Hands-on Zeek (Bro) Labs

Elias Bou-Harb, Ph.D., CISSP


The Cyber Center For Security and Analytics

Cyber Threat Intelligence
Digital Forensics

Cross-disciplinary
R&D Themes

Infrastructure Security

Data Science

Software/ICT Security
Industry-Public Sector-Academia
Vision

R&D Projects
- Blockchain-Based Cyber Security
- Early Detection of Stealthy User Behavior
- Analytics-Driven Profiling of Grid Security
- IoT Threat Analytics

Studies
- IoT Malware
- Cyberterrorism Communities
- Offensive Language Classifiers
- Modeling public response to data breaches

Training
- NSF/DoD Cybertraining
- NSA/CAE Training
- U.S. Army Reserve
- CTF/Hackathons

Impact
- R, D & T For K12/Minorities
- IoT Exploitations’ Database
- Ransomware Prevention
- TTP/Patents
Intro to Zeek (Bro)
Network Intrusion Detection Systems

- Software/hardware systems that actively monitor live networks for malicious traffic, policy violations and unidentified anomalies
- Deployed to protect operational networks without disturbing normal/benign packet traffic flows
- In contrast to firewalls, NIDS are most often passive, looking for signatures or anomalies, although they can operate as NIPS as well
• Zeek’s Development began in 1995 by Vern Paxon (as Bro)
• Zeek’s scripting language creates a versatile environment for fine-grained anomaly-related detection and processing
• Versatile formatting of output data for preprocessing and advanced analytics

### Zeek’s Core

<table>
<thead>
<tr>
<th>Packet Stream</th>
<th>Event Engine</th>
<th>Policy Interpreter</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>Live network capture/offline packet parsing</td>
<td>Breaks down a packet stream based on packet features or related connection information</td>
<td>Comprised of event handlers; determines what Zeek will do when an event has been recorded</td>
<td>Generates output log files, alerts and notices declared within policy scripts</td>
</tr>
</tbody>
</table>
Network Traffic Signatures: A *Snort* Signature

Follows a rule-based format

```
(Action) (Protocol) (Source Address) (Source Port) (Direction) (Destination Address) (Destination Port)

alert tcp any 80 -> 192.168.x.x any (msg: “TCP Packet”; sid:100)
```

Rule Header

```
alert tcp any any -> [a.b.0.0/16,c.d.e.0/24] 80 (msg: “WEB-ATTACKS conf/httpd.conf attempt”; nocase; sid:1373; flow:to_server, established; content:”conf/httpd.conf”; [...] )
```

Rule Option
Network Traffic Signatures: A Zeek Signature

Follows a variable/data object-based format
Variables support strings, integers and floats

```
signature sid-1371 {
    ip,proto == tcp
    dst-ip == a.b.0.0/16,c.d.e.0/24
    dst-port == 80
    payload /.*conf/\httpd\.conf/
tcp-state established, originator
event “WEB-ATTACKS conf/http.conf attempt”
}
```
Zeek Log Files

- After processing network traffic, Zeek will output statistical log files.
- By default, log files will be separated by the transport protocol and related characteristics.
- At a basic level, these log files can be used to determine the presence of an anomaly.
- Zeek log files can be formatted and exported to external processing software.

<table>
<thead>
<tr>
<th>Connection</th>
<th>Protocol-Specific</th>
<th>Detection</th>
<th>Observations</th>
</tr>
</thead>
<tbody>
<tr>
<td>conn.log</td>
<td>http.log</td>
<td>notice.log</td>
<td>known_certs.log</td>
</tr>
<tr>
<td>files.log</td>
<td>ftp.log</td>
<td>signatures.log</td>
<td>known_services.log</td>
</tr>
<tr>
<td>x509.log</td>
<td>dns.log</td>
<td>traceroute.log</td>
<td>weird.log</td>
</tr>
</tbody>
</table>
Zeek Policy Scripts and Filters

• The Zeek scripting language is used to develop and implement filters and policies for the event-based engine

• Scripts can be implemented to permanently update Zeek’s event handling or used as a non-permanent filter

• Script events include (but are not limited to):
  – Protocol-specific events
  – Application-level headers
  – Unknown/broken connection handling

• Packet data is accessible within the filters to be used for calculations or to be exported into separate log files
Zeek Policy Scripts and Filters

Protocol-oriented Zeek Filter

Custom-based detectors

```
event udp_request(u:connection) {
    print fmt("A UDP Request was found!");
    print fmt("Source Address: %s Destination Port: %s", u$id$orig_h, u$id$resp_p);
}

event udp_reply(u: connection) {
    print fmt("A UDP Reply was found!");
    print fmt("Source Address: %s Destination Address: %s", u$id$orig_h, u$id$resp_h);
}
```

```
export {
    const addr_scan_interval = 5min &redef;
    const addr_scan_threshold = 20 &redef;
}

function horizontal_scanning(c: connection):bool {
    if (num_requests(c$id$orig_h) > addr_scan_threshold &&
        time_alive(c$connection) < addr_scan_interval) {
        print fmt("Horizontal Scanner Detected!");
        return c$id$orig_h;
    }

}//end function
```
Zeek Inferring IoT-generating Scanning

The Insecurity of the IoT Paradigm
Zeek Inferring IoT-generating Scanning

Malicious scans from compromised IoT devices
- 2 TB of Darknet Data (Daily)
- 840K global IoT exploitations (25K in the US)
- Exploitations in health services, manufacturing plants, gov. entitles
Malicious scans from compromised IoT devices

- 2 TB of Darknet Data (Daily)
- 840K global IoT exploitations (25K in the US)
- Exploitations in health services, manufacturing plants, gov. entities

<table>
<thead>
<tr>
<th>Date</th>
<th>2019-10-08</th>
<th>2019-11-25</th>
<th>2020-03-11</th>
<th>2020-04-26</th>
</tr>
</thead>
<tbody>
<tr>
<td># all infected hosts</td>
<td>754,169</td>
<td>836,255</td>
<td>806,326</td>
<td>839,082 (752,348)</td>
</tr>
<tr>
<td># Compromised IoT</td>
<td>274,699</td>
<td>229,488</td>
<td>224,964</td>
<td>480,049 (405,184)</td>
</tr>
<tr>
<td># all infected hosts (in USA)</td>
<td>16,614</td>
<td>15,957</td>
<td>23,779</td>
<td>25,468 (16,981)</td>
</tr>
<tr>
<td># Compromised IoT (in USA)</td>
<td>6,569</td>
<td>5,489</td>
<td>8,541</td>
<td>12,909 (4,920)</td>
</tr>
<tr>
<td># infected hosts in Medical</td>
<td>131</td>
<td>6</td>
<td>160</td>
<td>323 (311)</td>
</tr>
<tr>
<td># Compromised IoT in Medical</td>
<td>17</td>
<td>0</td>
<td>10</td>
<td>76 (58)</td>
</tr>
<tr>
<td># infected hosts in Medical (US)</td>
<td>26</td>
<td>2</td>
<td>22</td>
<td>58 (54)</td>
</tr>
<tr>
<td># Compromised IoT in Medical (US)</td>
<td>3</td>
<td>0</td>
<td>2</td>
<td>11 (10)</td>
</tr>
</tbody>
</table>
Hands-on Zeek (Bro) Labs
Zeek Hands-on Labs

- Lab 1 - Introduction to the Capabilities of Zeek
- Lab 2 - An Overview of Zeek Logs
- Lab 3 - Parsing, Reading and Organizing Zeek Files
- Lab 4 - Generating, Capturing and Analyzing Network Scanner Traffic
- Lab 5 - Generating, Capturing and Analyzing DoS and DDoS-centric Network Traffic
- Lab 6 - Introduction to Zeek Scripting
- Lab 7 - Advanced Zeek Scripting for Anomaly and Malicious Event Detection
- Lab 8 - Preprocessing of Zeek Output Logs for Machine Learning
- Lab 9 - Developing Machine Learning Classifiers for Anomaly Inference and Classification
- Lab 10 - Profiling and Performance Metrics of Zeek
Lab 4 - Generating, Capturing and Analyzing Network Scanner Traffic

<table>
<thead>
<tr>
<th>Virtual Machine</th>
<th>IP Address</th>
<th>Account</th>
<th>Password</th>
</tr>
</thead>
<tbody>
<tr>
<td>Zeek1</td>
<td>192.168.1.2</td>
<td>admin</td>
<td>password</td>
</tr>
<tr>
<td>DTN</td>
<td>192.168.1.3</td>
<td>root</td>
<td>password</td>
</tr>
<tr>
<td>Client</td>
<td>192.168.3.2</td>
<td>root</td>
<td>@dmin123</td>
</tr>
<tr>
<td>Zeek2</td>
<td>192.168.2.2</td>
<td>admin</td>
<td>password</td>
</tr>
<tr>
<td></td>
<td>192.168.3.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Router</td>
<td>192.168.1.1</td>
<td>root</td>
<td>password</td>
</tr>
<tr>
<td></td>
<td>192.168.2.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>203.0.113.2</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Lab 4 - Generating, Capturing and Analyzing Network Scanner Traffic

```
admin@bro1:~$ sudo nmap -SS 192.168.2.2
[sudo] password for admin:
Starting Nmap 7.60 (https://nmap.org) at 2019-06-26 18:27 EDT
Nmap scan report for 192.168.2.2
Host is up (0.0012s latency).
All 1000 scanned ports on 192.168.2.2 are closed
```

```
admin@bro2:~/Zeek-Labs-Workspace/TCP-Traffic
admin@bro2:~/Zeek-Labs-Workspace/TCP-Traffic$ sudo tcpdump -i ens33 -s 0 -w scan_traffic.pcap
[sudo] password for admin:
tcpdump: listening on ens33, link-type EN10MB (Ethernet), capture size 262144 bytes
^C8045 packets captured
8045 packets received by filter
0 packets dropped by kernel
```

```
admin@bro2:~/Zeek-Labs-Workspace/TCP-Traffic$ zeek-cut id.resp_p < conn.log | so
```

```
   82  83
   8  80
   8  443
   5  995
   5  6667
   5   3
   5   25
  4 9999
  4 9998
  4 999
```
Lab 9 - Developing Machine Learning Classifiers for Anomaly Inference and Classification
Hands-on Zeek (Bro) Labs

Enjoy the Labs 😊
Thanks for the support!

Elias Bou-Harb, Ph.D., CISSP
elias.bouharb@utsa.edu