Hunting Threats That Use Encrypted Network Traffic

May 7, 2020
Introduction

Peter Manev -  
QA/Trainer/Exec team - OISF  
CSO - Stamus Networks  
Twitter @pevma

Eric Leblond -  
Developer/Trainer/Exec team - OISF  
CTO - Stamus Networks  
Twitter @regiteric
A Few Announcements

• Discourse trial
  • [https://forum.suricata.io](https://forum.suricata.io)

• Limited public trainings this year, but will continue to expand our virtual offerings
  • [https://suricata-ids.org/training/](https://suricata-ids.org/training/)

• We’re also booking customized training events for your teams at your location
  • email: info@oisf.net
Announcements Continued...

• OISF - the US non-profit organization with full-time developers dedicated to ensuring Suricata remains world-class

• Join our consortium - we offer several ways to support Suricata's development roadmap and growing community.
  • Details at https://oisf.net/ or contact us at info@oisf.net

• SuriCon 2020 in Boston, MA - annual community conference scheduled for November 16-20, 2020 - https://suricon.net
  • Decision to move SuriCon will be made on or before June 1, 2020
And Finally, Our Next Webinar!

• Joint webinar with Doug Burks from Security Onion!

• Discussing Community ID and how to correlate data between Suricata, Zeek and osquery in Security Onion Hybrid Hunter

• Tentative date of June 10th, 3pm EDT
  • Follow our blogs and social media for official announcement
  • Twitter: @suricata_ids @securityonion
  • https://suricata-ids.org/news/
  • https://blog.securityonion.net/
Suricata

- Suricata is a high-performance network IDS, IPS and network security monitoring (NSM) engine
- Open-source software
- Owned and developed by a community run non-profit foundation - Open Information Security Foundation (OISF)
- Produces a high-level of situational awareness and detailed application layer transaction records
<table>
<thead>
<tr>
<th>Outputs used for encrypted traffic hunting:</th>
<th>OSS tools used in this webinar for visualizing the outputs:</th>
</tr>
</thead>
<tbody>
<tr>
<td>- TLS events (TLS 1,1.2,1.3)</td>
<td>- ELK/SELKS6</td>
</tr>
<tr>
<td>- Anomaly events (new in 6.+ )</td>
<td>- Scirius CE</td>
</tr>
<tr>
<td>- Alerts</td>
<td>- EveBox</td>
</tr>
<tr>
<td>- JA3/JA3s correlation</td>
<td>- Moloch</td>
</tr>
</tbody>
</table>
The TLS Handshake

• Begins with a handshake
  • Asymmetric encryption: two different keys are used
  • AKA public-key cryptography
• Public-key:
  • Server makes this available publicly
• Private-key
  • Secret, only used on the server side
• Data encrypted with public key can only be decrypted by private key
TLS support in Suricata

• TLS handshake analysis
  • Decode TLS message
  • Extract information
  • Output information in dedicated events
    • A JSON event for each connection

• And more features
  • Allow alerting on fields via dedicated keywords
  • Certificate chain extraction

• Additional methods:
  • JA3: algorithm to identify the client by its implementation
  • JA3S: algorithm to identify the server by its implementation
TLS log example

```
Tls: lecvlist.microsoft.com - CN=*.vo.msecnd.net

{"tls": {
  "notbefore": "2020-03-18T19:52:29"
  "issuerdn": "C=US, ST=Washington, L=Redmond, O=Microsoft Corporation, OU=Microsoft IT, CN=Microsoft IT TLS CA 2"
  "ja3": {
    "hash": "1074895878955b2db60423ed2bf8ac23"
  }
  "notafter": "2022-03-18T19:52:29"
  "version": "TLS 1.2"
  "serial": "1C:00:14:5F:23:03:6B:8C:E6:2F:3F:2C:56:00:00:00:14:5F:23"
  "sni": "levclist.microsoft.com"
  "ja3s": {
    "hash": "0cac51a1efed65f5b6c847f539d24313e"
    "string": "771,49192,65281-0-11-23"
  }
  "subject": "CN=*.vo.msecnd.net"
}
```
TLS keywords

• Match on fields in the certificate
  • tls.cert_issuer
  • tls.cert_subject
  • tls.cert_fingerprint
  • tls.sni
  • ...

• Examples
  • Check your usage of internal PKI
    • alert tls any any -> $SERVERS any (tls.cert_issuer;
      content:"Cn=my,OU=awesome,O=company"; sid:1; rev:1;)
  • Pin your main server fingerprint
    • Alert tls any any -> $AUTH_SERVER any (tls.cert_fingerprint;
Real life detection on TLS

- Certificate by default, because bad guy ...likes lazy

```
alert tls $EXTERNAL_NET any -> $HOME_NET any (msg:"ET POLICY OpenSSL Demo CA - Internet Widgits Pty (0)"; flow:established,to_client; tls_cert_subject: "O=Internet Widgits Pty Ltd"; metadata: former_category POLICY; classtype: not-suspicious; sid:2011540; rev:6; metadata: created_at 2019_09_27, updated_at 2017_11_27;)
```

- JA3 phishing

```
alert tls $HOME_NET any -> $EXTERNAL_NET any (msg:"ET JA3 Hash - Possible Malware - Banking Phish": ja3_hash: content:"10ee8d30a5d01c042af7b2b205facc4"; metadata: former_category JA3; reference: url, github.com/trisuls/ja3; reference: url, www.malware-traffic-analysis.net; classtype: unknown; sid:2028362; rev:2; metadata: created_at 2019_09_10, updated_at 2019_10_27;)
```
Some existing sig list

Abuse.ch
  Fingerprint
  ja3
ET Open/Pro using TLS
  TLS rules
  JA3 hash rules
  SSL Blacklist rules
  Tag - deployment SSLDecrypt
TLS JA3 algorithm

- Client sends TLS Client Hello after TCP session established
  - Packet and the way in which it is generated is dependent on packages and methods used when building the client application
- Server responds with TLS Server Hello
  - Similar to client, respond depends on how software was built and data sent from client
- Negotiations are sent in the clear and allow for fingerprinting
  - Still compatible with TLS 1.3
The JA3 Hash

- Decimal values of the byte values of the following fields are concatenated from client hello
  - Version, Accepted Ciphers, List of Extensions, Elliptic Curves, and Elliptic Curve Formats
  - Concatenated in order using a “,” and a “-” to delimit values in fields
  - If no values the fields are left empty
- Result is then hashed using MD5
JA3 example

```json
# Tls: iecvlist.microsoft.com - CN=*,vo.msecnd.net

```
```
"tls": {
  "notbefore": "2020-03-18T19:52:20"
  "issuerdn": "C=US, ST=Washington, L=Redmond, O=Microsoft Corporation, OU=Microsoft IT, CN=Microsoft IT TLS CA 2"
  "ja3": {
    "hash": "187489587855b2db60423ed2bf8ac23"
  }
  "notafter": "2022-03-18T19:52:29"
  "version": "TLS 1.2"
  "sni": "iecvlist.microsoft.com"
  "ja3s": {
    "hash": "8c5c51ae5f5b6c07f539d24313e"
    "string": "771,49192,65281-0-11-23"
  }
  "subject": "CN=*,vo.msecnd.net"
}
```
```
Interesting JA3 resources

• Mapping to TLS user agent
  • Get a name of user agent behind the hash
  • Building a list of hashes to TLS user agent
    • By experiment
  • Example: https://ja3er.com/downloads.html

• Abuse.ch JA3 list
• Use your Suricata to generate (cleanlist/alertlist) hashes
TLS JA3S

• Method of creating a fingerprint from the server side of the TLS handshake – TLS Server Hello
• Decimal values of the bytes for the following fields:
  • Version, Accepted Cipher, and List of Extensions
  • Concatenated and delimited as JA3
• Resulting value is hashed with MD5
• Server doesn’t always respond the same to all clients
  • But responds the same to the same client
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Issuerdn</strong></td>
<td>C=US, O=Let's Encrypt, CN=Let's Encrypt Authority X3</td>
</tr>
<tr>
<td><strong>Ja3.Hash</strong></td>
<td>1d095e68489d3c535297cd8dffb06cb9</td>
</tr>
<tr>
<td><strong>Notafter</strong></td>
<td>2020-04-15T23:05:24</td>
</tr>
<tr>
<td><strong>Notbefore</strong></td>
<td>2020-01-16T23:05:24</td>
</tr>
</tbody>
</table>
One malware sample (demo)
TLS 1.3

- Better security
  - Improved privacy
    - Hide as much data as possible
    - Prevent interception
  - Remove deprecated algorithms (SHA1, RC4, …)
- Faster

(image source: https://kinsta.com/blog/tls-1-3/)
TLS 1.3 event example

```
{
  "in_iface": "eth1",
  "tls": {
    "sni": "app.pendo.io",
    "ja3": {
      "string": "771,4865,51-43",
      "hash": "eb1d94daa7e0344597e756a1fb6e7054"
    },
    "version": "TLS 1.3",
    "ja3": {
      "hash": "66918128f1b9b03303d77c6f2eefd128"
    }
  },
  "see_name": "SSPLAB",
  "@version": "1",
  "offset": 499967217
}
```
What’s remaining for Suricata in TLS 1.3

TLS data
- JA3
- JA3S
- TLS Server Name Indication
  - But there is draft on encryption…

Flow entries
- Data à la Netflow
  - In and Out volume and packets count
  - Enriched with
    - Application layer identification
    - Tunnel information
Flow event in Suricata

- Complete data
  - To server side
  - To client side
- Duration and timestamps
  - Start timestamp
  - End timestamp
  - Duration
- TCP flags
- Flow Identification
  - Flow_id for suricata cross event
  - Community_id for cross system correlation

```json
{
  "timestamp": "2018-07-23T21:21:08.915073+0200",
  "thread_id": 1,
  "flow_id": 1550404452389229,
  "event_type": "flow",
  "src_ip": "172.16.1.117",
  "src_port": 34152,
  "dest_ip": "172.16.1.130",
  "dest_port": 4433,
  "proto": "TCP",
  "app_proto": "tls",
  "flow": {
    "pkts_toserver": 7,
    "pkts_toclient": 6,
    "bytes_toserver": 843,
    "bytes_toclient": 1987,
    "start": "2018-07-23T21:21:08.889197+0200",
    "end": "2018-07-23T21:21:08.915073+0200",
    "age": 0,
    "state": "closed",
    "reason": "shutdown",
    "alerted": false,
    "wrong_thread": true
  },
  "community_id": "1:01+CvJkgApsa3kCF/2FvgQmW9=",
  "tcp": {
    "tcp_flags": "1b",
    "tcp_flags_ts": "1b",
    "tcp_flags_tc": "1b",
    "syn": true,
    "fin": true,
    "psh": true,
    "ack": true,
    "state": "closed"
  }
}
```
Using decryption

Where to put Suricata

• Behind SSL load balancers
• FWs/GWs can decrypt traffic and mirror it to a port
  • McAfee, Cisco, Palo Alto, Juniper...
• Behind/next to proxies
• Most important is to be able to see the traffic as end clients are
Conclusion

• Who said IDS is dead again?
  • TLS is a serious challenge
  • Visibility is decreasing
  • BUT Suricata can still do efficient analysis
• Come out and play
  • Pcap: https://github.com/jstrosch/malware-samples/tree/master/binaries/trickbot/2020/May
  • Suricata forums/help/discussions: https://forum.suricata.io/
  • Suricata trainings/webinars: https://suricata-ids.org/training/
  • OISF: https://oisf.net/