Blacklists Revisited

Aashish Sharma

asharma@lbl.gov
Blacklists Revisited

Lightening Talk
BroCon, 2017
Problem
Problem: Blocking Bad

Badness keeps increasing on the internet
How to manage blocking and more so unblocking
So, Can we identify……
Are blocked IPs coming back?
How long do we block before unblock?
Can we keep state forever

( that we can identify badness quickly )
Or, Are these one time visitors
Can we find out how many local IPs did the blacklisted IPs touched?
How long the scan lasted?
When was the last connection?
What's frequency of such connections?
Problem 2:

We can read a million IPs using input-framework, but how to send those to 50 workers?
Million IPs * 50 workers = 50 million Events
I want to be able to do this for 4 billion IPs
Bloomfilter

global Blacklist::m_w_add_bloom: event(val: opaque of bloomfilter);
1505245203.733616 1.2.3.4 8 128.3.x.y 0 icmp
Blacklist::Drop
[ip=1.2.3.4, source=blacklist.adhoc, comment=######## 2017-03-29: Multi-Causal Drop + COUNT=8, LOOKBACK=30 + Country_Analysis, COMMIT_COUNT=2488]

Result: [block_until=<uninitialized>, watch_until=0.0, num_reblocked=0, current_interval=0, current_block_id=, location=<uninitialized>]

Notice::ACTION_LOG 3600.000000 F
<table>
<thead>
<tr>
<th>Date</th>
<th>Time</th>
<th>IP Address</th>
<th>Source</th>
<th>Duration</th>
<th>Time Difference</th>
<th>Blacklist Domain</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aug 3</td>
<td>00:47:07</td>
<td>177.139.195.165</td>
<td>Blacklist::ONGOING</td>
<td>1</td>
<td>00:00:15:08</td>
<td>blacklist.adhoc</td>
</tr>
<tr>
<td>Aug 3</td>
<td>10:47:09</td>
<td>177.139.195.165</td>
<td>Blacklist::ONGOING</td>
<td>1</td>
<td>00:14:52:21</td>
<td>blacklist.adhoc</td>
</tr>
<tr>
<td>Aug 3</td>
<td>20:47:09</td>
<td>177.139.195.165</td>
<td>Blacklist::ONGOING</td>
<td>2</td>
<td>00:02:42:21</td>
<td>blacklist.adhoc</td>
</tr>
<tr>
<td>Aug 4</td>
<td>06:47:26</td>
<td>177.139.195.165</td>
<td>Blacklist::ONGOING</td>
<td>2</td>
<td>00:11:34:56</td>
<td>blacklist.adhoc</td>
</tr>
<tr>
<td>Aug 4</td>
<td>16:47:28</td>
<td>177.139.195.165</td>
<td>Blacklist::ONGOING</td>
<td>2</td>
<td>00:21:26:45</td>
<td>blacklist.adhoc</td>
</tr>
<tr>
<td>Aug 5</td>
<td>02:47:28</td>
<td>177.139.195.165</td>
<td>Blacklist::ONGOING</td>
<td>3</td>
<td>00:07:33:56</td>
<td>blacklist.adhoc</td>
</tr>
<tr>
<td>Aug 5</td>
<td>12:47:29</td>
<td>177.139.195.165</td>
<td>Blacklist::ONGOING</td>
<td>3</td>
<td>00:17:37:39</td>
<td>blacklist.adhoc</td>
</tr>
<tr>
<td>Aug 5</td>
<td>22:47:29</td>
<td>177.139.195.165</td>
<td>Blacklist::ONGOING</td>
<td>4</td>
<td>00:03:35:29</td>
<td>blacklist.adhoc</td>
</tr>
<tr>
<td>Aug 6</td>
<td>08:47:45</td>
<td>177.139.195.165</td>
<td>Blacklist::ONGOING</td>
<td>4</td>
<td>00:13:35:59</td>
<td>blacklist.adhoc</td>
</tr>
<tr>
<td>Aug 7</td>
<td>04:48:39</td>
<td>177.139.195.165</td>
<td>Blacklist::ONGOING</td>
<td>5</td>
<td>00:09:36:51</td>
<td>blacklist.adhoc</td>
</tr>
<tr>
<td>Aug 7</td>
<td>14:48:39</td>
<td>177.139.195.165</td>
<td>Blacklist::ONGOING</td>
<td>5</td>
<td>00:19:08:59</td>
<td>blacklist.adhoc</td>
</tr>
<tr>
<td>Aug 8</td>
<td>00:48:39</td>
<td>177.139.195.165</td>
<td>Blacklist::ONGOING</td>
<td>6</td>
<td>00:05:37:37</td>
<td>blacklist.adhoc</td>
</tr>
<tr>
<td>Aug 8</td>
<td>10:48:57</td>
<td>177.139.195.165</td>
<td>Blacklist::ONGOING</td>
<td>6</td>
<td>00:15:22:38</td>
<td>blacklist.adhoc</td>
</tr>
</tbody>
</table>
Bro in Apache Metron

Jon Zeolla

Jon.Zeolla@SeisoLLC.com

https://github.com/apache/metron
What is Metron?

Metron integrates a variety of open source big data technologies in order to offer a centralized tool for security monitoring and analysis. Metron provides capabilities for log aggregation, full packet capture indexing, storage, advanced behavioral analytics and data enrichment, while applying the most current threat intelligence information to security telemetry within a single platform.
What is Metron?

Metron integrates a ton of open source big data technologies in order to offer a centralized tool for security monitoring and analysis. Metron provides capabilities for log aggregation, full packet capture indexing, storage, advanced behavioral analytics and data enrichment, while applying the most current threat intelligence information to security telemetry within a single platform.
What is Metron?

Metron integrates a ton of Hadoop ecosystem technologies in order to offer a centralized tool for security monitoring and analysis. Metron provides capabilities for log aggregation, full packet capture indexing, storage, advanced behavioral analytics and data enrichment, while applying the most current threat intelligence information to security telemetry within a single platform.
What is Metron?

Metron integrates a ton of Hadoop ecosystem technologies in order to offer a way to use a large amount of security data (bro, snort, yaf, pcap, etc.).
What is Metron?

Metron integrates a ton of Hadoop ecosystem technologies in order to offer a way to use a large amount of security data (bro, snort, yaf, pcap, etc.).
Parsing and Normalizing
build_command = ( if [ ! -a /usr/local/lib/librdkafka.so ]; then curl -L https://github.com/edenhill/librdkafka/archive/v0.9.4.tar.gz | tar xvfz && cd librdkafka-0.9.4/ && ./configure --prefix=/usr/local --enable-sasl && make && sudo make install && cd - ; else if [ $(python -c "from ctypes import *; minver = 0x904ff; dll = cdll.LoadLibrary("/usr/local/lib/librdkafka.so"); version = dll.rd_kafka_version(); exit(0) if minver >= version else exit(1)") ]; then echo "At least version 0.9.4 of librdkafka is installed"; else echo "Please manually upgrade librdkafka to at least version 0.9.4"; exit 1; fi; fi && ./configure --bro-dist=%(bro_dist)s --with-librdkafka=/usr/local && make )
build_command = ( if [ ! -a /usr/local/lib/librdkafka.so ]; then curl -L https://github.com/edenhill/librdkafka/archive/v0.9.4.tar.gz | tar xvz && cd librdkafka-0.9.4/ && ./configure --prefix=/usr/local --enable-sasl && make && sudo make install && cd - ; else if [ $(python -c "from ctypes import *; minver = 0x904ff; dll = cdll.LoadLibrary("/usr/local/lib/librdkafka.so")"); version = dll.rd_kafka_version(); exit(0) if minver >= version else exit(1)" ]; then manually upgrade librdkafka to at least version 0.9.4; exit 1; fi; fi && ./configure --bro-dist=%(bro_dist)s --with-librdkafka=/usr/local && make)

Bro-pkg compile

Inability to solicit user feedback during bro-pkg install

#11 opened 12 hours ago by JonZeolla
InterpolationSyntaxError when referencing a variables multiple times

#10 opened 12 hours ago by JonZeolla
bro-pkg doesn't support external library dependencies

#9 opened 12 hours ago by JonZeolla
Bro-pkg is coming soon! Under step 6 of the documentation it shows that you can install a package with `bro-pkg install`, but I'm having some issues doing that. I've attached a screenshot.

Anybody know why this would be happening?
Enriching and Triaging
Indexing
Key Features

- Streaming data normalization and cleansing
- Ultra-high scale data processing with horizontal scaling
- Canned and custom, streaming enrichments that provide data-local context
- Native Threat Intelligence Integration
- “Modeling as a Service” platform
  - Heavily leveraging the profiler for feature extraction (IPs, Users, Subnets, Applications, etc.)
- PCAP storage/retrieval
Native bro log support in 0.4.1

- Conn
- DPD
- FTP
- Files
- CertsInfo
- SMTP
- SSL
- Weird
- Notice
- DHCP
- SSH
- Software
- Radius
- X509
- DevicesInfo
Taking Bro to the BSD Community

Michael Shirk

https://github.com/shirkdog/Presentations
Detecting Fakers & Attackers via Notice/Http Logs

Fatema Bannat Wala
fatema.bannatwala@gmail.com
Detecting Fake Google-Bots - I

Q: Internet Bots pretending to be Google-Bots and mining data from your sites?
A: Detect them and block them with BRO:)

- Characteristics of legit googlebot that Google uses for web-crawling:
  1. Uses CIDR: 66.249.0.0/16
  2. DNS’s ends in ‘googlebot.com’
  3. Uses UA having: ‘Googlebot’

```bash
$ cat notice.log | bro-cut -d | grep 'Scan::WebCrawler' | grep -i 'googlebot' | egrep -v "66\.249\." | awk -F'\t' '{print $1, $11, $12; system("host " $14)}'

  ts    note    msg

  2017-09-01T16:08:03-0400  Scan::WebCrawler  217.208.229.37 crawler is seen Mozilla/5.0 (compatible; Googlebot/2.1 +http://www.googlebot.com/bot.html) 37.229.208.217.in-addr.arpa domain name pointer 217-208-229-37-no205.tbcn.telia.com.

  2017-09-01T16:14:57-0400  Scan::WebCrawler  138.201.80.141 crawler is seen Googlebot-Image/1.0 141.80.201.138.in-addr.arpa domain name pointer static.141.80.201.138.clients.your-server.de.
```
Detecting Fake Google-Bots - II

Q: Have someone In-House pretending to be a Google-Bot?
A: Detect them and investigate them with BRO:)

Investigator questions:
• Is the host compromised?
• Is this user doing research?
• Is this a Proxy?

$ cat notice.log | bro-cut -d | grep 'Scan::WebCrawler' | grep -i 'googlebot' | egrep "128\.|128\.175\." | awk -F'\t' '{print $1, $11, $12; system("host " $14)}'

2017-09-01T16:08:03-0400  Scan::WebCrawler  128.xx.yy.zz crawler is seen Mozilla/5.0 (compatible; Googlebot/2.1; +http://www.google.com/bot.html)  zz.yy.xx.128.in-addr.arpa domain name pointer zz-yy-xx-128-aaa.bbb.ccc.

2017-09-01T16:14:57-0400  Scan::WebCrawler  128.ss.tt.vv crawler is seen Mozilla/5.0 (compatible; Googlebot/2.1; +http://www.google.com/bot.html)  uu.vv.tt.128.in-addr.arpa domain name pointer vv-tt-ss-128-ddd.eee.fff.
Detecting ShellShock Attempts

Q: Is someone still trying to give a shell shock to your servers?
A: Unveil them with BRO

$ cat http.log | bro-cut -d | awk -F'\t' '{ if ($13 ~ /cmd\..exe/ || $13 ~ /\bin\bash/) print $1, $2, $3, $4, $5, $6, $8, $13 }' | more

<table>
<thead>
<tr>
<th>ts</th>
<th>uid</th>
<th>id.orig_h</th>
<th>id.orig_p</th>
<th>id.resp_h</th>
<th>id.resp_p</th>
<th>method</th>
<th>user_agent</th>
</tr>
</thead>
<tbody>
<tr>
<td>2017-08-31T16:20:05-0400</td>
<td>Cjq5cD4agq22BN8cn9</td>
<td>31.210.47.92</td>
<td>58168</td>
<td>128.x.y.z</td>
<td>80   GET   () { foo;}; echo Content-Type: text/plain ; echo ; /bin/bash -c 'id ; uname -a ; whoami'</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2017-08-31T16:20:06-0400</td>
<td>CnGk7y4G6xBRYKlrd</td>
<td>31.210.47.92</td>
<td>58176</td>
<td>128.x.y.z</td>
<td>80   GET   () { foo;}; echo Content-Type: text/plain ; echo ; /bin/bash -c 'id ; uname -a ; whoami'</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2017-08-31T16:20:06-0400</td>
<td>CdshMA2SftnrUVBE</td>
<td>31.210.47.92</td>
<td>58175</td>
<td>128.x.y.z</td>
<td>80   GET   () { foo;}; echo Content-Type: text/plain ; echo ; /bin/bash -c 'id ; uname -a ; whoami'</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
CEASE: Leveraging Bro as a Network Feed

Nick Buraglio
CEASE: Leveraging Bro as a network intel feed

Nick Buraglio
Network Engineer,
ESnet Network Planning Team
Lawrence Berkeley National Laboratory

09/12/2017
Correlation Evaluation And Security Enforcement
Correlation Evaluation And Security Enforcement

- Deployed in high impact areas (public exchange points, etc.)
- Leverage existing data sets
  - Syslog
  - Netflow
  - Bro Alarms
  - Route topology
- Protect ESnet critical infrastructure
- Extend to an opt-in service for connectors
- Useful to any large network - not just ISPs
Bro alarms tuned properly...

- Allows us to...
- Correlate existing data sets to cross reference for:
  - Targeted attacks
  - Small[er] DDoS
  - Volumetric attacks
- Over a very large, carrier grade, international network
- Understand the topological path the given traffic may take
- Mitigate undesirable issues that may arise very far from any given sensor
What the heck is this “CEASE” thing?

Correlation Evaluation And Security Enforcement

.....at every transit POP

ESnet
We are hiring!!

● Do interesting things!
● Work on a **one of a kind, global scale** network!
● Learn from smart people!

Network Engineers!
[https://lbl.taleo.net/careersection/engineer/jobdetail.ftl?job=83959](https://lbl.taleo.net/careersection/engineer/jobdetail.ftl?job=83959)

Software Engineers!
[https://lbl.taleo.net/careersection/engineer/jobdetail.ftl?job=84046](https://lbl.taleo.net/careersection/engineer/jobdetail.ftl?job=84046)

Questions?

buraglio@es.net
Bro and PacketSled

Technical Overview

Leo Linsky

PacketSled
Challenges

● Our own pain points — Bro script is expensive.
● Customer use cases — documenting all interesting flows that other intrusion detection systems miss.
● Long term vision — we want our sensors to do more on the same hardware.
Options

- Compile Bro script and optimize the executables?
  - We want to run scripts dynamically, without restarting a sensor.
- Integrate a high performance alternative.
  - BIF’s, Binpac, and Bro plugins — need to be compiled and loaded with build, inaccessible for analysts looking to write and deploy detections.
  - LuaJIT is well supported, designed to be integrated via the Lua C API, and it gets faster as it runs.
Outcomes

- Project forked from Bro 2.5 (future versioning independent from mainline Bro.)
- Introduces alternative scripting framework built into the Bro-core to support Lua scripts.
- Changes in how we handle and generate metadata for unidentified flows.
- Performance improvements and customizations
Event generation from Bro kernel and plugins

- EventHandlers
  - Bro Scripts
  - Lua Manager
    - Lua script context
    - Run events
      - Script failure, context cleaned up
      - Faulty Lua script unloaded

nfps notice lua script uploaded to base/lua/ during runtime
Analyzers of Last Resort

- UDP
  - OpenVPN
  - TFTP
  - DHCPv6
  - ... (new flow)
  - conn (collect entropy, measure ASCII, log excerpts)

- TCP
  - SMB
  - SSH
  - ... (new flow)

- ICMP
  - ... (new flow)
  - No children
  - Collect entropy, measure ASCII, log excerpts
Other Additions

- Optimizing core loops (like `net_run()`) with preprocessor branch prediction macros `likely()` and `unlikely()` for ~3% speedup. We optimize for maximum load.
- UDP and TCP analyzers of last resort: modify analyzers to log the beginning of UDP and TCP flows which were not analyzed by any child analyzers. Includes entropy and ASCII counts, with thresholds that can be adjusted to identify plaintext protocols and pull an excerpt.
- General bug fixes (SMB, UID’s), improvements (mostly as BIF’s, such as bitwise operations), and customizations.
Next Steps...

Aaron Eppert

PacketSled
Thought Experiment

- How many of you have modified Bro?
- Are you productizing Bro?
- What does the sustainability model look like?
Challenges

- Political
- Commits
- Non-corruption of open source
- Risks as a vendor
We Want to Share

- PacketSled can share:
  - Lua
  - Analyzers of Last Resort
  - Optimizations and Bug fixes
Bro - Community?

- Vendor and Consumer Consortium
- What if we built a census roadmap balancing Vendor wants and Consumer needs with the realities of maintainers and committers?
The Bro Lognorm Plug-in

Jan Grashöfer

https://github.com/J-Gras
bro-lognorm

- “Wouldn’t it be cool to parse syslog messages inside of Bro?” – Seth
- Idea: Use *liblognorm* (rsyslog)
  - matches log lines against rules: *rule=greeting:Hello %who:word%*

Implementation:
- Bro plugin offering the *lognormalizer* opaque type
- Script-land interface for easy usage
bro-lognorm

**Setup:**

```plaintext
# test.rulebase:
# rule=greeting:Hello %who:word%

@load Bro/Lognorm
redef Lognorm::rule_files += {"test.rulebase"};

event greeting(who: string) {
    print fmt("Hi '%s'", who);
}

event Lognorm::unparsed_line(line: string) {
    print fmt("No rule for: '%s'", line);
}
```

**Usage:**

```plaintext
# Manually:
event bro_init() {
    Lognorm::normalize("Hello BroCon");
}

# Read files:
@load Bro/Lognorm/read_logs
redef Lognorm::log_file += {"test.log"}

# Read syslog:
@load Bro/Lognorm/read_syslog
```

- Use cases: ＿(ツ)_/－
- Example-plugin implementing an opaque type

- [github.com/J-Gras/bro-lognorm](github.com/J-Gras/bro-lognorm)
- jan.grashoefer@kit.edu