Introduction to Practical Signature Development for Open Source IDS

Jason Williams & Jack Mott

#sf20v  •  Online  •  October 12-16
<table>
<thead>
<tr>
<th>Jack Mott (has mustache)</th>
<th>Jae Williams (no mustache)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Security Research Analyst</td>
<td>Security Research Analyst</td>
</tr>
<tr>
<td>Malware analysis</td>
<td>Malware analysis</td>
</tr>
<tr>
<td>Maldoc Research</td>
<td>Phishing Research</td>
</tr>
<tr>
<td>Signatures for ETPRO/OPEN, ClamAV, yara, etc</td>
<td>Signatures for ETPRO/OPEN, ClamAV (Phishing)</td>
</tr>
<tr>
<td>Sigdev Trainer</td>
<td>Sigdev Trainer</td>
</tr>
</tbody>
</table>
Why IDS/IPS?

• Still an extremely valuable tool in your arsenal

• Applicable when discussing defense in depth

• Not perfect

• Provide context

• What can full PCAP provide?
IDS Rule Theory

• Generally, we want agile but effective rules
  • Don’t be like generic AV names and hash-based detections

• Specific enough to capture desired traffic without False Negatives

• Loose enough to capture variants without False Positives

• Balance!

• Won’t always work this way 😞
WIRESHARK!

• We detonate a ton of malware and make a LOT of pcaps

• We need a way to view individual sessions for generating rules

• We write ET rules to be a highlighter for your network traffic and we use wireshark ALL DAY EVERYDAY
What is an IDS rule?

POST /generate.php HTTP/1.1
User-Agent: DetoxCrypto2
Content-Type: application/x-www-form-urlencoded
Host: detoxcrypto.net16.net
Content-Length: 26
Expect: 100-continue
Connection: Keep-Alive

definition=HTTP/1.1 200 OK
Date: Sat, 13 Aug 2016 04:45:30 GMT
Server: Apache
X-Powered-By: PHP/5.2.17
Content-Length: 186
Connection: close
Content-Type: text/html

MQoxjmeGRPmHgh2GVdFSPHnycHwL5i7Z4
<!--- Hosting24 Analytics Code -->
<!--- End Of Analytics Code -->
Rule Foundations
IDS Rule Basic Format

action protocol from_ip port -> to_ip port

(msg:"something"; content:"something";
content:"something else"; sid:10000000; rev:1;)
Rule Action

• Tells the IDS engine what to do when traffic matches this rule
  • alert
    • Generate alert, and log matching packets, but let the traffic through
  • log
    • Log traffic—no alert
  • pass
    • Ignore the packet, allow it through
  • drop
    • If IPS mode, sensor should drop the offending packet
  • reject
    • IDS will send TCP reset packet
Rule Protocol

- Suricata and Snort have the ability to detect specific protocols declared by the rule writer
  - tcp
  - udp
  - icmp
  - ip
  - http (Suricata only)
  - tls (Suricata only)
  - dns (Suricata only)
  - smb (Suricata only)
Rule Hosts Variables

- This is how you declare who is sending traffic to who
- Configurable via suricata.yaml and snort.conf
  - Contains defaults, but double check them
- \$HOME_NET
  - Refers to internal networks, specified in the conf/yaml
- \$EXTERNAL_NET
  - Not \$HOME_NET, or what you choose in conf/yaml
- \$HTTP_SERVERS, \$SMTP_SERVERS, etc...
- Single IP
Rule Direction

• Simply stated by an arrow: ->
• This tells the engine what direction traffic is flowing between hosts
• Traffic from internal host -> outbound
  • \$HOME_NET any -> \$EXTERNAL_NET any
• Traffic from external host -> inbound
  • \$EXTERNAL_NET any -> \$HOME_NET any
• Can be bidirectional (not recommended) by using: <>
  • \$EXTERNAL_NET any <> \$HOME_NET any
Rule Ports

• Used in tandem with the src/dst host variables

• Declares the port in which traffic for this rule will be evaluated
  • `alert tcp $HOME_NET any -> $EXTERNAL_NET 9003`

• Like the Hosts variables, ports may have variables as well
  • `$HTTP_PORTS`, `$SMTP_PORTS`, `$FTP_PORTS`, etc...
  • Configurable in conf/yaml

• Ports may be negated by placing a ! in front of it
  • `$EXTERNAL_NET !80`
Rule Ports (cont...)

- Ports may be expressed in various ways
  - Single port
    - 80
  - Multiple ports
    - [80,8080,443,9000]
  - Port ranges
    - [8000:9000]
  - Combination
    - $HOME_NET [1024:] -> $EXTERNAL_NET [80,800,6667:6669,!200]
    - What does this say?
## Exercise – Rule Foundations

<table>
<thead>
<tr>
<th>Source</th>
<th>SrcPort</th>
<th>Host</th>
<th>Destination</th>
<th>DstPort</th>
<th>Protocol</th>
<th>Stat</th>
<th>Length</th>
<th>Info</th>
</tr>
</thead>
<tbody>
<tr>
<td>192.168.4.151</td>
<td>49689</td>
<td></td>
<td>137.74.223.62</td>
<td>80</td>
<td>TCP</td>
<td>0</td>
<td>49689-80 [SYN] Seq=0 Win=8192 Len=0 MSS=1464 WS=4 SACK [ACK] Seq=1 Ack=1 Win=65500 Len=0</td>
<td></td>
</tr>
<tr>
<td>137.74.223.62</td>
<td>80</td>
<td></td>
<td>192.168.4.151</td>
<td>49689</td>
<td>TCP</td>
<td>0</td>
<td>80-49689 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1464 WS=4 SACK [ACK] Seq=1 Ack=1 Win=65500 Len=0</td>
<td></td>
</tr>
<tr>
<td>192.168.4.151</td>
<td>49689</td>
<td></td>
<td>137.74.223.62</td>
<td>80</td>
<td>TCP</td>
<td>0</td>
<td>49689-80 [ACK] Seq=1 Ack=1 Win=65500 Len=0</td>
<td></td>
</tr>
<tr>
<td>192.168.4.151</td>
<td>49689</td>
<td>free.diendancacanh.net</td>
<td>137.74.223.62</td>
<td>80</td>
<td>HTTP</td>
<td>336</td>
<td>GET /radio/sometime-estate-sleepy-10006700 HTTP/1.1</td>
<td></td>
</tr>
<tr>
<td>137.74.223.62</td>
<td>80</td>
<td></td>
<td>192.168.4.151</td>
<td>49689</td>
<td>TCP</td>
<td>0</td>
<td>80-49689 [ACK] Seq=1 Ack=337 Win=30336 Len=0</td>
<td></td>
</tr>
<tr>
<td>137.74.223.62</td>
<td>80</td>
<td></td>
<td>192.168.4.151</td>
<td>49689</td>
<td>TCP</td>
<td>1318</td>
<td>[TCP segment of a reassembled PDU]</td>
<td></td>
</tr>
<tr>
<td>137.74.223.62</td>
<td>80</td>
<td></td>
<td>192.168.4.151</td>
<td>49689</td>
<td>TCP</td>
<td>1209</td>
<td>[TCP segment of a reassembled PDU]</td>
<td></td>
</tr>
<tr>
<td>192.168.4.151</td>
<td>49689</td>
<td></td>
<td>137.74.223.62</td>
<td>80</td>
<td>TCP</td>
<td>0</td>
<td>49689-80 [ACK] Seq=337 Ack=1319 Win=65900 Len=0</td>
<td></td>
</tr>
<tr>
<td>137.74.223.62</td>
<td>80</td>
<td></td>
<td>192.168.4.151</td>
<td>49689</td>
<td>HTTP</td>
<td>200</td>
<td>5 HTTP/1.1 200 OK (text/html)</td>
<td></td>
</tr>
</tbody>
</table>

alert ___ $_________ ______ -> $_________ _______
Exercise – Rule Foundations

### Suricata

```
alert http $HOME_NET any -> $EXTERNAL_NET any
```

### Snort

```
alert tcp $HOME_NET any -> $EXTERNAL_NET $HTTP_PORTS
```
Rule Message

- msg: "DetoxCrypto Ransomware CnC Activity"

- Arbitrary text that appears when the rule fires and is logged/alert

- Consistency is key!

- Consider adding:
  - Malware architecture: Win32/64, MSIL, ELF, OSX, etc
  - Malware family/name: njRAT, Locky, CryptXXX, Zeus
  - Malware action: Checkin, Activity, Key Exchange, Heartbeat, Exfil
Rule Content

• The most basic building block for pattern matching
• Matching unique content in packets for detection
• Careful on what you choose

• Must use hex for certain special characters
  • ; “ : 
  • content: ”some thing”;
  • content: ”some|20|thing”;
  • content: ”User-Agent|3a 20|”;
  • content: ”s|00|o|00|m|00|e|00|t|00|h|00|i|00|n|00|g”;
Rule Meta

- SID
  - Signature ID
  - sid:10000000
- Reference
  - reference:md5,e273508a2f2ed45c20a2412f7d62eceb;
  - reference:url,malwarefor.me/2015-12-27-sundown-ek;
  - reference:cve,2016-3254;
- Metadata
  - Free form key/pair valued text
  - metadata: created_at 2020_10_01, author_name jae;
- Revision
  - Tells us what version of the rule we are on
  - rev:9;
Writing Signatures for DNS
Sig all the domains!

• A very common thing we do

micrs0ft-update.com

• You have an IOC and you need to apply it to your network as fast as possible

• What value would you place on DNS signatures?
Frame 4: 85 bytes on wire (680 bits), 85 bytes captured (680 bits)
Internet Protocol Version 4, Src: 172.16.57.213 (172.16.57.213), Dst: 172.16.57.2 (172.16.57.2)
User Datagram Protocol, Src Port: 58575 (58575), Dst Port: domain (53)
Domain Name System (query)

[Response In: 9]
Transaction ID: 0x971c

Flags: 0x0100 Standard query
  0... = Response: Message is a query
  .000 0... = Opcode: Standard query (0)
  .0... = Truncated: Message is not truncated
  ....1... = Recursion desired: Do query recursively
  ....0... = Z: reserved (0)
  .......0 = Non-authenticated data: Unacceptable

Questions: 1
Answer RRs: 0
Authority RRs: 0
Additional RRs: 0

Queries

it-administrator.sitey.me: type AAAA, class IN
  Name: it-administrator.sitey.me
  Type: AAAA (IPv6 address)
  Class: IN (0x0001)
dns.query; to the rescue!

• DNS request query field

dns.query; content:”microsoft-update.com”;

• Sticky buffer

• Suricata has lots of these, we’ll see some more later
So much nicer looking, so much more efficient (so much less typing)

<table>
<thead>
<tr>
<th>Suricata 5/6</th>
</tr>
</thead>
<tbody>
<tr>
<td>alert dns $HOME_NET any -&gt; any any (msg:&quot;ET MALWARE PowerGhost Staging CnC in DNS Query&quot;; dns.query; content:&quot;box.conf1g.com&quot;; endswith; classtype:domain-c2; sid:2030998; rev:2;)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Suricata &gt; 3.2 and 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>alert dns $HOME_NET any -&gt; any any (msg:&quot;ET TROJAN PowerGhost Staging CnC in DNS Query&quot;; dns_query; content:&quot;box.conf1g.com&quot;; isdataat:!1,relative; classtype:trojan-activity; sid:2030998; rev:2;)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Snort 2.9 and Suricata &lt; 3.2</th>
</tr>
</thead>
<tbody>
<tr>
<td>alert udp $HOME_NET any -&gt; any 53 (msg:&quot;ET TROJAN PowerGhost Staging CnC in DNS Query&quot;; content:&quot;</td>
</tr>
</tbody>
</table>
Things you do over and over should be scripted (not abused)

https://github.com/darienhuss/dns_sigs
Writing rules for SSL / TLS
SSL / TLS

• Obviously a bit of a blind spot for most IDS/IPS
• Cannot see into the traffic, just that it’s happening
  • Unless MITM, which is cool too!
• Let’s Encrypt!
• Wait!
• We can do something with SSL/TLS

SSL Handshake Process (simplified)

1) Client requests HTTPS session
2) Certificate sent back (with Public Key)
3) Client creates random session key
4) Session key encrypted with public key
   At this point, only client knows session key
5) Encrypted session key sent to server
6) Session key decrypted with private key
   At this point, both client and server know session key
7) Session encrypted with symmetric session key
SSL / TLS Basics

TLSv1 Record Layer: Handshake Protocol: Certificate
- Content Type: Handshake (22)
- Version: TLS 1.0 (0x0301)
- Length: 641

- Handshake Protocol: Certificate
  - Handshake Type: Certificate (11)
  - Length: 637
  - Certificates Length: 634
- Certificates (634 bytes)
  - Certificate Length: 631

- Certificate: 30820273308201da0302010202045da0dec300d06092a... (id-at-commonName=vuinuzhz.com)

  - signedCertificate
    - version: v3 (2)
    - serialNumber: 1570823884
      - signature (sha256WithRSAEncryption)
      - issuer: rdnSequence (0)
      - validity
        - subject: rdnSequence (0)
          - rdnSequence: 1 item (id-at-commonName=vuinuzhz.com)
          - RDNSequence item: 1 item (id-at-commonName=vuinuzhz.com)
          - RelativeDistinguishedName item (id-at-commonName=vuinuzhz.com)
SSL / TLS - What to sig on

• Serial Number
  • content: "|09 00 97 ae 20 7e 61 5f 58 15|"

• Common Name (CN)
  • content: "|55 04 03|"

• Organization (O)
  • content: "|55 04 0a|

• Organizational Unit (OU)
  • content: "|55 04 0b|

• Country (C)
  • content: "|55 04 06|

• State (S)
  • content: "|55 04 08|

• Locality (L)
  • content: "|55 04 07|"
SSL/TLS Keywords (Suricata only)

• Suricata has protocol detection for TLS/SSL
  • Using the ‘tls’ protocol, we can observe TLS/SSL on any port
• Suricata also has TLS/SSL specific keywords for various buffers
tls.cert_subject

• Match on the content found in the certificates Subject field (OU,CN,ST,etc)
• Direction for this is usually $EXTERNAL_NET -> $HOME_NET
• Sticky Buffer
• Usage:

  tls.cert_subject; content:"CN=z55gc.com"; nocase;
tls.cert_serial

• Match on the certificates serial number
• Direction for this is usually $EXTERNAL_NET → $HOME_NET
• Sticky Buffer
• Important formatting note: must use uppercase, colon separated hex (or lowercase colon separated hex and nocase;)
• Usage:
  
  `tls.cert_serial;
  content:"0A:01:41:42:00:00:01:53:85:73:6A:0B:85:EC:A7:08";`
Writing Signatures for SSL / TLS

alert tls $EXTERNAL_NET any -> $HOME_NET any (msg:"ET MALWARE Observed Malicious SSL Cert (BazaLoader CnC)"; flow:from_server,established; tls.cert_subject; content:"CN=z55gc.com"; nocase; fast_pattern; endswith; tls.cert_issuer; content:"C=US, O=Let's Encrypt, CN=Let's Encrypt Authority X3"; classtype:domain-c2; sid:2030988; rev:1;)

• Protocol: tls (Suricata)
• Ports: any/any (Suricata)
• Content 1: Serial number
• Content 2: Certificate Issuer
Writing Signatures for SSL / TLS

alert tcp $EXTERNAL_NET 443 -> $HOME_NET any (msg:"ET TROJAN Observed Malicious SSL Cert (BazaLoader CnC)"; flow:from_server,established; content:"|16|"; content:"|0b|"; within:8; content:"|55 04 03|"; distance:0; content:"|09|z55gc.com"; distance:1; within:10; fast_pattern; content:"|55 04 06|"; distance:0; content:"|02|US"; distance:1; within:3; content:"|55 04 0a|"; distance:0; content:"|0d|Let's Encrypt"; distance:1; within:14; content:"|55 04 03|"; distance:0; content:"|1a|Let's Encrypt Authority X3"; distance:1; within:27; classtype:trojan-activity; sid:2030988; rev:2;)

• Protocol: tcp (Snort)
• Ports: 443/any (Snort)
• Content: Serial number
• Content: Certificate Issuer
tks jack - https://github.com/malwareforme/ssl_sigs

puncher@sigdev:/tools/ssl_sigs$ python ssl_sigs.py
usage: ssl_sigs.py [-h] -d DOMAIN -m MESSAGE [-r REFERENCE] [-c CLASSTYPE]
                  [-s SID]
ssl_sigs.py: error: argument -d/--domain is required
puncher@sigdev:/tools/ssl_sigs$
puncher@sigdev:/tools/ssl_sigs$
puncher@sigdev:/tools/ssl_sigs$ python ssl_sigs.py -d google.com -m 'bad google ssl'
#Suricata 3.2+ rule:
alert tls $EXTERNAL_NET any -> $HOME_NET any (msg:"bad google ssl"; flow:established,to_client;
  tls_cert_subject; content:"CN=google.com"; nocase; isdataat::1,relative; classtype:trojan-activity;
  sid:10000000; rev:1;)

#Suricata 1.3+ rule:
alert tls $EXTERNAL_NET any -> $HOME_NET any (msg:"bad google ssl"; flow:established,to_client;
  content:"|55 04 03|"; content:"|0a|go0gle.com"; distance:1; within:11; fast_pattern; classtype:
  trojan-activity; sid:10000000; rev:1;)

#Snort 2.9+ rule:
alert tcp $EXTERNAL_NET 443 -> $HOME_NET any (msg:"bad google ssl"; flow:established,to_client;
  content:"|55 04 03|"; content:"|0a|go0gle.com"; distance:1; within:11; fast_pattern; classtype:
  trojan-activity; sid:10000000; rev:1;)

Writing Signatures for HTTP
HTTP Buffers

• Suricata and Snort have the ability to parse HTTP and place packet contents into buffers to improve matching
• Much faster than searching raw packet
• We can use these to our advantage in conjunction with the other keywords and modifiers!
  • http.user_agent; content:”DetoxCrypto”;
  • content:”User-Agent|3a 20|DetoxCrypto”; http_header;
The `http.method` keyword can be used for a content involving the method in which the HTTP Request was made:

- `http.method; content:”GET”;
- `http.method; content:”POST”;
- `http.method; content:”HEAD”;

POST /generate.php HTTP/1.1
User-Agent: DetoxCrypto2
Content-Type: application/x-www-form-urlencoded
Host: detoxcrypto.net16.net
Content-Length: 26
Expect: 100-continue
Connection: Keep-Alive
http.uri

• Used for capturing any content present in the URI string of a request
• `http.uri; content:"/generate.php";`
• `urilen` keyword
  • `urilen:<number>;`

```plaintext
POST /generate.php HTTP/1.1
User-Agent: DetoxCrypto2
Content-Type: application/x-www-form-urlencoded
Host: detoxcrypto.net16.net
Content-Length: 26
Expect: 100-continue
Connection: Keep-Alive

publickey=sdJoFsAv3jSNMEYxHTTP/1.1 200 OK
```
http.header

• This is used for any field present in the Header section

```plaintext
http.header; content:”User-Agent|3a|”;
http.header; content:!”Referer|3a|”;
```

• Cookie is not able to be used with this buffer
  • It has its own buffer —> http.cookie

```
POST /generate.php HTTP/1.1
User-Agent: DetoxCrypto2
Content-Type: application/x-www-form-urlencoded
Host: detoxcrypto.net16.net
Content-Length: 26
Expect: 100-continue
Connection: Keep-Alive
```

```plaintext
publickey=sdJoFsAv3jSNMEYxHTTP/1.1 200 OK
Date: Sat, 13 Aug 2016 04:45:30 GMT
```
http.request_body

- Used for an HTTP requests payload
- Commonly observed with POST requests

```plaintext
http.request_body; content: "publickey="
```

POST /generate.php HTTP/1.1
User-Agent: DetoxCrypto2
Content-Type: application/x-www-form-urlencoded
Host: detoxcrypto.net16.net
Content-Length: 26
Expect: 100-continue
Connection: Keep-Alive

```
publickey=sdJoFsAv3jSNMEYx HTTP/1.1 200 OK
Date: Sat, 13 Aug 2016 04:45:30 GMT
Server: Apache
X-Powered-By: PHP/5.2.17
```

```plaintext
```
http.user_agent

• Suricata only! Fast! Use it!
• Will parse the field between User-Agent|3a 20| and |0d 0a|
• Suricata

    http.user_agent; content:="DetoxCrypto";

• Snort

    content:="User-Agent|3a 20|DetoxCrypto"; http_header;

POST /generate.php HTTP/1.1
User-Agent: DetoxCrypto2
Content-Type: application/x-www-form-urlencoded
Host: detoxcrypto.net16.net
Content-Length: 26
Expect: 100-continue
What is an IDS rule?

```
POST /generate.php HTTP/1.1
User-Agent: DetoxCrypto2
Content-Type: application/x-www-form-urlencoded
Host: detoxcrypto.net16.net
Content-Length: 26
Expect: 100-continue
Connection: Keep-Alive

publickey=sdJoFsAv3jSNMYxHTTP/1.1 200 OK
Date: Sat, 13 Aug 2016 04:45:30 GMT
Server: Apache
X-Powered-By: PHP/5.2.17
Content-Length: 186
Connection: close
Content-Type: text/html

MQoxjmeGRPHGh2GVdFSPHnycHwL5i7Z4
<!-- Hosting24 Analytics Code -->
<!-- End Of Analytics Code -->
```
What does a Production IDS rule look like?

**Suricata 5/6**

```plaintext
alert http $HOME_NET any -> $EXTERNAL_NET any (msg:"ETPRO MALWARE DetoxCrypto Ransomware CnC Activity"; flow:established,to_server; http.method; content:"POST"; http.uri; content:"/generate.php"; endswith; http.user_agent; content:"DetoxCrypto"; fast_pattern; http.request_body; content:"publickey="; startswith; http.header_names; content:""Referer""; classtype:command-and-control; sid:100; rev:1;)
```

**Snort 2.9.x**

```plaintext
alert tcp $HOME_NET any -> $EXTERNAL_NET $HTTP_PORTS (msg:"ETPRO TROJAN DetoxCrypto Ransomware CnC Activity"; flow:established,to_server; content:"POST"; http_method; content:"/generate.php"; http_uri; content:"User-Agent|3a 20|DetoxCrypto"; fast_pattern:3,20; http_header; content:"publickey="; depth:10; http_client_body; content:""Referer|3a|""; http_header; pcre:"/\..php$/U"; sid:101; rev:1;)
```
Wrapping Up

• Network analysis!

• The more you look into your network, the more likely you will be to know what “normal” and “abnormal” look like.

• Use multiple rule options together for maximum detection/efficiency

• Continue working…
  • ET OPEN Ruleset - Free to download and play with (learn from)
  • Snort Community Ruleset - Free to download and play with (learn from)
  • Security Onion – Free Ubuntu distro with Network Analysis tools
  • malware-traffic-analysis.net– PCAPs and malware samples galore
  • broadanalysis.net- PCAPs and malware samples galore
Workshop Materials

• If you want to dig into this stuff on your own, we’re making our workshop materials available online for free

• A few exercises with Suri5, a helpful workbook, and the tools we mentioned

• Download the Virtual Machine at vm.sigdev.me

• Download the Workbook at workbook.sigdev.me

• Good for 48 hours!
Questions?

Jack Mott
@malwareformeme
Security Research Analyst
jmott@emergingthreats.net

Jae Williams
@switchingtoguns
Security Research Analyst
jae@emergingthreats.net

Virtual Machine: vm.sigdev.me

Workbook: workbook.sigdev.me

content:”stay safe”; content:”wear a mask”;