Suricata Intrusion Detection System

Tutorial Agenda

- Topics
  - Setting up and installation (on Ubuntu VM)
    - Configuration
    - Rule-set management
  - Suricata in action – some use cases
  - Signature/Rule Writing
- Hands-on:
  - Lab Sheet
  - Need to ssh to our backend for installation
  - Wireshark is recommended for Lab 5,*6*

Objectives

- High level overview of Suricata
  - Functionality
  - Features
  - Setup
- Threats / Detection
  - Contextual
  - Rules / Signatures
- Deploy, try or play!

Suricata Intrusion Detection System

- It is open source and owned by a community-run non-profit foundation, the Open Information Security Foundation (OISF).
- The Suricata source code is licensed under version 2 of the GNU General Public License

Suricata - History

- Beta release – Dec 2009
- First standard release – July 2010
- Features
  - Multi-threading
  - Automatic protocol detection
  - JSON standard outputs
  - file matching, logging, extraction, md5 checksum
  - DNS logger
  - Many more!

https://suricata.io/features/all-features/

In a nutshell

- The Suricata engine is capable of real time intrusion detection (IDS), inline intrusion prevention (IPS), network security monitoring (NSM) and offline pcap processing
- Suricata inspects the network traffic using a powerful and extensive rules and signature language, and has powerful Lua scripting support for detection of complex threats
- With standard input and output formats like YAML and JSON integrations with tools like existing SIEMs, Splunk, Logstash/Elasticsearch, Kibana, and other database become effortless

Suricata Installation and Configuration

Installation

- We will install a couple of things
  - suricata
  - jq
  - evebox
- Refer to installation.pdf

```
apnic@suricata:~$ mkdir Downloads
cdapnic@suricata:~$ cd Downloads
apnic@suricata:~/Downloads$ wget https://evebox.org/files/release/0.14.0/evebox_
0.14.0_amd64.deb
sudo dpkg--2021-07-26 03:29:40--  https://evebox.org/files/release/0.14.0/evebox
_0.14.0_amd64.deb
 -i eveResolving evebox.org (evebox.org)... box_0.14.0_amd64.deb172.105.5.173
Connecting to evebox.org (evebox.org)|172.105.5.173|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 7899390 (7.5M) [application/vnd.debian.binary-package]
Saving to: ‘evebox_0.14.0_amd64.deb’

evebox_0.14.0_amd64 100%[===================>]   7.53M  16.0MB/s    in 0.5s

2021-07-26 03:29:41 (16.0 MB/s) - ‘evebox_0.14.0_amd64.deb’ saved [7899390/78993
90]

apnic@suricata:~/Downloads$ sudo dpkg -i evebox_0.14.0_amd64.deb
```

```
apnic@suricata:~$ sudo add-apt-repository ppa:oisf/suricata-stable
 Suricata IDS/IPS/NSM stable packages
http://www.openinfosecfoundation.org/
http://planet.suricata-ids.org/
http://suricata-ids.org/

Suricata IDS/IPS/NSM - Suricata is a high performance Intrusion Detection and Pr
evention System and Network Security Monitoring engine.

Open Source and owned by a community run non-profit foundation, the Open Informa
tion Security Foundation (OISF). Suricata is developed by the OISF, its supporti
ng vendors and the community.

This Engine supports:

- Multi-Threading - provides for extremely fast and flexible operation on multic
ore systems.
- Multi Tenancy - Per vlan/Per interface
- Uses Rust for most protocol detection/parsing
- TLS/SSL certificate matching/logging
- JA3 TLS client fingerprinting
```

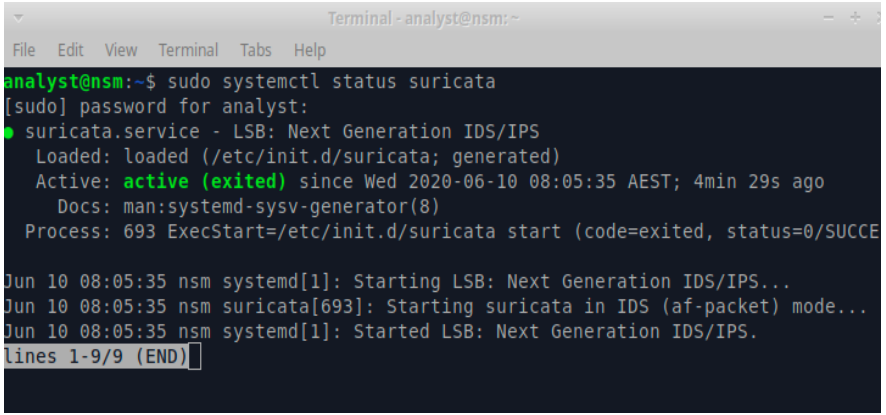
```
- VXLAN support
- All JSON output/logging capability
- IDS runmode
- IPS runmode
- IDPS runmode
- NSM runmode
- eBPF/XDP
- Automatic Protocol Detection and logging - IPv4/6, TCP, UDP, ICMP, HTTP, SMTP,
 TLS, SSH, FTP, SMB, DNS, NFS, TFTP, KRB5, DHCP, IKEv2, SNMP, SIP, RDP
- SCADA automatic protocol detection - ENIP/DNP3/MODBUS
- File Extraction HTTP/SMTP/FTP/NFS/SMB - over 4000 file types recognized and ex
tracted from live traffic.
- File MD5/SHA1/SHA256 matching
- Gzip Decompression
- Fast IP Matching
- Datasets matching
- Rustlang enabled protocol detection
- Lua scripting

and many more great features -
http://suricata-ids.org/features/all-features/
 More info: https://launchpad.net/~oisf/+archive/ubuntu/suricata-stable
Press [ENTER] to continue or Ctrl-c to cancel adding it.
```

System Check

- Suricata is already running as a service

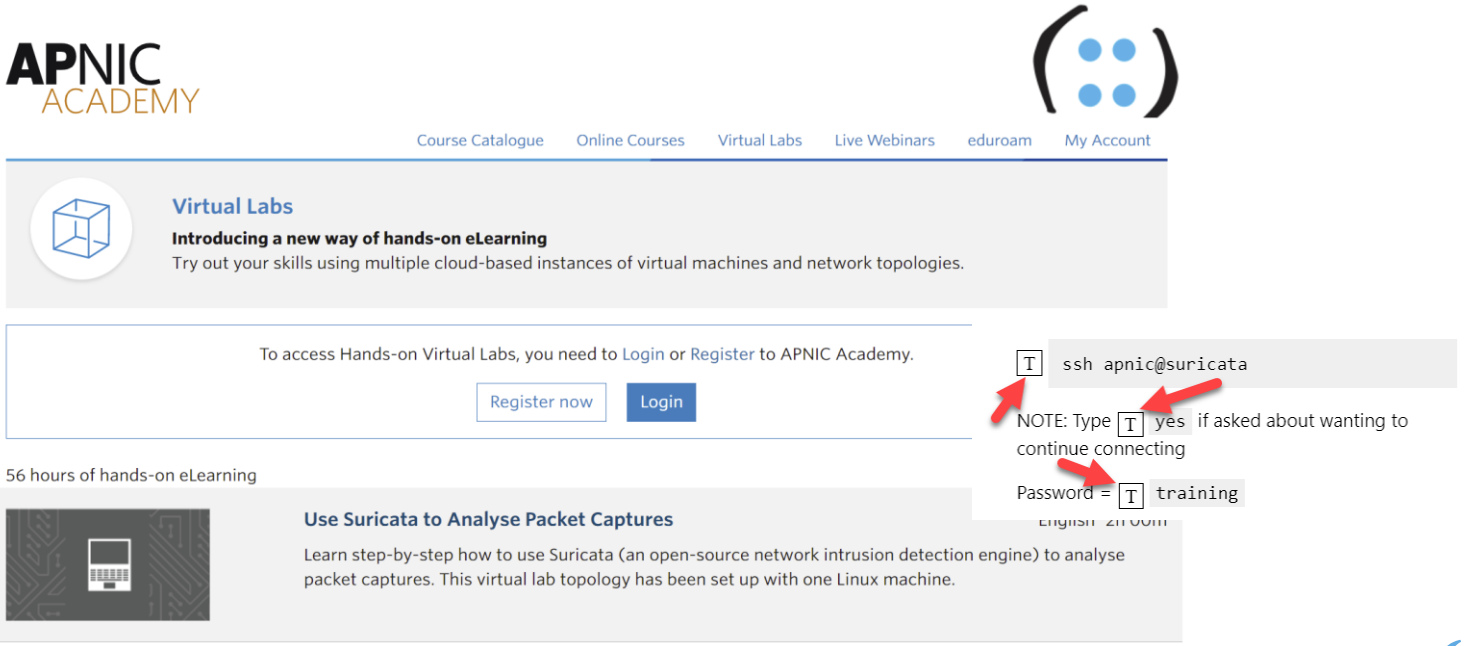
sudo systemctl status suricata

sudo systemctl stop suricata



Lab Overview

- Create an Account at APNIC Academy
  - https://academy.apnic.net/en/virtual-labs/?labId=97224



Lab1

- cd ~/workshop/lab1
- Wannacry / Eternalblue
  - Credits to Malware Traffic Analysis
    - https://www.malware-traffic-analysis.net/2017/05/18/index2.html
  - 2017-05-18-WannaCry-ransomware-using-EnternalBlue-exploit.pcap
  - What is in the PCAP

192.168.116.143 - a4:1f:72:20:54:01 - Windows 2012 R2 domain controller - TestDC1
192.168.116.150 - a4:1f:72:49:11:6d - Windows 2012 R2 server with a file share - WIN-2012-R2-1
192.168.116.138 - 00:19:bb:4f:4c:d8 - Windows 7 x64 - domain-joined workstation - DFIR_Win7_x64
192.168.116.149 - 00:25:b3:f5:fa:74 - Windows 7 x86 - domain-joined workstation - DFIR_Win7_x86 (wannacry launched here)
192.168.116.172 - 00:1c:c4:33:c6:dd - Windows 7 x86 - clone of DFIR_Win7_x86 - C-DFIR_Win7_x86

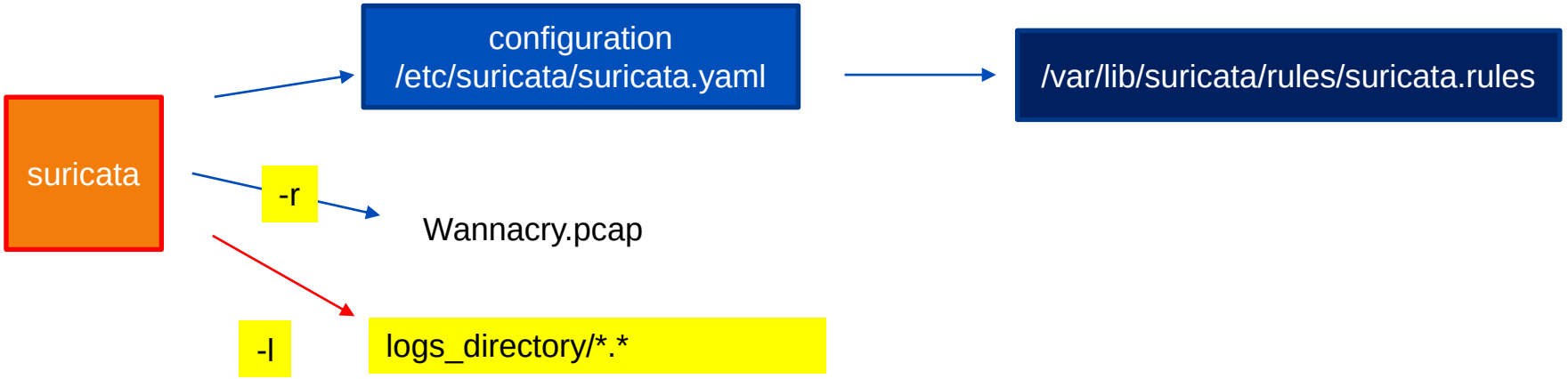
~/workshop/lab1

sudo suricata –r 2017-05-18-WannaCry-ransomware-using-EnternalBlue-exploit.pcap –l logs –k none

- The directory “log” was already created
  - Else the logs will be in /var/log/suricata/
- -k none (no checksum checking)

```
apnic@suricata:~/workshop/lab1$ suricata | egrep "offline|log|checksum"
        -r <path>                            : run in pcap file/offline mode
        -l <dir>                             : default log directory
        -k [all|none]                        : force checksum check (all) or disabled it (
none)
apnic@suricata:~/workshop/lab1$ sudo suricata -r 2017-05-18-WannaCry-ransomware-using-Ente
rnalBlue-exploit.pcap -l logs -k none
26/7/2021 -- 04:01:57 - <Notice> - This is Suricata version 6.0.3 RELEASE running in USER
mode
26/7/2021 -- 04:02:23 - <Notice> - all 5 packet processing threads, 4 management threads i
nitialized, engine started.
26/7/2021 -- 04:02:23 - <Notice> - Signal Received.  Stopping engine.
26/7/2021 -- 04:02:24 - <Notice> - Pcap-file module read 1 files, 46654 packets, 37044839
bytes
```

Behind the ‘scene’



If –l is not specified the logs will use configuration in /etc/suricata/suricata.yaml – normally /var/log/suricata
We can also specify which interface to monitor with –i eth0 (example) or specifying it in /etc/suricata/suricata.yaml

Parsing the logs (the hard way ☺)

- cd logs

```
-rw-r--r-- 1 root    root     54K Jun  9 09:48 eve.json
-rw-r--r-- 1 root    root    3.3K Jun  9 09:48 fast.log
-rw-r--r-- 1 root    root    2.4K Jun  9 09:48 stats.log
-rw-r--r-- 1 root    root    1.6K Jun  9 09:48 suricata.log
```

- less fast.log
  - (type Ctrl-C to get out of less ☺)

Parsing the logs (the hard way ☺) #2

- Inside ex1/logs
- cat eve.json | jq . | less
- cat eve.json | jq 'select (.event_type == "alert")' | less
- cat eve.json | jq 'select (.event_type == "smb")' | less


```
{
  "timestamp": "2017-05-18T18:06:21.120061+1000",
  "flow_id": 6188343065504,
  "pcap_cnt": 59,
  "event_type": "smb",
  "src_ip": "fe80:0000:0000:0000:ed6a:d848:6059:3c0e",
  "src_port": 49166,
  "dest_ip": "fe80:0000:0000:0000:6992:5661:9d0d:3f96",
  "dest_port": 445,
  "proto": "TCP",
  "smb": {
    "id": 6,
    "dialect": "2.10",
    "command": "SMB2_COMMAND_TREE_CONNECT",
    "status": "STATUS_SUCCESS",
    "status_code": "0x0",
    "session_id": 114349209288709,
    "tree_id": 5,
    "share": "\\\\WIN-2012-R2-1\\Users",
    "share_type": "FILE"
  }
}
```

What’s the story?

- Who is attacking who?
- Time
- Anything else?
- What was the vulnerability

Alerts & Signatures

- 05/18/2017-18:12:07.220702 [**] [1:2025649:3] **ET EXPLOIT Possible ETERNALBLUE Probe MS17-010 (MSF style)** [**] [Classification: A Network Trojan was detected] [Priority: 1] {TCP} 192.168.116.149:49368 -> 192.168.116.138:445
- 05/18/2017-18:12:11.553081 [**] [1:2025650:3] **ET EXPLOIT ETERNALBLUE Probe Vulnerable System Response** MS17-010 [**] [Classification: A Network Trojan was detected] [Priority: 1] {TCP} 192.168.116.172:445 -> 192.168.116.149:49444
- 05/18/2017-18:12:13.428436 [**] [1:2024217:3] **ET EXPLOIT Possible ETERNALBLUE MS17-010 Heap Spray** [**] [Classification: A Network Trojan was detected] [Priority: 1] {TCP} 192.168.116.149:49472 -> 192.168.116.138:445

cd /var/lib/suricata

- grep -i "ET EXPLOIT Possible ETERNALBLUE MS17-010 Heap Spray" suricata.rules
- alert smb any any -> $HOME_NET any (msg:"ET EXPLOIT Possible ETERNALBLUE MS17-010 Heap Spray"; flow:to_server,established; content:"|ff|SMB|33 00 00 00 00 18 07 c0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 08 ff fe 00 08|"; offset:4; depth:30; fast_pattern:10,20; content:"|00 09 00 00 00 10|"; distance:1; within:6; content:"|00 00 00 00 00 00 00 10|"; within:8; content:"|00 00 00 10|"; distance:4; within:4; pcre:"/^[a-zA-Z0-9+/]{1000,}/R"; threshold: type both, track by_src, count 3, seconds 30; metadata: former_category EXPLOIT; classtype:trojan-activity; sid:2024217; rev:3; metadata:attack_target SMB_Server, deployment Internal, signature_severity Critical, created_at 2017_04_17, updated_at 2017_05_13;)

Lab 1 - Take Aways

- IDS provide context
  - What is the story
  - Known or unknown
  - Protocol aware
- Suricata must see traffic
  - Network interface
  - Feed it packets / Pcap
- What is the difference between Wireshark / Tcpdump vs Suricata

About rules – suricata-update

- Rulesets
  - Commercial and free rules
  - By default Open Emerging Threats rules included

- List of rulesets
  - sudo suricata-update list-sources
  - sudo suricata-update enable-source _ruleset_name_
- Check signature.pdf for additional notes
- Update suricata-rules
  - suricata-update
- Automatic update
  - Use crontab to run suricata-update daily

https://readthedocs.org/projects/suricata-update/downloads/pdf/latest/

Lab 2 – Dridex

~/workshop/lab2

- cd ~/workshop/lab2
- sudo suricata -r 2019-07-09-password-protected-Word-doc-pushes-Dridex.pcap -l logs -k none
  - Pcap from this files : https://www.malware-traffic-analysis.net/2019/07/09/index.html
- We'll use Evebox to parse the eve.json

```
"timestamp": "2019-07-10T04:26:29.628847+1000",
 "flow_id": 1226676421106482,
 "pcap_cnt": 268,
 "event_type": "alert",
 "src_ip": "188.166.156.241",
 "src_port": 443,
 "dest_ip": "10.7.9.101",
 "dest_port": 49205,
 "proto": "TCP",
 "alert": {
   "action": "allowed",
   "gid": 1,
   "signature_id": 2023476,
   "rev": 5,
   "signature": "ET MALWARE ABUSE.CH SSL Blacklist Malicious
SSL certificate detected (Dridex)",
   "category": "A Network Trojan was detected",
   "severity": 1,
```

What does the signature look like?

alert tls $EXTERNAL_NET any -> $HOME_NET any (msg:"ET MALWARE ABUSE.CH SSL Blacklist Malicious SSL certificate detected (Dridex)"; flow:established,from_server; content:"|16|"; content:"|0b|"; within:8; byte_test:3,<,1200,0,relative; content:"|03 02 01 02 02 09 00|"; fast_pattern; content:"|30 09 06 03 55 04 06 13 02|"; distance:0; pcre:"/^[A-Z]{2}/R"; content:"|55 04 07|"; distance:0; content:"|55 04 0a|"; distance:0; pcre:"/^.{2}[A-Z][a-z]{3,}\s(?:[A-Z][a-z]{3,}\s)?(?:[A-Z](?:[A-Za-z]{0,4}?[A-Z]|(?:\.[A-Za-z]){1,3})|[A-Z]?[a-z]+|[a-z](?:\.[A-Za-z]){1,3})\.?[01]/Rs"; content:"|55 04 03|"; distance:0; byte_test:1,>,13,1,relative; content:!"www."; distance:2; within:4; pcre:"/^.{2}(?P<CN>(?:(?:\d?[A-Z]?|[A-Z]?\d?)(?:[a-z]{3,20}|[a-z]{3,6}[0-9_][a-z]{3,6})\.){0,2}?(?:\d?[A-Z]?|[A-Z]?\d?)[a-z]{3,}(?:[0-9_-][a-z]{3,})?\.(?!com|org|net|tv)[a-z]{2,9})[01].*?(?P=CN)[01]/Rs"; content:!"|2a 86 48 86 f7 0d 01 09 01|"; content:!"GoDaddy"; reference:url,sslbl.abuse.ch; classtype:trojan-activity; sid:2023476; rev:5; metadata:affected_product Windows_XP_Vista_7_8_10_Server_32_64_Bit, attack_target Client_Endpoint, created_at 2016_11_02, deployment Perimeter, performance_impact Low, signature_severity Major, tag SSL_Malicious_Cert, updated_at 2017_02_23;)

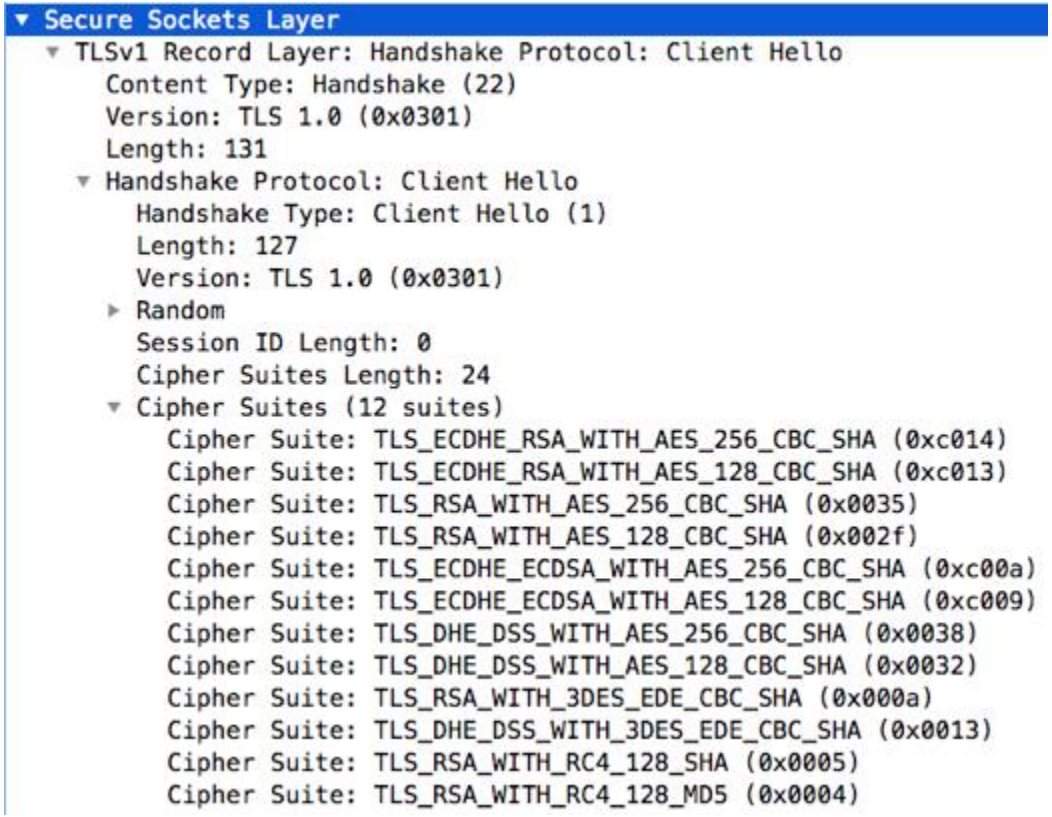
TLS / SSL client and server fingerprinting

- Encrypted communication
  - Suricata not able to perform full packet inspection / check application payload
  - Signature that goes into payload won't match
  - Some header information is still available
- Client and server TLS negotiation fingerprint
  - In a nutshell – TLS parameters combined -> md5
  - Both client and server
  - Confidence is in the manner response is in the same way
  - Multiple use-cases here
    - Client and server detection (malware, TOR, command and control, phishing sites etc)
- Ja3/ja3s developed by Salesforce
  - Integrated by Suricata
  - Abuse.ch has rulesets for TLS fingerprints
  - Ja3 = client , Ja3S = server
- Abuse.ch maintains an SSL blacklist repo
  - https://sslbl.abuse.ch/blacklist/
- Read more
  - https://engineering.salesforce.com/tls-fingerprinting-with-ja3-and-ja3s-247362855967
  - https://www.trustwave.com/en-us/resources/blogs/spiderlabs-blog/inspecting-encrypted-network-traffic-with-ja3/

https://github.com/salesforce/ja3


```
{
    "destination_ip": "188.166.156.241",
    "destination_port": 443,
    "ja3": "771,60-47-61-53-5-10-49191-49171-49172-49195-49187-49196-49188-49161-49162-64-50-106-56-19-4,65281-10-11-13,23-24,0",
    "ja3_digest": "74927e242d6c3febf8cb9cab10a7f889",
    "source_ip": "10.7.9.101",
    "source_port": 49205,
    "timestamp": 1562696789.50146
  },
  {
    "destination_ip": "188.166.156.241",
    "destination_port": 443,
    "ja3": "771,60-47-61-53-5-10-49191-49171-49172-49195-49187-49196-49188-49161-49162-64-50-106-56-19-4,65281-10-11-13,23-24,0",
    "ja3_digest": "74927e242d6c3febf8cb9cab10a7f889",
    "source_ip": "10.7.9.101",
    "source_port": 49206,
    "timestamp": 1562696791.350588
  },
```


Optional, install ja3 & run ja3 on the pcap
$ja3 -a --json *pcap



{
"destination_ip": "185.67.0.108",
"destination_port": 443,
"ja3": "769,49172-49171-53-47-49162-49161-56-50-10-19-5-4,0-5-10-11-65281,23-24-25,0",
"ja3_digest": "1eede9d19dc45c2cb66d2f5c6849e843",
"source_ip": "192.168.56.101",
"source_port": 49161,
"timestamp": 1527008276.377147
}

SSLVersion,Cipher,SSLExtension,EllipticCurve,EllipticCurvePointFormat - > string -> md5

echo –n “

Lab 2 - Take Aways

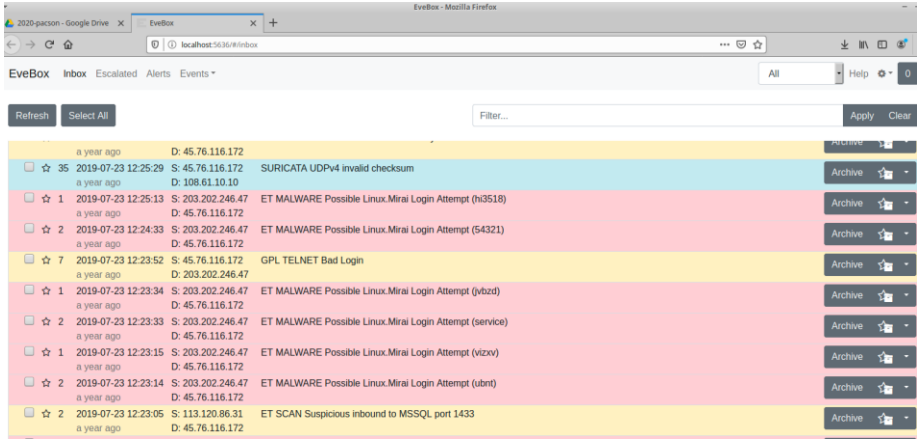
- Protocol analysis / Protocol aware
  - Do not have to specify the port
  - Web traffic not on port 80, ssh on 1337
  - Check out the supported protocols on Suricata documentation
    - https://suricata.io/features/all-features/
- Signature
  - Contents of packets, headers not just plain text payload
  - TLS/ja3/ja3s
  - HASSH (profile SSH)
    - https://github.com/salesforce/hassh

Lab3 – Honeypots

Lab3 ~/workshop/lab3

- Traffic from Cowrie Honeypot
  - APNIC Community Honeynet Project
  - Cowrie emulate ssh/telnet service
  - Interact with client, serves shell and log activities upon ‘successful login’
  - DDoS agent, miners, compromised IoTs

24 hour activity



Lab3

- Our honeypot IP
  - 45.76.116.172
  - We should include this into our configuration file $HOME_NETWORK
  - Signature has direction, flow as criteria
- sudo suricata -r cowrie.pcap -l logs/ -k none
  - Verify logs in /logs
- Evebox
  - **evebox oneshot logs/eve.json**
  - Let's Explore
    - SSH
    - Alerts

HOME_NET:
"[192.168.0.0/16,10.0.0.0/8,172.16.0.0/12,45.76.116.172/32]"

Lab 3 - Take Aways

- IP/Domain Reputation
  - ET CINS Active Threat Intelligence Poor Reputation IP group 87
  - OSINT (open source intel)
  - Why is our hosts talking to a suspicious host, TOR exit node, malicious domain
  - VirusTotal, Dshield
  - https://www.virustotal.com/gui/ip-address/92.118.160.57/relations
  - https://www.dshield.org/ipinfo.html?ip=92.118.160.57
- Suricata configuration
  - My network vs the world
- More Context
  - ALERT: ET MALWARE Possible Linux.Mirai Login Attempt (hi3518)
  - ET EXPLOIT HiSilicon DVR - Default Telnet Root Password Inbound

sudo less /var/lib/suricata/rules/suricata.rules | grep 2027973

- alert tcp $EXTERNAL_NET any -> $HOME_NET 23 (msg:"ET EXPLOIT HiSilicon DVR - Default Telnet Root Password Inbound"; flow:established,to_server; content:"xc3511"; fast_pattern; reference:url,github.com/tothi/pwn-hisilicon-dvr; classtype:default-login-attempt; sid:**2027973**; rev:2; metadata:affected_product DVR, attack_target IoT, created_at 2019_09_09, deployment Perimeter, former_category EXPLOIT, signature_severity Major, updated_at 2019_09_09;)

Lab 4 – File Extraction

Lab 4

- cd ~/workshop/lab4
- sudo suricata -r 20202904.pcap -l logs -k none
- *Evebox it* and explore
  - evebox oneshot logs/eve.json
- File extraction time!

File Extraction – Step 1

- Always read the docs!
  - https://suricata.readthedocs.io/en/suricata-6.0.3/file-extraction/file-extraction.html
  - Look for file-store in /etc/suricata/suricata.yaml
  - Use nano or vim to edit
    - Careful yaml files are sensitive

Default

```
- file-store:
      version: 2
      enabled: no
```

Change To

```
- file-store:
      version: 2
      enabled: yes
```

File Extraction Step 2

- Need to have as specific rule for this
- In lab4 folder create a file called extract.rule
- Copy the following rule:

  alert http any any -> any any (msg:"FILE store all"; filestore; sid:1; rev:1;)

- Paste in extract.rule

File Extraction: Step 3

- Run Suricata with the extraction.rule

  $sudo –r 20202904.pcap –l logs –S extract.rule –k none

- This should create a filestore directory inside logs
- Run the following to see the extracted files

  $file */*

- See sample output in next slide

file */*

04/049431fab0d3461408345dd7ed70f9be994dc99dd56af2cd99f35a183cb9d859: **XZ compressed data**
06/067609f84812a154ef6c246e8b8cfbd4c7f6bba49450418227fa2acf523bba7b: **ELF 64-bit LSB executable, x86-64, version 1 (SYSV), statically linked, stripped**
1b/1bc56ca730194475721509e13c54a32005a6db2919f07eeacc46945c6e4b667f**: ASCII text**
20/206ad6aca45f12e07ede8032137b7536648b2b79302ed559df00d397e816432c: **Bourne-Again shell script, ASCII text executable**
34/34ed56e258232ea0be2d79f3eca3d09147f46880ce86c9da0c0473a1060669aa: **XZ compressed data**
36/366e3fbe8767805b2efb5b0df28d151b806c7e140ae53c57cacd390af2df11cf**: XZ compressed data**
4a/4a8389496b4f0fb164444fbd36ecd4cf38c3dd62c2bf190f20d937e605c3db73**: ELF 32-bit MSB executable, MIPS, MIPS-I version 1 (SYSV), statically linked, not stripped**
69/690bd875ab5799394af7406f32f2e698c48e772ac367d520f74ec1095662da4d**: ASCII text**
97/97c42ab6f8c385658eed005846a9618d32fbaf4bcb5376ad863476ae21df8ffe: **XZ compressed data**
9e/9e3f14260ac0e015756468f191da390910c20117de8730e189613ca61429961a: **ASCII text**
cd/cd888ded56c3882397edf3c28376fee3134d45537826de9dc437cf57837d2c43: **Debian binary package (format 2.0)**
d0/d043d8bb305d66cf6b64b0e7ff48c84182c55697b91326f988eb9db0581eb831**: ASCII text**
da/da7d093bae12980e3e8b2e27318a6997831fbc5999d5d128942cb9b521aafcdf: **XZ compressed data**
e7/e7805f20300d402d030f1fef536b4267117f4d5bcc6c95664ff3adde93d88628**: Bourne-Again shell script, ASCII text executable**

Try

- In the filestore folder run the following command
- `less 20/206ad6aca45f12e07ede8032137b7536648b2b79302ed559df00d397e816432c`
- Go to virustotal.com
  - Search for **4a8389496b4f0fb164444fbd36ecd4cf38c3dd62c2bf190f20d937e605c3db73**

Take Aways

- File Extraction
  - Supported protocols http, smb, nfs, smtp, ftp
  - A few use cases – store all, certain md5sum, certain extension
  - Rule required to trigger extraction
- -S and –s
  - -S exclusively run this rule file ignore settings in suricata.yaml
  - -s run this rule file and the one mentioned in suricata.yaml
- Context
  - File reputation
  - Malware databases
  - Sites like virustotal.com and a few others can give context
  - Tools like TheHive allows to make query to multiple places
- Host Based IDS
  - Wazuh – file integrity checker
  - Better insight/context network AND host

Wazuh

›	Dec 3, 2020 @ 07:14:44.721	/etc/hosts	modified	Integrity checksum changed.	7	550
›	Dec 3, 2020 @ 07:14:44.703	/etc/cups/subscriptions.conf.O	modified	Integrity checksum changed.	7	550
›	Dec 2, 2020 @ 23:06:53.279	/etc/test	modified	Integrity checksum changed.	7	550

t	agent.id	001
t	agent.ip	10.0.2.4
t	agent.name	osboxes
t	decoder.name	syscheck_integrity_changed
t	full_log	File '/etc/hosts' modified Mode: whodata Changed attributes: size,mtime,inode,md5,sha1,sha256 Size changed from '300' to '289' Old modification time was: '1606913905', now it is '1606943464' Old inode was: '666413', now it is '666686' Old md5sum was: '87e08d6373633640e86bc3245ae6b8ef' New md5sum is : '469d03062b0040eeec2099cd186b629c' Old sha1sum was: '95d575350dde6614dbed0cdc3f297270d125a745' New sha1sum is : '2e51d84da6d581b88f574523bff580622097462b' Old sha256sum was: '53a3753652e533711d9863645563eaa51dba77181fd1b404f9290980201ee070' New sha256sum is : '9ddca8a95247329d1a652fe4b1932fb9493dfec5bbe89b520e57961b0e02d262'
t	id	1606943684.83166639

t	syscheck.changed_attributes	size, mtime, inode, md5, sha1, sha256
t	syscheck.diff	4d3 < 7.7.7.7 vpn.my.home.xyz 6a6,7 > 5.5.5.5 test >
t	syscheck.event	modified
t	syscheck.gid_after	0

https://github.com/wazuh/wazuh

Part 2 - Writing Suricata Signatures

Overview

- Suricata can dissect packets & is protocol aware
- Rules are applied against content of packets & protocol related information
  - do something (alert) if packet contains the word “ransom=“
  - do something (alert) if http traffic & http method is POST and content of packet has the word “ransom”
  - do something (alert) if there is dns request
  - do something (drop) if there is dns request and the record asked is ‘ransomware.com’

General Workflow

- Traffic
  - PCAP
  - Generate network activity on host
    - ping hostname.com
    - dig A apnic.net
    - wget http://www.testmyids.com
- Malware Analysis
  - Look for interesting strings & behavior
  - User agent, check-in command, file names, etc
  - Malware-traffic-analysis.net
- Get indicators of compromise from reports, advisories, threat sharing platform (i.e. MISP)
- Write signature
- Test signature
  - suricata -S rulefile –r file.pcap –l logdirectory –k none
  - suricata –S rulefile –i eth0 –l logdirectory –k none
  - -S load signature file exclusively
- Additional notes
  - Take note of the ip address to include in $HOME_NET in /etc/suricata/suricata.yaml
  - Check pcap or interface

Signature writing use cases

- You know what you're looking for
  - Write specific rule
- Payload specific
  - Alert if packet contains xyz
  - Alert if packet contains IP address from known IOCs
  - Alert if there is telnet traffic to IP address in country XYZ
  - Alert if TLS fingerprint == XYZ
- Not payload specific
  - Alert if there is outbound ssh traffic
  - Alert if EXTERNAL_NET is a Tor Exit Node

Read the Docs

- Beware of out-of-date tutorials on the Internet

https://suricata.readthedocs.io/en/suricata-6.0.2/file-extraction/file-extraction.html

Rule Format

- There's 3 parts to it

Action Header (Options)

alert tcp $HOME_NET any -> $EXTERNAL_NET any (msg: "AW Suspicious Traffic "; sid:99999; rev:1;

- Action – what happens when signature matches
- Header – defines protocol, IP address, ports and direction of the rule
- Rule Options – define the specifics of the rule. Can be specific to the protocol/action/payloads or metadata.

Hello World Rule

Save the following rule into a file called myrule

```
alert tcp any any -> any any (msg:”Hello World TCP”; sid:202010000;
rev:1;)
```

sudo suricata –S myrule -i eth0 -k none –l logs

cat logs/fast.log

evebox one shot logs/eve.json

Hello World ICMP or UDP

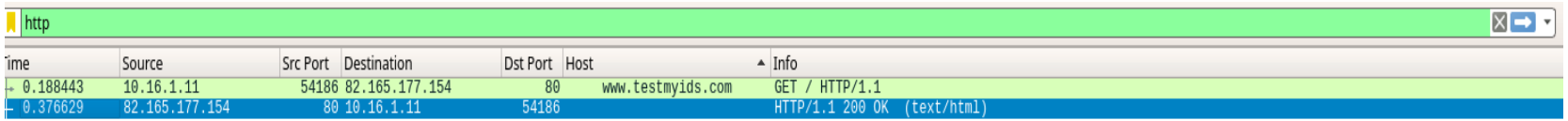
```
alert ___ any any -> any any (msg:”AW Hello World ICMP”;
sid:202010001; rev:1;)
```

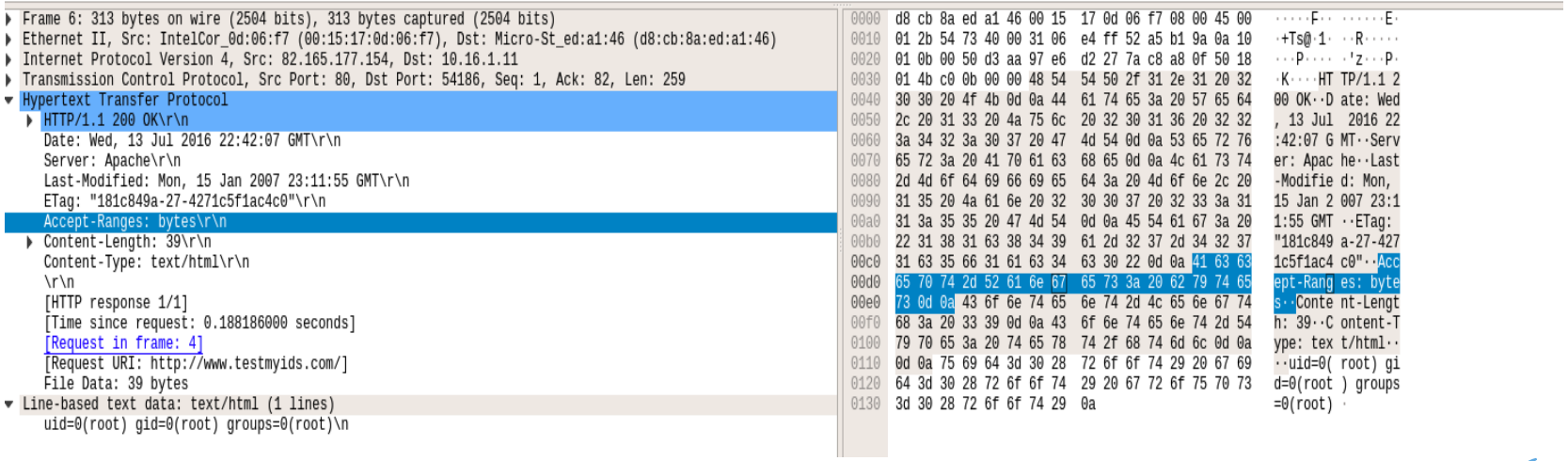
```
alert ___ $HOME_NETWORK any -> $EXTERNAL_NETWORK any (msg:”AW
Hello World UDP”; sid:202010002; rev:1;)
```

Lab 5 – Signature

cd ~/workshop/lab5

- Optional: open testmyidsPCAP in Wireshark
- Check content of http


http
Time Source Src Port Destination Dst Port Host Info
0.188443 10.16.1.11 54186 82.165.177.154 80 www.testmyids.com GET / HTTP/1.1
0.376629 82.165.177.154 80 10.16.1.11 54186 HTTP/1.1 200 OK (text/html)


Frame 6: 313 bytes on wire (2504 bits), 313 bytes captured (2504 bits)
Ethernet II, Src: IntelCor_0d:06:f7 (00:15:17:0d:06:f7), Dst: Micro-St_ed:a1:46 (d8:cb:8a:ed:a1:46)
Internet Protocol Version 4, Src: 82.165.177.154, Dst: 10.16.1.11
Transmission Control Protocol, Src Port: 80, Dst Port: 54186, Seq: 1, Ack: 82, Len: 259
Hypertext Transfer Protocol
HTTP/1.1 200 OK\r\n
Date: Wed, 13 Jul 2016 22:42:07 GMT\r\n
Server: Apache\r\n
Last-Modified: Mon, 15 Jan 2007 23:11:55 GMT\r\n
ETag: "181c849a-27-4271c5f1ac4c0"\r\n
Accept-Ranges: bytes\r\n
Content-Length: 39\r\n
Content-Type: text/html\r\n
\r\n
[HTTP response 1/1]
[Time since request: 0.188186000 seconds]
[Request in frame: 4]
[Request URI: http://www.testmyids.com/]
File Data: 39 bytes
Line-based text data: text/html (1 lines)
uid=0(root) gid=0(root) groups=0(root)\n
0000 d8 cb 8a ed a1 46 00 15 17 0d 06 f7 08 00 45 00 .....F.. ......E.
0010 01 2b 54 73 40 00 31 06 e4 ff 52 a5 b1 9a 0a 10 .+Ts@.1. ..R.....
0020 01 0b 00 50 d3 aa 97 e6 d2 27 7a c8 a8 0f 50 18 ...P.... .'z...P.
0030 01 4b c0 0b 00 00 48 54 54 50 2f 31 2e 31 20 32 .K....HT TP/1.1 2
0040 30 30 20 4f 4b 0d 0a 44 61 74 65 3a 20 57 65 64 00 OK..D ate: Wed
0050 2c 20 31 33 20 4a 75 6c 20 32 30 31 36 20 32 32 , 13 Jul 2016 22
0060 3a 34 32 3a 30 37 20 47 4d 54 0d 0a 53 65 72 76 :42:07 G MT..Serv
0070 65 72 3a 20 41 70 61 63 68 65 0d 0a 4c 61 73 74 er: Apac he..Last
0080 2d 4d 6f 64 69 66 69 65 64 3a 20 4d 6f 6e 2c 20 -Modifie d: Mon,
0090 31 35 20 4a 61 6e 20 32 30 30 37 20 32 33 3a 31 15 Jan 2 007 23:1
00a0 31 3a 35 35 20 47 4d 54 0d 0a 45 54 61 67 3a 20 1:55 GMT ..ETag:
00b0 22 31 38 31 63 38 34 39 61 2d 32 37 2d 34 32 37 "181c849 a-27-427
00c0 31 63 35 66 31 61 63 34 63 30 22 0d 0a 41 63 63 1c5f1ac4 c0"..Acc
00d0 65 70 74 2d 52 61 6e 67 65 73 3a 20 62 79 74 65 ept-Rang es: byte
00e0 73 0d 0a 43 6f 6e 74 65 6e 74 2d 4c 65 6e 67 74 s..Conte nt-Lengt
00f0 68 3a 20 33 39 0d 0a 43 6f 6e 74 65 6e 74 2d 54 h: 39..C ontent-T
0100 79 70 65 3a 20 74 65 78 74 2f 68 74 6d 6c 0d 0a ype: tex t/html..
0110 0d 0a 75 69 64 3d 30 28 72 6f 6f 74 29 20 67 69 ..uid=0(root) gi
0120 64 3d 30 28 72 6f 6f 74 29 20 67 72 6f 75 70 73 d=0(root) groups
0130 3d 30 28 72 6f 6f 74 29 0a =0(root) .

content: uid=0(root) gid=0(root) groups=0(root)

content: www.testmyids.com

(External) IP Address: 82.165.177.154

Basic Rule

alert http any any -> any any (msg:"Suspicious Root Privilege"; content: "uid=0(root) gid=0(root) groups=0(root)"; sid:2020100004; rev:1;)

Take Aways

- Signature should be specific
  - Use parameters, pattern matching etc
  - Think of false positives
  - Alert fatigue, Counter productive
  - Validate the IOCs thoroughly

```
/var/lib/suricata/rules/classification.config
config classification: unknown,Unknown Traffic,3
config classification: bad-unknown,Potentially Bad Traffic, 2
config classification: attempted-recon,Attempted Information Leak,2
config classification: successful-recon-limited,Information Leak,2
config classification: successful-recon-largescale,Large Scale Information Leak,2
config classification: attempted-dos,Attempted Denial of Service,2
config classification: successful-dos,Denial of Service,2
config classification: attempted-user,Attempted User Privilege Gain,1
```

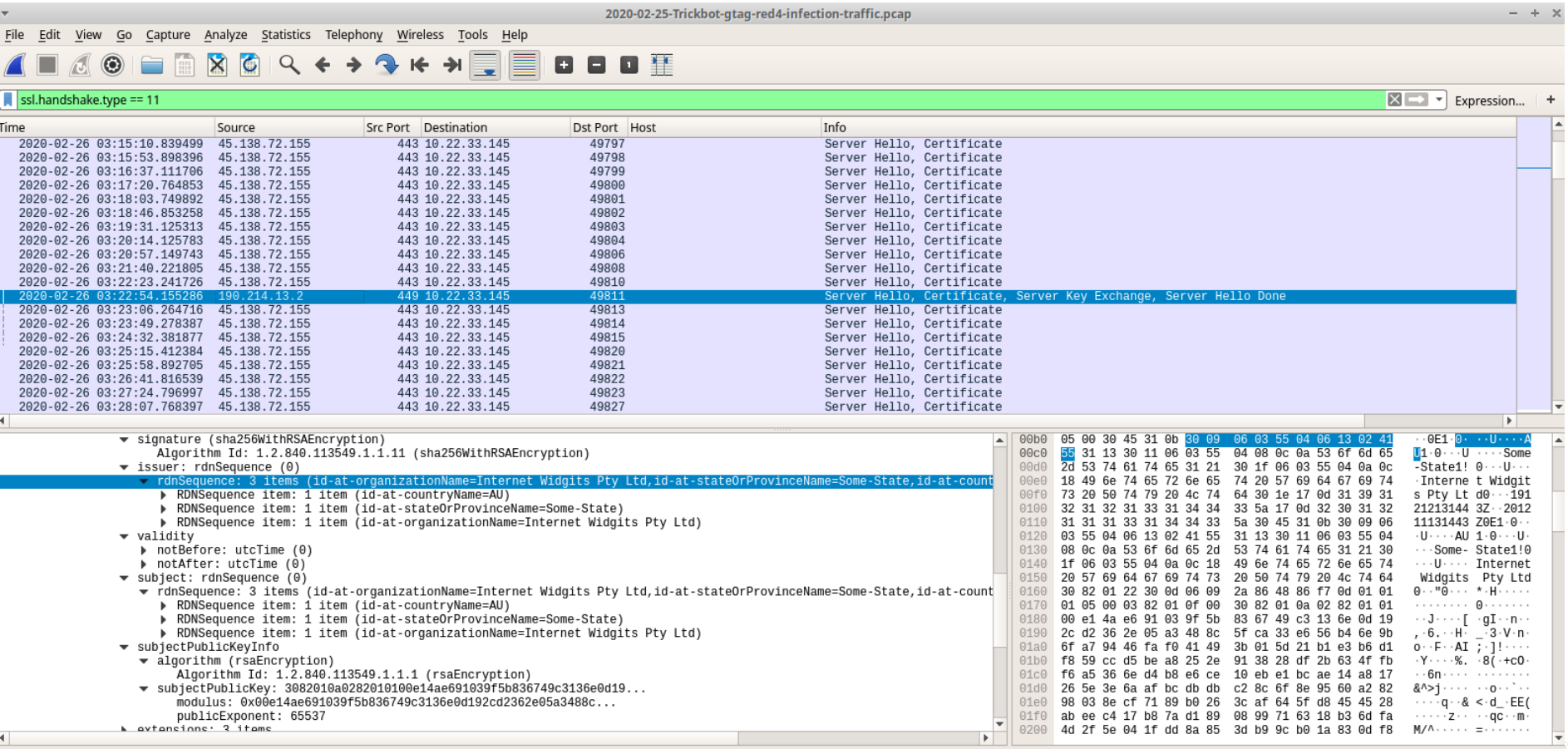
- Classification and priority
  - Classification
  - Classification.config
  - The classification.config file includes information for prioritizing rules
  - Any rule can override it

Lab 6 - Trickbot

Lab6

- Trickbot
  - https://attack.mitre.org/software/S0266/
  - https://unit42.paloaltonetworks.com/wireshark-tutorial-examining-trickbot-infections/
  - Pcap 2020-02 from Malware-traffic-analysis.net - https://www.malware-traffic-analysis.net/2020/02/25/index.html
- Uses TLS/SSL to communicate with server on port 449 and 447

TLS activities over port 447 – Internet Widgits Pty LTD



“Global Security” and “IT Department”

ssl.handshake.type == 11 and tcp.port == 447

Time	Source	Src Port	Destination	Dst Port	Host	Info
2020-02-26 03:28:36.786937	5.2.77.18	447	10.22.33.145	49834		Server Hello, Certificate, Server Key Exchange, Server Hello Done
2020-02-26 03:57:33.344929	5.2.77.18	447	10.22.33.145	49709		Server Hello, Certificate, Server Key Exchange, Server Hello Done
2020-02-26 04:11:29.694310	5.2.77.18	447	10.22.33.145	49764		Server Hello, Certificate, Server Key Exchange, Server Hello Done
2020-02-26 04:30:30.432909	66.85.173.20	447	10.22.33.145	50057		Server Hello, Certificate, Server Key Exchange, Server Hello Done
2020-02-26 04:39:02.444681	66.85.173.20	447	10.22.33.145	50062		Server Hello, Certificate, Server Key Exchange, Server Hello Done

```
    issuer: rdnSequence (0)
    validity
    subject: rdnSequence (0)
      rdnSequence: 6 items (id-at-commonName=example.com,id-at-organizationalUnitName=IT Department,id-at-organizationName=G
        RDNSequence item: 1 item (id-at-countryName=GB)
          RelativeDistinguishedName item (id-at-countryName=GB)
            Id: 2.5.4.6 (id-at-countryName)
            CountryName: GB
        RDNSequence item: 1 item (id-at-stateOrProvinceName=London)
          RelativeDistinguishedName item (id-at-stateOrProvinceName=London)
            Id: 2.5.4.8 (id-at-stateOrProvinceName)
            DirectoryString: uTF8String (4)
              uTF8String: London
        RDNSequence item: 1 item (id-at-localityName=London)
          RelativeDistinguishedName item (id-at-localityName=London)
            Id: 2.5.4.7 (id-at-localityName)
            DirectoryString: uTF8String (4)
              uTF8String: London
        RDNSequence item: 1 item (id-at-organizationName=Global Security)
          RelativeDistinguishedName item (id-at-organizationName=Global Security)
            Id: 2.5.4.10 (id-at-organizationName)
            DirectoryString: uTF8String (4)
              uTF8String: Global Security
        RDNSequence item: 1 item (id-at-organizationalUnitName=IT Department)
          RelativeDistinguishedName item (id-at-organizationalUnitName=IT Department)
            Id: 2.5.4.11 (id-at-organizationalUnitName)
            DirectoryString: uTF8String (4)
              uTF8String: IT Department
        RDNSequence item: 1 item (id-at-commonName=example.com)
          RelativeDistinguishedName item (id-at-commonName=example.com)
            Id: 2.5.4.3 (id-at-commonName)
            DirectoryString: uTF8String (4)
              uTF8String: example.com
    subjectPublicKeyInfo
    extensions: 3 items
  algorithmIdentifier (sha256WithRSAEncryption)
  Padding: 0
  encrypted: 683474e7cea5418cbbccffffecec146146e8e9eb7f43efaf...
TLSv1.2 Record Layer: Handshake Protocol: Server Key Exchange
```

```
0000  00 08 02 1c 47 ae 20 e5  2a b6 93 f1 08 00 45 00   ····G· · *·····E·
0010  05 99 4c d0 00 00 80 06  6a d4 05 02 4d 12 0a 16   ··L····· j···M···
0020  21 91 01 bf c2 aa 62 0e  cc 2b d7 d3 fd 42 50 18   !·····b· ·+···BP·
0030  fa f0 6c 49 00 00 16 03  03 00 3d 02 00 00 39 03   ··lI···· ··=···9·
0040  03 36 b1 9d 67 b8 6e f8  cc 45 dd 11 7e 31 7e 76   ·6··g·n· ·E··~1~v
0050  35 3e 55 95 08 48 f5 5b  b5 70 0e 44 6c 7e 2a cf   5>U··H·[ ·p·Dl~*·
0060  83 00 c0 30 00 00 11 ff  01 00 01 00 00 0b 00 04   ···0···· ········
0070  03 00 01 02 00 23 00 00  16 03 03 03 cf 0b 00 03   ·····#·· ········
0080  cb 00 03 c8 00 03 c5 30  82 03 c1 30 82 02 a9 a0   ·······0 ···0····
0090  03 02 01 02 02 09 00 b5  1a a5 df 97 f8 e9 55 30   ········ ······U0
00a0  0d 06 09 2a 86 48 86 f7  0d 01 01 0b 05 00 30 77   ···*·H·· ······0w
00b0  31 0b 30 09 06 03 55 04  06 13 02 47 42 31 0f 30   1·0···U· ···GB1·0
00c0  0d 06 03 55 04 08 0c 06  4c 6f 6e 64 6f 6e 31 0f   ···U···· London1·
00d0  30 0d 06 03 55 04 07 0c  06 4c 6f 6e 64 6f 6e 31   0···U··· ·London1
00e0  18 30 16 06 03 55 04 0a  0c 0f 47 6c 6f 62 61 6c   ·0···U·· ··Global
00f0  20 53 65 63 75 72 69 74  79 31 16 30 14 06 03 55    Securit y1·0···U
0100  04 0b 0c 0d 49 54 20 44  65 70 61 72 74 6d 65 6e   ····IT D epartmen
0110  74 31 14 30 12 06 03 55  04 03 0c 0b 65 78 61 6d   t1·0···U ····exam
0120  70 6c 65 2e 63 6f 6d 30  1e 17 0d 32 30 30 31 33   ple.com0 ···20013
0130  30 31 39 32 30 34 37 5a  17 0d 32 31 30 31 32 39   0192047Z ··210129
0140  31 39 32 30 34 37 5a 30  77 31 0b 30 09 06 03 55   192047Z0 w1·0···U
0150  04 06 13 02 47 42 31 0f  30 0d 06 03 55 04 08 0c   ····GB1· 0···U···
0160  06 4c 6f 6e 64 6f 6e 31  0f 30 0d 06 03 55 04 07   ·London1 ·0···U··
0170  0c 06 4c 6f 6e 64 6f 6e  31 18 30 16 06 03 55 04   ··London 1·0···U·
0180  0a 0c 0f 47 6c 6f 62 61  6c 20 53 65 63 75 72 69   ···Globa l Securi
0190  74 79 31 16 30 14 06 03  55 04 0b 0c 0d 49 54 20   ty1·0··· U····IT
01a0  44 65 70 61 72 74 6d 65  6e 74 31 14 30 12 06 03   Departme nt1·0···
01b0  55 04 03 0c 0b 65 78 61  6d 70 6c 65 2e 63 6f 6d   U····exa mple.com
01c0  30 82 01 22 30 0d 06 09  2a 86 48 86 f7 0d 01 01   0··"0··· *·H·····
01d0  01 05 00 03 82 01 0f 00  30 82 01 0a 02 82 01 01   ········ 0·······
01e0  00 ba 6c ea e6 a7 59 bb  d8 ab 06 e4 01 1c 91 60   ··l···Y· ·······`
01f0  76 ca a1 40 2f e7 6a a5  21 7a e8 47 25 62 0d 36   v··@/·j· !z·G%b·6
0200  c8 28 e4 c7 23 49 42 22  6b ec 09 12 d5 19 79 96   ·(··#IB" k·····y·
0210  8d fe 01 d7 f2 b6 e6 18  81 d7 44 27 7e cb 14 48   ········ ··D'~··H
0220  6b 07 84 99 76 fb 3c c1  7f b8 c7 9f e4 cd cf d0   k···v·<· ········
0230  95 a4 0f 28 d2 7a 97 31  03 52 51 bd 8a 98 d8 92   ···(·z·1 ·RQ·····
0240  ca 2e e7 fd 59 01 39 20  14 57 ed e3 f2 d0 ca 8c   ·.··Y·9  ·W······
0250  27 91 60 3f d2 44 97 11  7c ef 5e 91 eb c6 2f 6e   '·`?·D·· |·^···/n
0260  23 f8 2b 8a c9 a2 8a 28  d3 5d 43 42 7d 90 dd e9   #·+····( ·]CB}···
```

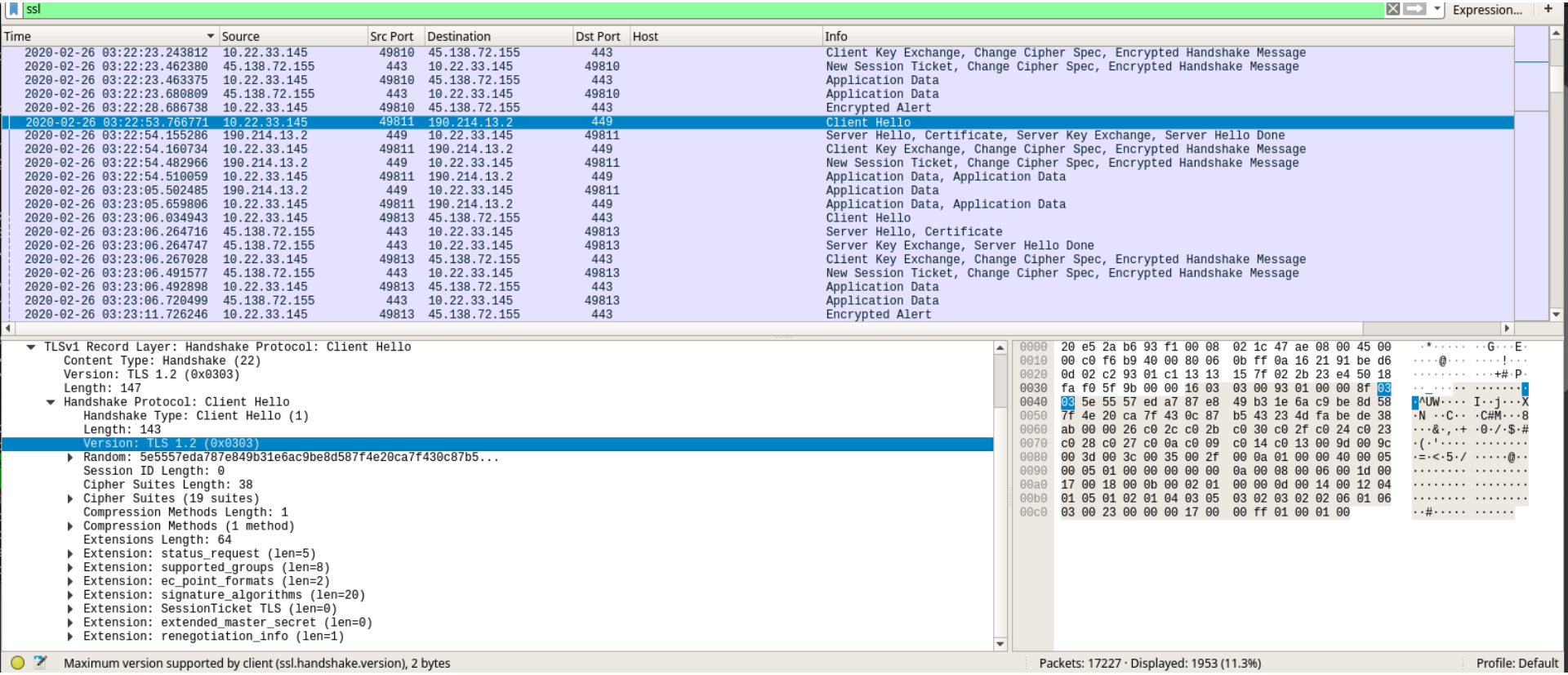
Rules

```
alert tls $EXTERNAL_NET any -> $HOME_NET any (msg:"
Selfsigned  TLS Certificate"; tls.cert_issuer; content:"Internet
Widgits Pty Ltd"; sid:10008; rev:1;)
```

```
alert tls $EXTERNAL_NET any -> $HOME_NET any
(msg:"Trickbot CNC"; tls.cert_issuer; content:"Global
Security"; content:"IT Department"; sid:10009; rev:1;)
```

ja3 and ja3s

- Ja3 – method for profiling TLS/SSL clients
  - Clients - browser agents, malware, etc
  - Internal -> External
- Ja3s – method for profiling TLS/SSL servers
  - Command and Control, Services, Websites
- Fingerprints are based on configurations and details of TLS/SSL handshakes*
  - Not encrypted i.e ClientHello
  - SSLVersion,Cipher,SSLExtension,EllipticCurve,EllipticCurvePointFormat
  - 769,4-5-10-9-100-98-3-6-19-18-99,,,
  - 769,4-5-10-9-100-98-3-6-19-18-99,,, --> de350869b8c85de67a350c8d186f11e6
- https://engineering.salesforce.com/tls-fingerprinting-with-ja3-and-ja3s-247362855967


ssl
Expression...
Time Source Src Port Destination Dst Port Host Info
2020-02-26 03:22:23.243812 10.22.33.145 49810 45.138.72.155 443 Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
2020-02-26 03:22:23.462380 45.138.72.155 443 10.22.33.145 49810 New Session Ticket, Change Cipher Spec, Encrypted Handshake Message
2020-02-26 03:22:23.463375 10.22.33.145 49810 45.138.72.155 443 Application Data
2020-02-26 03:22:23.680809 45.138.72.155 443 10.22.33.145 49810 Application Data
2020-02-26 03:22:28.686738 10.22.33.145 49810 45.138.72.155 443 Encrypted Alert
2020-02-26 03:22:53.766771 10.22.33.145 49811 190.214.13.2 449 Client Hello
2020-02-26 03:22:54.155286 190.214.13.2 449 10.22.33.145 49811 Server Hello, Certificate, Server Key Exchange, Server Hello Done
2020-02-26 03:22:54.160734 10.22.33.145 49811 190.214.13.2 449 Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
2020-02-26 03:22:54.482966 190.214.13.2 449 10.22.33.145 49811 New Session Ticket, Change Cipher Spec, Encrypted Handshake Message
2020-02-26 03:22:54.510059 10.22.33.145 49811 190.214.13.2 449 Application Data, Application Data
2020-02-26 03:23:05.502485 190.214.13.2 449 10.22.33.145 49811 Application Data
2020-02-26 03:23:05.659806 10.22.33.145 49811 190.214.13.2 449 Application Data, Application Data
2020-02-26 03:23:06.034943 10.22.33.145 49813 45.138.72.155 443 Client Hello
2020-02-26 03:23:06.264716 45.138.72.155 443 10.22.33.145 49813 Server Hello, Certificate
2020-02-26 03:23:06.264747 45.138.72.155 443 10.22.33.145 49813 Server Key Exchange, Server Hello Done
2020-02-26 03:23:06.267028 10.22.33.145 49813 45.138.72.155 443 Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
2020-02-26 03:23:06.491577 45.138.72.155 443 10.22.33.145 49813 New Session Ticket, Change Cipher Spec, Encrypted Handshake Message
2020-02-26 03:23:06.492898 10.22.33.145 49813 45.138.72.155 443 Application Data
2020-02-26 03:23:06.720499 45.138.72.155 443 10.22.33.145 49813 Application Data
2020-02-26 03:23:11.726246 10.22.33.145 49813 45.138.72.155 443 Encrypted Alert
TLSv1 Record Layer: Handshake Protocol: Client Hello
Content Type: Handshake (22)
Version: TLS 1.2 (0x0303)
Length: 147
Handshake Protocol: Client Hello
Handshake Type: Client Hello (1)
Length: 143
Version: TLS 1.2 (0x0303)
Random: 5e5557eda787e849b31e6ac9be8d587f4e20ca7f430c87b5...
Session ID Length: 0
Cipher Suites Length: 38
Cipher Suites (19 suites)
Compression Methods Length: 1
Compression Methods (1 method)
Extensions Length: 64
Extension: status_request (len=5)
Extension: supported_groups (len=8)
Extension: ec_point_formats (len=2)
Extension: signature_algorithms (len=20)
Extension: SessionTicket TLS (len=0)
Extension: extended_master_secret (len=0)
Extension: renegotiation_info (len=1)
0000 20 e5 2a b6 93 f1 00 08 02 1c 47 ae 08 00 45 00
0010 00 c0 f6 b9 40 00 80 06 0b ff 0a 16 21 91 be d6
0020 0d 02 c2 93 01 c1 13 13 15 7f 02 2b 23 e4 50 18
0030 fa f0 5f 9b 00 00 16 03 03 00 93 01 00 00 8f 03
0040 03 5e 55 57 ed a7 87 e8 49 b3 1e 6a c9 be 8d 58
0050 7f 4e 20 ca 7f 43 0c 87 b5 43 23 4d fa be de 38
0060 ab 00 00 26 c0 2c c0 2b c0 30 c0 2f c0 24 c0 23
0070 c0 28 c0 27 c0 0a c0 09 c0 14 c0 13 00 9d 00 9c
0080 00 3d 00 3c 00 35 00 2f 00 0a 01 00 00 40 00 05
0090 00 05 01 00 00 00 00 00 0a 00 08 00 06 00 1d 00
00a0 17 00 18 00 0b 00 02 01 00 00 0d 00 14 00 12 04
00b0 01 05 01 02 01 04 03 05 03 02 03 02 02 06 01 06
00c0 03 00 23 00 00 00 17 00 00 ff 01 00 01 00
Maximum version supported by client (ssl.handshake.version), 2 bytes
Packets: 17227 · Displayed: 1953 (11.3%)
Profile: Default

Ja3

- Generate ja3 hash with ja3 tool
  - https://github.com/salesforce/ja3
  - Check out ja3.json in your folder
- Let’s search for the client information on ja3er
  - 3b5074b1b5d032e5620f69f9f700ff0e
  - 72a589da586844d7f0818ce684948eea
- Check out more information here: https://ja3er.com/

Rule

```
alert tls $HOME_NET any -> $EXTERNAL_NET any (msg:"AW -
Trickbot Infection"; ja3.hash;
content:"72a589da586844d7f0818ce684948eea";  sid:10010;
rev:1; )
```

Integration with MISP

MISP Detection with Suricata

- MISP is a threat intel sharing platform
  - Check out APNIC Webinar on Practical Threat sharing
  - Allows community to share threats attributes (indicators)
  - APNIC runs a MISP instance
- Concept
  - From attributes/indicators -> generate Suricata rules automatically
- Export relevant indicators as Suricata Rules
  - Download rules formatted to work with Suricata IDS
  - Feed it to Suricata
  - Get alerts

Solarwinds Example – event from another MISP instance/feed

OSINT Threat Advisory: SolarWinds supply chain attack

Event ID	1358
UUID	e6d2f7c9-c183-43c9-bd3c-3dcfbb34665c
Creator org	CIRCL
Tags	type:OSINT, osint:lifetime="perpetual", osint:certainty="50", tlp:white
Date	2020-12-15
Threat Level	High
Analysis	Completed
Distribution	All communities
Info	OSINT Threat Advisory: SolarWinds supply chain attack
Published	Yes (2021-01-13 01:23:05)
#Attributes	54 (10 Objects)
First recorded change	2020-12-15 08:16:20
Last change	2020-12-15 08:48:11
Modification map	
Extends	632aaf17-44db-4c3e-bf97-59820990491a
Sightings	0 (0) - restricted to own organisation only.

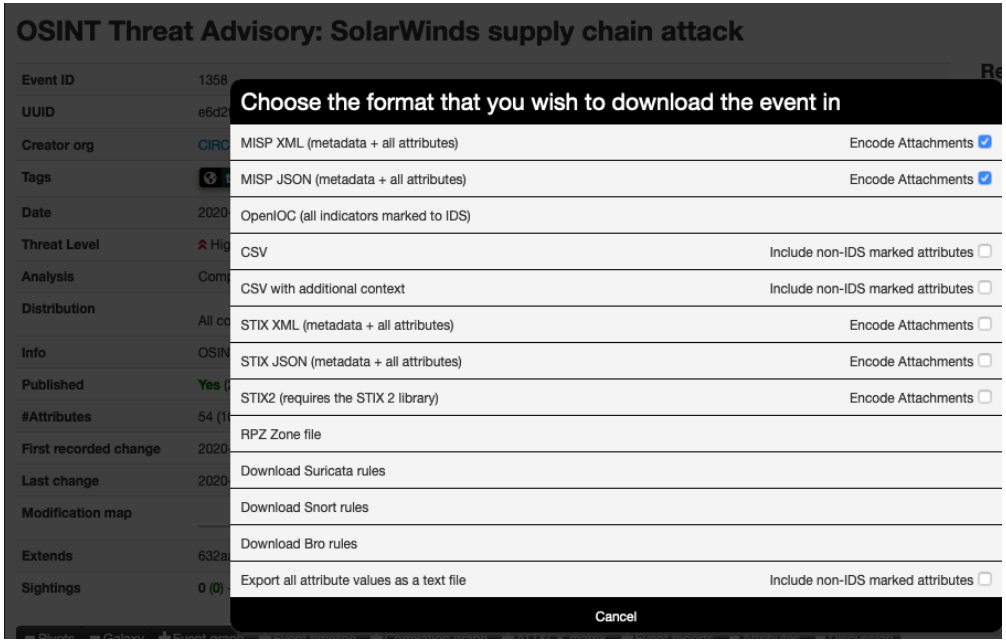
Related Events

OSINT - UNC2452 / SUNBURST @vxunderground OSINT related findings
2020-12-14 12

Solarwinds - Indicators (domain)

2020-12-15	Network activity	domain	databasegalore.com			Inherit	(0/0/0)	
2020-12-15	Network activity	domain	incomeupdate.com			Inherit	(0/0/0)	
2020-12-15	Network activity	domain	highdatabase.com			Inherit	(0/0/0)	
2020-12-15	Network activity	domain	websitetheme.com			Inherit	(0/0/0)	
2020-12-15	Network activity	domain	deftsecurity.com			Inherit	(0/0/0)	
2020-12-15	Network activity	domain	virtualdataserver.com			Inherit	(0/0/0)	
2020-12-15	Network activity	domain	thedoccloud.com			Inherit	(0/0/0)	
2020-12-15	Network activity	domain	digitalcollege.org			Inherit	(0/0/0)	
2020-12-15	Network activity	domain	globalnetworkissues.com			Inherit	(0/0/0)	
2020-12-15	Network activity	domain	seobundlekit.com			Inherit	(0/0/0)	
2020-12-15	Network activity	domain	virtualwebdata.com			Inherit	(0/0/0)	
2020-12-15	Network activity	domain	avsvmcloud.com			Inherit	(0/0/0)	

Export to Suricata rules



Fetch Rules from the event

```
curl -X POST -k -H 'Accept: application/json' -H
'Authorization:  [API Key]' -H 'Content-Type:
application/json'  -o misp.suricata.rules
'https://misp.honeynet.asia/attributes/restSearch' --data

'{"eventid":"1358", "returnFormat":"suricata","to_ids":"1"}'
```

Suricata Rules generated (snip)

```
alert dns any any -> any any (msg: "MISP e1358 [] Domain avsvmcloud.com";
dns_query; content:"avsvmcloud.com"; nocase; pcre: "/(^|[^A-Za-z0-9-
])avsvmcloud\.com$/i";  classtype:trojan-activity; sid:9823577; rev:1;
priority:1; reference:url,https://misp.honeynet.asia/events/view/1358;)

alert http $HOME_NET any -> $EXTERNAL_NET any (msg: "MISP e1358 [] Outgoing HTTP
Domain avsvmcloud.com"; flow:to_server,established; content: "Host|3a|"; nocase;
http_header; content:"avsvmcloud.com"; fast_pattern; nocase; http_header; pcre:
"/(^|[^A-Za-z0-9-])avsvmcloud\.com[^A-Za-z0-9-\.]/Hi"; tag:session,600,seconds;
classtype:trojan-activity; sid:9823578; rev:1; priority:1;
reference:url,https://misp.honeynet.asia/events/view/1358;)
```

MISP – Take Aways

- MISP and Community Sharing is awesome ☺
- Integration Incident Response and Detection
- Allows automation
  - Distribute rules to a distributed sensors via api

Summary

- We’ve gone through some features but there are many other parts not covered
- Context is important to define use case
  - What are we trying to detect or ‘hunt’
  - False positives / False Negatives
  - Infrastructure is unique for everyone – so different hardware capacity and requirements
- Read the official docs!
  - Deprecated keywords
  - New syntax, features, keywords

Suricata Recap

- Check out the webinars
  - https://suricata-ids.org/webinars/
- Forum
  - https://forum.suricata.io/
- Twitter: https://twitter.com/Suricata_IDS
- Read the Docs - suricata.readthedocs.io/
- Women of Suricata Community Initiative
  - https://forum.suricata.io/t/new-women-of-suricata-community-initiative/282

Also Check Out

- https://www.stamus-networks.com/scirius-open-source

Credits

- Adli Wahid created most of the content. If you want to connect with him:
  - LinkedIn – Adli Wahid
  - Instagram/Twitter @adliwahid
  - Email: adli@apnic.net