyses.CFG(); cfg.function_manager.functions
{4195600L: <Function sub_400510 (0x400510)>,
 4195616L: <Function sub_400520 (0x400520)>,
 4195632L: <Function sub_400530 (0x400530)>,
 4195648L: <Function sub_400540 (0x400540)>,
 4195664L: <Function sub_400550 (0x400550)>,
 4195680L: <Function sub_400560 (0x400560)>,
 4195696L: <Function sub_400570 (0x400570)>,
 4195712L: <Function _start (0x400580)>,
 4195940L: <Function authenticate (0x400664)>,
 4196077L: <Function accepted (0x4006ed)>,
 4196093L: <Function rejected (0x4006fd)>,
 4196125L: <Function main (0x40071d)>}
>>> ex = proj.surveyors.Explorer(find=0x4006ed).run()
>>> ex.found[0].state.posix.dumps(0)
'\x00\x00\x00\x00\x00\x00\x00\x00\x00SOSNEAKY\x00'
```

What is angr?

angr is a framework for analyzing binaries. It focuses on both static and dynamic symbolic ("concolic") analysis, making it applicable to a variety of tasks.

What's it made of?

angr is made up of several subprojects, all of which are open-source!

- an executable and library loader, CLE
- a library describing various architectures, archinfo
- a Python wrapper around the binary code lifter VEX, PyVEX
- a VEX simulation engine, SimuVEX
- a data backend to abstract away differences between static and symbolic domains, Claripy
- the full-program analysis suite itself, angr
- a GUI for some features of angr, angr-management

How has it been used academically?

If you have used angr or its sub-components in research, please cite the paper that it was developed for:

```
@article{shoshitaishvili2015firmalice,
    title={Firmalice - Automatic Detection of Authentication Bypass Vuln
           in Binary Firmware},
    author={Shoshitaishvili, Yan and Wang, Ruoyu and Hauser, Christophe
            and Kruegel, Christopher and Vigna, Giovanni},
    booktitle={NDSS},
    year={2015}
  }
```

You can also *read* the paper here!

And non-academically?

angr was one of the underpinnings of Shellphish's Cyber Reasoning System for the DARPA Cyber Grand Challenge, enabling them to qualify for the CGC finals! Shellphish has also used angr in many CTFs!

Whom can I contact?

If you have questions with a subcomponent of angr, please open an issue on github (or send us a pull request!). If you have questions or comments, drop us a line at the mailing list at *angr AT lists.cs.ucsb.edu* or hang out on the IRC channel (**#angr** on freenode).

Who works on angr?

angr is worked on by several researchers in the Computer Security Lab at UC Santa Barbara. Major contributers (arbirtrarily, 1000+ lines of code!) include:

- Yan Shoshitaishvili
- Ruoyu (Fish) Wang
- Andrew Dutcher
- Christophe Hauser
- John Grosen
- Chris Salls
- Nick Stephens

angr owes its existence to research sponsored by DARPA under agreement number N66001-13-2-4039!

How do I learn?

There are a few resources you can use to help you get up to speed!

- The documentation repository, including ready-to-run examples.
- The presentations from angr's debut at DEFCON 23 (video) and Blackhat 2015 (video).
- Presentations discussing Shellphish's use of angr in the DARPA Cyber Grand Challenge at

HITCON ENT 2015, HITCON CMT 2015, and 32C3.

How can I help?

There are many ways to participate! Here are some ideas:

- Report bugs. We know they exist, but it's not always clear where they are!
- Write documentation. An analysis system like angr can be a bit overwhelming. If you're using it, and could send a pull request for our documentation repo, we would be eternally grateful! This includes examples -- if you use angr for something cool, send us a pull request with an example!
- Implement more environment support. We use the concept of "function summaries" in angr to model the environment of operating systems (i.e., the effects of their system calls) and library functions. Extending this would be greatly helpful in increasing angr's utility. These function summaries can be found here.

*The angr workshop at 32c3 has been scheduled for a second day! If you would like to learn about the basics of using angr, please attend this event in **room 13, Wednesday at 14:30**.*