ADD / XOR / ROL

A blog about reverse engineering, mathematics, politricks and some more ...

Tuesday, December 22, 2015

Open-Source BinNavi ... and fREedom

One of the cool things that my former zynamics colleagues (now at Google) did was the open-sourcing of BinNavi - a tool that I used to blog about quite frequently in the old days (here for example when it came to debugging old ScreenOS devices, or here for much more - kernel debugging, REIL etc.).

BinNavi is a GUI / IDE for performing multi-user reverse engineering, debugging, and code analysis. BinNavi allows the interactive exploration and annotation of disassemblies, displayed as browsable, clickable, and searchable graphs - based on a disassembly read from a PostgreSQL database, which can, in theory, be written by any other engine.

Writing UIs is hard work, and while there are many very impressive open-source reverse engineering tools around (Radare comes to mind first, but there are many others), the UI is often not very pretty - or convenient. My hope is that BinNavi can become the "default UI" to a plethora of open-source reverse engineering tools, and grow to realize it's full potential as "the open-source reverse engineering IDE".

One of the biggest obstacles to BinNavi becoming more widely adopted is the fact that IDA is the only "data source" for BinNavi - e.g. while BinNavi is FOSS, somebody that wishes to start reverse engineering still needs IDA to fill the Postgres database with disassembly.

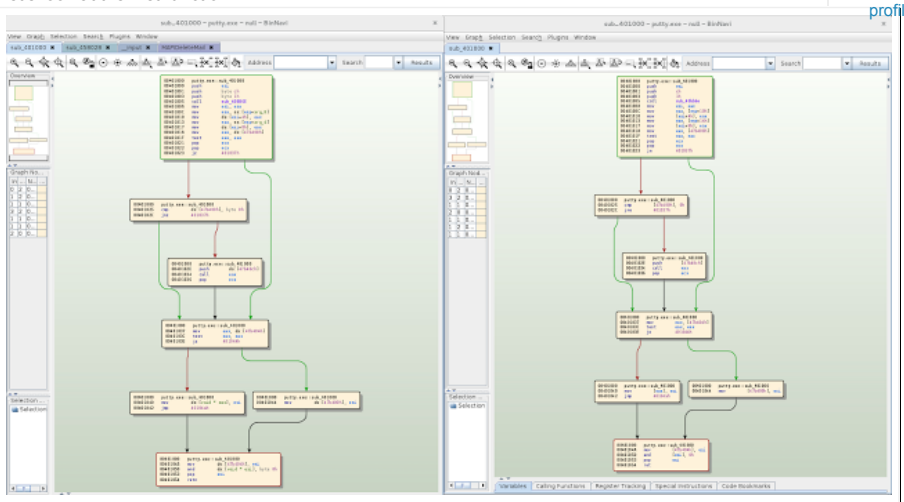
To remedy this situation, Dave Aitel put up a contest: Anybody that either builds a Capstone-to-BinNavi-SQL-bridge or that adds decompilation as a feature to BinNavi gets free tickets to INFILTRATE 2016.

Last week Chris Eagle published fREedom, a Python-based tool to disassemble x86 and x86_64 programs in the form of PE32, PE32+, and ELF files. This is pretty awesome - because it means that BinNavi moves much closer to being usable without any non-free tools.

In this blog post, I will post some first impressions, observations, and screenshots of fREedom in action.

My first test file is putty.exe (91b21fffe934d856c43e35a388c78fccce7471ea) - a relatively small Win32 PE file, with about ~1800 functions when disassembled in IDA.

Let's look at the first function:



Left: IDA's disassembly output. Right: fREedom's disassembly output

So disassembly, CFG building etc. has worked nicely. Multi-user commenting works as expected, as does translation to REIL. Callgraph browsing works, too:

About Me

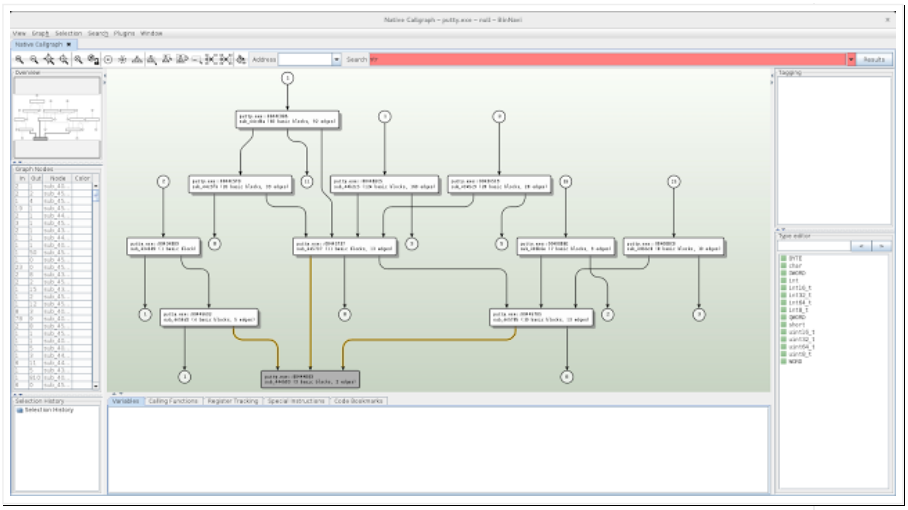
halvar.flake

I like simple things. And complex things. And drinking beer with people like Fyodor Yarochkin. I like South America. And some parts of Asia, specifically Kuala Lumpur. I like French. I like Spanish. I'd like to like more languages.

View my complete profile

Links

- zynamics (now Google)
- Ero's Blog
- Ilfak's Blog
- SP's blog
- SinFest
- OpenRCE
- OffensiveComputing
- Mark Dowd / John McDonald blog



The great thing about having fREedom to start from is that further improvements can be incremental and layered - people have something good to work from now :-) So what is missing / needs to come next?

1. fREedom: Function entry point recognition is still relatively poor - out of the ~1800 functions that IDA recognizes in putty, only 430 or so are found. This seems like an excellent target for one of those classical "using Python and some machine learning to do XYZ" blog posts.
2. fREedom: The CFG reconstruction and disassembly needs to be put through it's paces on big and harder executables.
3. BinNavi: Stack frame information should be reconstructed - but not by fREedom, but within BinNavi (and via REIL). This will require digging into (and documenting) the powerful-but-obscure type system design.
4. BinNavi: There has been some bitrot in many areas of BinNavi since 2011 - platforms change, systems change, and there are quite some areas that are somewhat broken or need updating (for example debugging on x64 etc.). Time to brush off the dust :-)

Personally, I am both super happy and pretty psyched about fREedom + BinNavi, and I hope that the two can be fully integrated so that BinNavi always has fREedom as default disassembly backend.

Posted by halvar.flake at 2:38 PM

No comments:

Post a Comment

Home Older Post

Subscribe to: Post Comments (Atom)