

Bro stuff

Justin Azoff

Aug 4, 2015

try.bro.org on github

The screenshot shows the GitHub repository page for `try.bro.org`. At the top, the repository name is displayed as `bro / try-bro`. To the right, there are interaction buttons: `Unwatch - 12`, `Star 4`, and `Fork 0`. Below this, the text `Code for try.bro.org. — Edit` is visible. A progress bar shows the repository's activity: `157 commits`, `1 branch`, `0 releases`, and `0 contributors`. The current branch is `master`. The main content area displays the file structure:

- `simplify log file handling` (latest commit `e99caf0708`)
 - `client` (Simplifications, 4 months ago)
 - `manager` (simplify log file handling, 3 months ago)
 - `.gitignore` (ignore, a year ago)
 - `trybro.conf` (update image name, 4 months ago)
 - `trybro.service` (initial systemd unit file, 3 months ago)

At the bottom of the file list, there is a prompt: `Help people interested in this repository understand your project by adding a README!` with a green `Add a README` button.

On the right side, a sidebar contains navigation options: `Code`, `Issues 0`, `Pull requests 0`, `Wiki`, `Pulse`, and `Graphs`.

Figure : try.bro on github

Bro Dockerfiles on github

bro / bro-docker

Unwatch - 12 Star 2 Fork 1

Docker files for building Bro. — Edit

26 commits 1 branch 0 releases 1 contributor

Branch: master - bro-docker / +

update url for 2.3.2 tarball

Justin Azoff authored on Jun 16 latest commit f6f2408d24

common	fixes for 1.5	6 months ago
.gitignore	add gitignore	6 months ago
1.5.Dockerfile	set PATH in the dockerfile	5 months ago
2.1.Dockerfile	set PATH in the dockerfile	5 months ago
2.2.Dockerfile	update for openssl issue	2 months ago
2.3.1.Dockerfile	update for openssl issue	2 months ago
2.3.2.Dockerfile	update url for 2.3.2 tarball	2 months ago
2.4.Dockerfile	update for openssl issue	2 months ago

Code

Issues 0

Pull requests 0

Wiki

Pulse

Graphs

Settings

Figure : try.bro on github

- ▶ CORS is enabled on API endpoints.
- ▶ `http://www.ncsa.illinois.edu/People/jazoff/bro.html`

BHR

New implementation of a BlackHole Router with bro integration.

- ▶ <https://github.com/JustinAzoff/bhr-site>
- ▶ <https://github.com/JustinAzoff/bhr-bro>

Use:

```
@load ./bhr-bro
```

```
redef BHR::block_types += {  
    Scan::Port_Scan,  
    Scan::Address_Scan,  
};
```

- ▶ Let's fuzz bro!

- ▶ Let's fuzz bro!
- ▶ Basic approach yields 1.8 executions/second - too slow
- ▶ Tabled for a while.

Fuzz take 2

New features to the rescue:

afl-fuzz persistent mode

<http://lcamtuf.blogspot.com/2015/06/new-in-afl-persistent-mode.html>

Hack up bro and try again:

```
src/Net.cc | 12 ++++++-----  
src/main.cc | 25 ++++++-----  
2 files changed, 29 insertions(+), 8 deletions(-)
```


Fuzz take 2

New features to the rescue:

afl-fuzz persistent mode

<http://lcamtuf.blogspot.com/2015/06/new-in-afl-persistent-mode.html>

Hack up bro and try again:

```
src/Net.cc | 12 ++++++-----  
src/main.cc | 25 ++++++-----  
2 files changed, 29 insertions(+), 8 deletions(-)
```

Result: 1000+ executions/second.

Fuzz take 2

New features to the rescue:

afl-fuzz persistent mode

<http://lcamtuf.blogspot.com/2015/06/new-in-afl-persistent-mode.html>

Hack up bro and try again:

```
src/Net.cc | 12 ++++++-----  
src/main.cc | 25 ++++++-----  
2 files changed, 29 insertions(+), 8 deletions(-)
```

Result: 1000+ executions/second.

But no crashes :(

TODO: Fuzz take 3

Need to build a test bro binary that bypasses libpcap and basic tcp reassembly to feed data directly into analyzers.

Fuzz Detour

Maybe I should try something simpler like bro-cut.

Fuzz Detour

Maybe I should try something simpler like bro-cut.

```
american fuzzy lop 1.83b (buggy_bro-cut)

┌── process timing ───────────────────────────────────────────────────┐ overall results ───────────────────────────────────────────────────┐
│   run time : 0 days, 0 hrs, 19 min, 9 sec   │ cycles done : 1 │
│ last new path : 0 days, 0 hrs, 0 min, 22 sec │ total paths : 100 │
│ last uniq crash : 0 days, 0 hrs, 0 min, 20 sec │ uniq crashes : 2 │
│ last uniq hang : 0 days, 0 hrs, 8 min, 51 sec │  uniq hangs : 13 │
├── cycle progress ───────────────────────────────────────────────────┐ map coverage ───────────────────────────────────────────────────┐
│ now processing : 98* (98.00%) │ map density : 138 (0.21%) │
│ paths timed out : 0 (0.00%) │ count coverage : 4.69 bits/tuple │
├── stage progress ───────────────────────────────────────────────────┐ findings in depth ───────────────────────────────────────────────────┐
│ now trying : havoc │ favored paths : 8 (8.00%) │
│ stage execs : 29.9k/30.0k (99.50%) │ new edges on : 8 (8.00%) │
│ total execs : 1.21M │ total crashes : 2 (2 unique) │
│ exec speed : 1290/sec │ total hangs : 10.2k (13 unique) │
├── fuzzing strategy yields ───────────────────────────────────────────┐ path geometry ───────────────────────────────────────────────────┐
│ bit flips : 11/27.2k, 0/27.2k, 0/27.2k │ levels : 5 │
│ byte flips : 0/3405, 0/3331, 1/3279 │ pending : 74 │
│ arithmetics : 10/187k, 0/41.3k, 0/142 │ pend fav : 0 │
│ known ints : 0/17.5k, 0/92.9k, 0/144k │ own finds : 99 │
│ dictionary : 0/0, 0/0, 13/22.5k │ imported : n/a │
│   havoc : 53/578k, 0/0 │ variable : 0 │
│   trim : 50.12%/1871, 1.39% │ ───────────────────────────────────────────────────┐
└──────────────────────────────────────────────────┘ [cpu: 50%]
```

Figure : bro-cut fuzz crashes

bro-cut bugs

Failed conversion of out of range or invalid timestamps

```
#fields ts
#types time
777777777777777777
```

File header contains a missing or null separator

```
#separator
#fields a
hi
```

```
#separator \x00
#fields a
hi
```

bro-cut bugs ==2

```
american fuzzy lop 1.83b (bro-cut)

┌── process timing ───────────────────────────────────────────────────┐ overall results ───────────────────────────────────────────────────┐
│ run time : 0 days, 13 hrs, 39 min, 44 sec │ cycles done : 210 │
│ last new path : 0 days, 12 hrs, 52 min, 34 sec │ total paths : 160 │
│ last uniq crash : none seen yet │ uniq crashes : 0 │
│ last uniq hang : 0 days, 0 hrs, 31 min, 47 sec │ uniq hangs : 9 │
├── cycle progress ───────────────────────────────────────────────────┐ map coverage ───────────────────────────────────────────────────┐
│ now processing : 1* (0.62%) │ map density : 141 (0.22%) │
│ paths timed out : 0 (0.00%) │ count coverage : 6.05 bits/tuple │
├── stage progress ───────────────────────────────────────────────────┐ findings in depth ───────────────────────────────────────────────────┐
│ now trying : splice 11 │ favored paths : 9 (5.62%) │
│ stage execs : 210/500 (42.00%) │ new edges on : 9 (5.62%) │
│ total execs : 75.7M │ total crashes : 0 (0 unique) │
│ exec speed : 1543/sec │ total hangs : 123 (9 unique) │
├── fuzzing strategy yields ───────────────────────────────────────────┐ path geometry ───────────────────────────────────────────────────┐
│ bit flips : 2/117k, 0/117k, 0/117k │ levels : 9 │
│ byte flips : 0/14.7k, 0/14.7k, 0/14.6k │ pending : 0 │
│ arithmetics : 0/821k, 0/221k, 0/770 │ pend fav : 0 │
│ known ints : 0/74.5k, 0/408k, 0/642k │ own finds : 42 │
│ dictionary : 0/0, 0/0, 0/246k │ imported : n/a │
│ havoc : 37/26.0M, 2/46.9M │ variable : 0 │
│ trim : 42.30%/18.4k, 0.00% │ ───────────────────────────────────────────────────┐
```

Figure : bro-cut fuzz success