The Bro Network Security Monitor

Bro Integrations:

Some Misc. Bro Related Stuff

Jon Schipp, NCSA

BroCon15
MIT, Cambridge, Massachusetts

INTERNATIONAL COMPUTER SCIENCE INSTITUTE

NCSA

Agenda

- Outlining a few things I've work on
  - ISLET - Software that can be used for Bro training
  - Mal-dnssearch - Create Bro intel feeds from command-line
  - Sagan - Log analysis on Bro logs
  - Nagios - A plug-in to monitor your Bro cluster

ISLET

Isolated Scalable and Lightweight Environment for Training

- Background
  - The brototype released at BroCon'14 as BroLive!
  - Saw something greater and morphed into ISLET
- How?
  - Linux kernel has namespaces and control groups
    - Lightweight process virtualization
  - A container based solution for easy deployment
- Why?
  - Improve Bro training
    - Containers have millisecond startup times
    - Scalability - hundreds or thousands of users
    - VM's are slower, costlier, and larger

ISLET Cont.

- **User Perspective:** Looks and feels like a Virtual Machine
- **User Perspective:** Only needs a remote access tool like a ssh client
- **Admin Perspective:** Deployment of ISLET is dead simple

Deploying Bro with ISLET

```
$ git clone http://github.com/jonschipp/islet && cd islet
$ make install && make user-config && make security-config
$ make install-brolive-config
```

Use

```
$ ssh demo@islet.server.org
```

Official Image: https://registry.hub.docker.com/u/broplatform/brolive/

ISLET Cont.

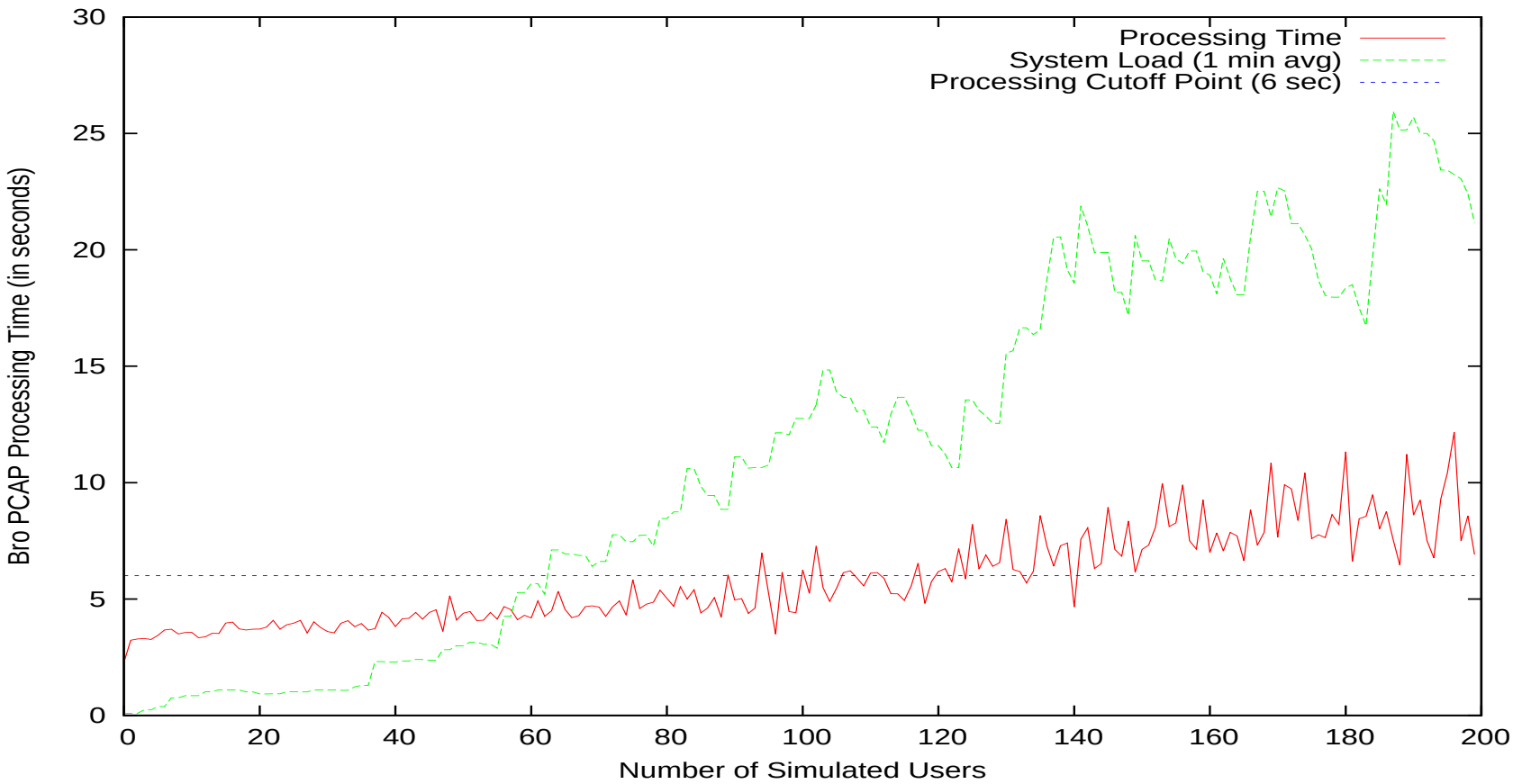
- Published a paper on ISLET using Bro
  - Substantiated container startup times with shell
  - Compared costs using virtual machines and containers
  - Benchmarked concurrent containers and simulated Bro users

Retrieve Paper

```
$ curl http://jonschipp.com/islet/islet-paper.pdf > islet-paper.pdf
```

ISLET/Bro Benchmark

- Simulated Bro training benchmark
  - Program execution/response time is good indicator for training software
  - EC2 c4.4xlarge(16CPU,30GB RAM) handles 100+ overly active users
- Anecdotally, 100+ users in the wild doing real training



Mal-dnssearch

Intel tool

- What?
  - Command-line intelligence pulling and matching script
  - Pulls existing feeds and supports many input logs e.g. PCAP, bind, Bro
  - Can generate data for Bro Intelligence Framework
- Why?
  - While writing the Bro and Intelligence Data post for the Bro blog I was looking for quick and easy way to test and create intel feeds.
- How?
  - mal-dnssearch pulls latest feed and notifies on match with input log
  - mal-dns2bro formats feeds for Intel Framework

Mal-dnssearch Cont.

Intel tool

- Intel Framework generation examples

Generate Snort Intel

```
$ mal-dnssearch -M snort -p | mal-dns2bro -T ip -s snort -n false -u http://labs.snort.org/feeds/ip-filter.blf > snort.intel
```

Generate Mandiant APT1 Intel

```
$ mal-dnssearch -M mandiant -p | mal-dns2bro -T dns -s mandiant > mandiant.intel
```

Generate custom feed

```
$ mal-dns2bro -f my.md5 -T filehashes -s myorg -n true -u file://my.md5 > custom.intel
```

Sagan

Log Analysis

- Background
  - Plenty of people integrate Bro logs with SIEMs
  - Many also do system log analysis, why not apply this to Bro's logs?
- How?
  - Use an existing log analysis tool
    - OSSEC, Sagan
  - Choice was Sagan because of existing Bro support and format language
    - Bro Intel preprocessor to read feeds
    - Popular and simple rule language
    - Unified2 output, for easy integration with other tools e.g. Snorby, SGuil, Squert.
- Why?
  - Wanted a quick way to write signatures without touching the cluster
  - Analysis across host and Bro logs
  - Maybe offload some work from a saturated Bro cluster

Sagan Detection

- Alert on Hola VPN attempts

Simple pattern match

```
alert tcp $EXTERNAL_NET any -> $HOME_NET $HTTP_PORT
(msg: "[BRO] Hola Client"; content: "client.hola.org"; content: " POST
"; parse_src_ip: 1; parse_dst_ip: 2; threshold: type limit, track by_src,
count 1, seconds 86400; classtype: suspicious-traffic; sid: 11000000; rev:1;)
```

Sagan Detection Cont.

- Alert on excessive non-existent domains from source IP

Count of NXDOMAIN matches

```
alert udp $EXTERNAL_NET any -> $HOME_NET $DNS_PORT (msg:
"[BRO] Excessive NXDOMAIN Responses (10k)"; content:
"NXDOMAIN"; after: track by_src, count 10000, seconds 3600;
parse_src_ip: 1; parse_dst_ip: 2; threshold: type limit, track by_src,
count 1, seconds 3600; classtype: suspicious-traffic; sid: 11000005; rev:1;)
```

- Use Bro Intel preprocessor to alert after 10+ bad domains from src IP

Count of intel DNS matches

```
alert udp $EXTERNAL_NET any -> $HOME_NET $DNS_PORT (msg:
"[BRO] Excessive Bad Domains (10+)"; bro-intel: domain; after: track
by_src, count 10, seconds 3600; parse_src_ip: 1; parse_dst_ip: 2;
threshold: type limit, track by_src, count 1, seconds 3600; classtype:
suspicious-traffic; sid: 13000000; rev:1;)
```

Sagan Detection Cont.

- Proxy detection via CONNECT method using flowbits - no alert

Possible proxy detection

```
alert tcp $EXTERNAL_NET any -> $HOME_NET $HTTP_PORT
(msg: "[BRO] Possible Proxy via CONNECT"; content: " CONNECT ";
content: "ROXY-CONNECTION"; parse_src_ip: 1; parse_dst_ip: 2;
flowbits: set, bro_possible_proxy_connect, 60; flowbits: noalert;
threshold: type limit, track by_src, count 1, seconds 86400; classtype:
suspicious-traffic; sid: 11000002; rev:1;)
```

- Alert if we see a transfer from files.log after

Proxy detection validation

```
alert tcp $EXTERNAL_NET any -> $HOME_NET $HTTP_PORT
(msg: "[BRO] Proxy Detected via CONNECT"; content: "SHA";
content:!"0.00"; pcre: "/SSL|HTTP|FTP/"; parse_src_ip: 2;
parse_dst_ip: 1; flowbits: isset,by_src,bro_possible_proxy_connect;
threshold: type limit, track by_src, count 1, seconds 86400; classtype:
suspicious-traffic; sid: 11000004; rev:1;)
```

Sagan Detection Cont.

- Proxy detection via GET or POST method using flowbits - no alert

Possible proxy detection

```
alert tcp $EXTERNAL_NET any -> $HOME_NET $HTTP_PORT
(msg: "[BRO] Possible Proxy via GET or POST"; pcre: "/ GET | POST
/"; content: "ROXY-CONNECTION"; pcre: "/http|https|ftp:/";
parse_src_ip: 1; parse_dst_ip: 2; flowbits: set, bro_possible_proxy_get,
60; flowbits: noalert; threshold: type limit, track by_src, count 1, seconds
86400; classtype: suspicious-traffic; sid: 11000001; rev:1;)
```

- Alert if we see a transfer from files.log after

Proxy detection validation

```
alert tcp $EXTERNAL_NET any -> $HOME_NET $HTTP_PORT
(msg: "[BRO] Proxy Detected via GET or POST"; content: "SHA";
content:!"0.00"; pcre: "/SSL|HTTP|FTP/"; parse_src_ip: 2;
parse_dst_ip: 1; flowbits: isset,by_src,bro_possible_proxy_get;
threshold: type limit, track by_src, count 1, seconds 86400; classtype:
suspicious-traffic; sid: 11000003; rev:1;)
```

Sagan

Plans

- Write more rules and get them in upstream sagan-rules
- Write Bro log normalization rules with liblognorm (testing them now)
- Continue to work with Champ "Da Beave" on improving Sagan for Bro

Nagios

Plugin

- What?
  - Nagios plug-in to monitor a Bro cluster
    - Worker status
    - Packet loss (netstats, Myricom)
    - Capture loss (capture_loss.log)
- Why?
  - Very the cluster is working and running as expected
- How?
  - Using the Nagios plugin API

Nagios Cont.

- Check worker status, critical on stopped or crashed workers

Status

```
check_bro.sh -f /bro/bin/broctl -T status
```

- Critical if average packet loss is 10% or greater for specified workers

Packet Loss

```
check_bro.sh -f /bro/bin/broctl -T loss -i "nids01,nids02" -c 10
```

- Critical if capture loss is 10% or greater

Capture Loss

```
check_bro.sh -f /bro/logs/current/capture_loss.log -T capture_loss -c 10
```

- Check packet counters for the following nodes

Myricom Packet Counters

```
check_bro.sh -f /opt/snf/bin/myri_counters -T myricom -i
"1.1.1.4,1.1.1.5"
```

Feedback/Questions

- If you play with this stuff let me know how it's going
- Patches welcome

Contact

Talk to me

Tweet me: @JonSchipp

E-mail me: jonschipp@gmail.com, jschipp@illinois.edu

References I

Official repository on Github.
In *https://github.com/jonschipp/islet*

Schipp, J., Dopheide, J., and Slagell, A., in the proceedings of XSEDE 2015, St. Louis, MO, Jul., 15.
ISLET: An Isolated, Scalable, & Lightweight Environment for Training.
In *http://jonschipp.com/islet/islet-paper.pdf*

Officical repository on Github.
In *https://github.com/jonschipp/mal-dnssearch*

Sagan: A multi-threaded log analysis engine.
In *http://sagan.quadrantsec.com/*

References II

Officical repository on Github.
In *https://github.com/jonschipp/nagios-plugins*