



Installation and Configuration

Robin Sommer

*International Computer Science Institute, &
Lawrence Berkeley National Laboratory*

`robin@icir.org`

`http://www.icir.org`

Bro Workshop 2009

Release Process

- At any time, we have two Bro distributions
 - A *stable release*, downloadable from <http://www.bro-ids.org/download.html>
 - A *development version* (aka “trunk”), available in a Subversion repository
svn checkout <http://svn.icir.org/bro/trunk/bro>
- We aim for a release cycle of 6-12 months
 - At that point, a snapshot of trunk becomes the new major release
 - If a release needs important bug fixes, we do minor releases in between
- Developers have their own *branches*
 - Most new features get developed there before they are merged into trunk
 - Some branches are public, others aren't
 - Branches are generally considered *unstable*

Resources

- Online resources:

- Web page <http://www.bro-ids.org>
- Wiki <http://www.bro-ids.org/wiki>
- Mailing list <http://www.bro-ids.org/mailling-list.html>
- ICIR Blog <http://blog.icir.org>
- Bug Tracker <http://tracker.icir.org/bro> (new!)

- Contributions are welcome!

- The CHANGES file

- Located in the top-level directory of the Bro distribution
- Bro's documentation does not reflect many of the new features
- CHANGES however records all changes and *is* kept up-to-date

Supported Platforms

- **Development focuses on three platforms:**
 - FreeBSD, Linux, and Mac OS
 - We use all of these frequently ourselves and things usually Just Work.
- **Users are running Bro on other platforms as well**
 - NetBSD, OpenBSD, Solaris
 - These often need small changes; we are happy to receive patches ...
- **FreeBSD seems to have best capture performance**
 - Most high-load Bros are running on FreeBSD machines (Mostly release 6, recently also release 7)
- **OS tuning can increase performance**
 - <http://www.net.t-labs.tu-berlin.de/research/hppc>

Basic Installation

- After downloading a release, do the usual
`./configure && make && make install`

Bro Executable	<code>/usr/local/bin/bro</code>
Standard Policy Scripts	<code>/usr/local/share/bro/ /usr/local/share/bro/sigs/ /usr/local/share/bro/time-machine/</code>
Site Policy Scripts	<code>/usr/local/share/bro/site/</code>
Broccoli	<code>/usr/local/{etc,include,lib}</code>

(Note that this layout has changed in 1.4)

Configure Options

- **Common `./configure` options**
 - `--prefix=/a/b/c` Install to different location
 - `--enable-bro6` Enables IPv6 support
 - `--enable-debug` Enables debugging (no optimization, debug information)
- `./configure` reports a status worth double-checking

Bro Configuration Summary

```
-----  
- Debugging enabled:            no  
- OpenSSL support:            yes  
- Non-blocking main loop:        no  
- Non-blocking resolver:        yes  
- Installation prefix:        /usr/local/bro  
- Perl interpreter:              /usr/bin/perl  
- Using basic_string:            yes  
- Using libmagic:             Yes  
- Using perftools:               no  
- Binpac used:                   shipped with Bro  
- Using libGeoIP:             Yes  
- Pcap used:                   system-provided
```

Running Bro

- **General usage:**

```
bro [-r <trace> | -i <interface> ] <scripts>
```

Examples:

```
bro -r web.pcap tcp scan weird alarm  
bro -i em0 mt
```

- **Command line options (excerpt, --help gives all)**

```
bro version 1.4  
usage: ../bin/bro [options] [file ...]  
-f|--filter <filter> | tcpdump filter  
-h|--help|-? | command line help  
-i|--iface <interface> | read from given interface  
-l|--print-scripts | print all loaded scripts  
-r|--readfile <readfile> | read from given tcpdump file  
-y|--flowfile <file>[=<ident>] | read from given flow file  
-Y|--netflow <ip>:<prt>[=<id>] | read flow from socket  
-s|--rulefile <rulefile> | read rules from given file  
-w|--writefile <writefile> | write to given tcpdump file  
-C|--no-checksums | ignore checksums  
-W|--watchdog | activate watchdog timer
```

Environment Variables

- BROPATH

Defines an alternative search path for policy scripts. Example:

```
export BROPATH=/path/one:/path/two
```

- BRO_LOG_SUFFIX

Defines a suffix for all log files (default *.log). Example:

```
export BRO_LOG_SUFFIX=`date +%Y-%m-%d_%H.%M.%S`
```

- BRO_DNS_FAKE

Set to run on laptop without working DNS

Support Scripts

Bro comes with a set of support tools/scripts in `aux/` * that can be useful but are not installed by default.

<code>aux/cf</code>	Converts time-stamps in log file to human-readable format.
<code>aux/hf</code>	Replaces numeric IP addresses in a log file with resolved host names.
<code>aux/rst</code>	Actively terminates live connections by injecting faked RST packets.
<code>nftools</code>	Tools for working with NetFlow data, including a converter for the flow data format Bro uses.

So, what is this “Bro-Lite”?

- Bro-Lite was a project 5 years ago to repackage Bro into a “turnkey” solution
 - Included a framework built around the Bro binary for configuration, installation, and maintenance of a Bro installation.
- As funding ran out, Bro-Lite is no longer supported
 - Still ships with Bro 1.4 (`make install-brolite`)
 - However, deprecated and scheduled for removal in Bro 1.5.
 - We do not recommend to use it anymore for new installations.
- Good news: A new framework is in the works
 - The “Cluster Shell” will provide an administration interface also for stand-alone Bro installations (and will likely be rebranded accordingly :-)

Thanks for your attention.

Robin Sommer

*International Computer Science Institute, &
Lawrence Berkeley National Laboratory*

`robin@icir.org`

`http://www.icir.org`