Suricata Rule Development Part 2 (Encrypted Traffic)

In the ever-evolving landscape of network security, we're often faced with a significant challenge: encrypted traffic. Encrypted traffic can pose significant obstacles when it comes to effectively analyzing traffic and developing reliable Intrusion Detection System (IDS) and Intrusion Prevention System (IPS) rules.

There are still several aspects we can leverage to detect potential security threats. Specifically, we can turn our attention to the elements within SSL/TLS certificates and the JA3 fingerprint.

SSL/TLS certificates, exchanged during the initial handshake of an SSL/TLS connection, contain a plethora of details that remain unencrypted. These details can include the issuer, the issue date, the expiry date, and the subject (containing information about who the certificate is for and the domain name). Suspicious or malicious domains might utilize SSL/TLS certificates with anomalous or unique characteristics. Recognizing these anomalies in SSL/TLS certificates can be a stepping stone to crafting effective Suricata rules.

Further, we can also utilize the JA3 hash — a fingerprinting method that provides a unique representation for each SSL/TLS client. The JA3 hash combines details from the client hello packet during the SSL/TLS handshake, creating a digest that could be unique for specific malware families or suspicious software. Again, these hashes can be a powerful tool in formulating detection rules for encrypted traffic.

Let's now navigate to the bottom of this section and click on "Click here to spawn the target system!". Then, let's SSH into the Target IP using the provided credentials. The vast majority of the commands covered from this point up to end of this section can be replicated inside the target, offering a more comprehensive grasp of the topics presented.

Suricata Rule Development Example 5: Detecting Dridex (TLS Encrypted)

Suricata Rule Development Part 2 (Encrypted Traffic)

```
alert tls $EXTERNAL_NET any -> $HOME_NET any (msg:"ET MALWARE ABUSE.CH SSL Blacklist Malicious SSL
```

The rule above triggers an alert upon detecting a TLS session from the external network to the home network, where the payload of the session contains specific byte patterns and meets several conditions. These patterns and conditions correspond to SSL certificates that have been linked to certain variations of the Dridex trojan, as referenced by the SSL blacklist on `abuse.ch`.

No need to understand the rule in its entirety, but let's break down the important parts of it.

- `content:"|16|"; content:"|0b|"; within:8;`: The rule looks for the hex values 16 and 0b within the first 8 bytes of the payload. These represent the handshake message (0x16) and the certificate type (0x0b) in the TLS record.
- `content:"|03 02 01 02 02 09 00|"; fast_pattern;`: The rule looks for this specific pattern of bytes in the packet, which may be characteristic of the certificates used by Dridex.
- `content:"|30 09 06 03 55 04 06 13 02|"; distance:0; pcre:"/^[A-Z]{2}/R";`: This checks for the 'countryName' field in the certificate's subject. The content match here corresponds to an ASN.1 sequence specifying an attribute type and value for 'countryName' (OID 2.5.4.6). The following PCRE checks that the value for 'countryName' begins with two uppercase letters,

? Go to Questions

Table of Contents

Introduction To IDS/IPS

Suricata

Suricata Fundamentals

Suricata Rule Development Part 1

Suricata Rule Development Part 2 (Encrypted Traffic)

Snort

Snort Fundamentals

Snort Rule Development

Zeek

Zeek Fundamentals

Intrusion Detection With Zeek

Skills Assessment

Skills Assessment - Suricata

Skills Assessment - Snort

Skills Assessment - Zeek

My Workstation

OFFLINE

Start Instance

∞ / 1 spawns left

which is a standard format for country codes.

- `content:"|55 04 07|"; distance:0;`: This checks for the 'localityName' field in the certificate's subject (OID 2.5.4.7).
- `content:"|55 04 0a|"; distance:0;`: This checks for the `organizationName` field in the certificate's subject (OID 2.5.4.10).
- `content:"|55 04 03|"; distance:0; byte_test:1,>,13,1,relative;`: This checks for the `commonName` field in the certificate's subject (OID 2.5.4.3). The following byte_test checks that the length of the `commonName` field is more than 13.
- Please also give this very interesting resource on Dridex SSL certificates a look.

The mentioned OIDs (Object Identifiers) are part of the X.509 standard for PKI and are used to uniquely identify the types of fields contained within certificates.

The above rule is already incorporated in the `local.rules` file found in the `/home/htb-student` directory of this section's target. To test it, first, you need to uncomment the rule. Then, execute Suricata on the `dridex.pcap` file, which is located in the `/home/htb-student/pcaps` directory.

Suricata Rule Development Part 2 (Encrypted Traffic)

```
MisaelMacias@htb[/htb]$ sudo suricata -r /home/htb-student/pcaps/dridex.pcap -l . -k none
15/7/2023 -- 20:34:11 - <Notice> - This is Suricata version 6.0.13 RELEASE running in USER mode
15/7/2023 -- 20:34:11 - <Notice> - all 3 packet processing threads, 4 management threads initialize
15/7/2023 -- 20:34:11 - <Notice> - Signal Received.  Stopping engine.
15/7/2023 -- 20:34:11 - <Notice> - Pcap-file module read 1 files, 3683 packets, 3276706 bytes
```

Suricata Rule Development Part 2 (Encrypted Traffic)

```
MisaelMacias@htb[/htb]$ cat fast.log
07/09/2019-18:26:31.480302  [**] [1:2023476:5] ET MALWARE ABUSE.CH SSL Blacklist Malicious SSL cert
07/09/2019-18:26:33.937036  [**] [1:2023476:5] ET MALWARE ABUSE.CH SSL Blacklist Malicious SSL cert
07/09/2019-18:26:39.373287  [**] [1:2023476:5] ET MALWARE ABUSE.CH SSL Blacklist Malicious SSL cert
07/09/2019-18:26:29.628847  [**] [1:2023476:5] ET MALWARE ABUSE.CH SSL Blacklist Malicious SSL cert
07/09/2019-18:30:08.787378  [**] [1:2023476:5] ET MALWARE ABUSE.CH SSL Blacklist Malicious SSL cert
---SNIP---
```

Suricata Rule Development Example 6: Detecting Sliver (TLS Encrypted)

Suricata Rule Development Part 2 (Encrypted Traffic)

```
alert tls any any -> any any (msg:"Sliver C2 SSL"; ja3.hash; content:"473cd7cb9faa642487833865d516e
```

The Suricata rule above is designed to detect certain variations of Sliver whenever it identifies a TLS connection with a specific JA3 hash.

A PCAP file named `sliverenc.pcap` containing encrypted Sliver traffic is located in the `/home/htb-student/pcaps` directory of this section's target.

The JA3 hash can be calculated as follows.

Suricata Rule Development Part 2 (Encrypted Traffic)

```
MisaelMacias@htb[/htb]$ ja3 -a --json /home/htb-student/pcaps/sliverenc.pcap
[
    {
        "destination_ip": "23.152.0.91",
        "destination_port": 443,
        "ja3": "771,49195-49199-49196-49200-52393-52392-49161-49171-49162-49172-156-157-47-53-49170
        "ja3_digest": "473cd7cb9faa642487833865d516e578",
        "source_ip": "10.10.20.101",
        "source_port": 53222,
        "timestamp": 1634749464.600896
    },
    {
```

```
        "destination_ip": "23.152.0.91",
        "destination_port": 443,
        "ja3": "771,49195-49199-49196-49200-52393-52392-49161-49171-49162-49172-156-157-47-53-49170
        "ja3_digest": "473cd7cb9faa642487833865d516e578",
        "source_ip": "10.10.20.101",
        "source_port": 53225,
        "timestamp": 1634749465.069819
    },
    {
        "destination_ip": "23.152.0.91",
        "destination_port": 443,
        "ja3": "771,49195-49199-49196-49200-52393-52392-49161-49171-49162-49172-156-157-47-53-49170
        "ja3_digest": "473cd7cb9faa642487833865d516e578",
        "source_ip": "10.10.20.101",
        "source_port": 53229,
        "timestamp": 1634749585.240773
    },
---SNIP---
```


The above rule is already incorporated in the `local.rules` file found in the `/home/htb-student` directory of this section's target. To test it, first, you need to uncomment the rule. Then, execute Suricata on the `sliverenc.pcap` file, which is located in the `/home/htb-student/pcaps` directory.

Suricata Rule Development Part 2 (Encrypted Traffic)

```
MisaelMacias@htb[/htb]$ sudo suricata -r /home/htb-student/pcaps/sliverenc.pcap -l . -k none
15/7/2023 -- 22:30:37 - <Notice> - This is Suricata version 6.0.13 RELEASE running in USER mode
15/7/2023 -- 22:30:37 - <Notice> - all 3 packet processing threads, 4 management threads initialize
15/7/2023 -- 22:30:37 - <Notice> - Signal Received.  Stopping engine.
15/7/2023 -- 22:30:37 - <Notice> - Pcap-file module read 1 files, 15547 packets, 11904606 bytes
```

Suricata Rule Development Part 2 (Encrypted Traffic)

```
MisaelMacias@htb[/htb]$ cat fast.log
10/20/2021-17:04:25.166658  [**] [1:1002:1] Sliver C2 SSL [**] [Classification: (null)] [Priority:
10/20/2021-17:07:25.315183  [**] [1:1002:1] Sliver C2 SSL [**] [Classification: (null)] [Priority:
10/20/2021-17:04:24.700690  [**] [1:1002:1] Sliver C2 SSL [**] [Classification: (null)] [Priority:
10/20/2021-17:06:25.328173  [**] [1:1002:1] Sliver C2 SSL [**] [Classification: (null)] [Priority:
10/20/2021-17:10:25.311929  [**] [1:1002:1] Sliver C2 SSL [**] [Classification: (null)] [Priority:
10/20/2021-17:08:25.312485  [**] [1:1002:1] Sliver C2 SSL [**] [Classification: (null)] [Priority:
10/20/2021-17:09:25.210256  [**] [1:1002:1] Sliver C2 SSL [**] [Classification: (null)] [Priority:
10/20/2021-17:14:25.235711  [**] [1:1002:1] Sliver C2 SSL [**] [Classification: (null)] [Priority:
10/20/2021-17:11:25.213759  [**] [1:1002:1] Sliver C2 SSL [**] [Classification: (null)] [Priority:
10/20/2021-17:12:25.302237  [**] [1:1002:1] Sliver C2 SSL [**] [Classification: (null)] [Priority:
10/20/2021-17:13:25.236776  [**] [1:1002:1] Sliver C2 SSL [**] [Classification: (null)] [Priority:
---SNIP---
```

Start Instance

∞ / 1 spawns left

Waiting to start...

Enable step-by-step solutions for all questions

Questions

Answer the question(s) below to complete this Section and earn cubes!

Download VPN Connection File

Target(s): Click here to spawn the target system!

SSH to with user "htb-student" and password "HTB_@cademy_stdnt!"

+ 1 There is a file named trickbot.pcap in the /home/htb-student/pcaps directory, which contains network traffic related to a certain variation of the Trickbot malware. Enter the precise string that should be specified in the content keyword of the rule with sid 100299 within the local.rules file so that an alert is triggered as your answer.

72a589da586844d7f0818ce684948eea

Submit

← Previous Next →

Mark Complete & Next