

# BSidesJeddah-Part1

## 1. What is the victim's MAC address?

Wraz z plikiem pcap otrzymaliśmy plik suricata.rules który jest sygnaturami systemu IDS/IPS możemy go użyć do przeskanowania pliku pcap suricata bądź snortem w moim przypadku suricata.

Z wygenerowanych logów możemy dowiedzieć się że adresacja ip w sieci LAN opiera się na klasie adresu C 192... dzięki tej informacji możemy wyfiltrować same adresy zaczynające się na 192.

```
michał@linux:~/Desktop/log$ cat fast.log | cut -d " " -f 21 | sort | uniq -c | sort | grep 192
11 192.168.112.128:443
1 192.168.112.139:110
1 192.168.112.139:59491
1 192.168.112.2:137
27 192.168.112.128:80
3 192.168.112.139:0
michał@linux:~/Desktop/log$
```

Jak można zauważyć najczęściej podejrzanego ruchu zostało zalogowane między adresami 192.168.112.139 oraz 192.168.112.128 ale kiedy przyjrzymy się ruchowi w wiresharku można łatwo wywnioskować że ofiarą jest 192.168.112.139. Pozostaje odczytać jego adres fizyczny:

The image shows a Wireshark packet capture. The packet list on the left shows several packets between 192.168.112.139 and 192.168.112.128. Packet 2 is selected, showing details for Ethernet II, Internet Protocol Version 4, and Transmission Control Protocol. The Ethernet II section shows Source: VMWare\_b7:ca:91 (00:0c:29:b7:ca:91) and Destination: VMWare\_f9:ee:df (00:50:56:f9:ee:df). The packet bytes pane at the bottom shows the hexadecimal representation of the frame, with the source MAC address 00:0c:29:b7:ca:91 highlighted in blue.

Answer: 00:0C:29:B7:CA:91

## 2. What is the address of the company associated with the victim's machine MAC address?

Dla przykładu ja użyłem tego narzędzia do znalezienia adresu:

<https://www.ipchecktool.com/tool/macfinder>

Answer: 3401 Hillview Avenue Palo Alto CA 94304 US

### 3. What is the attacker's IP address?

Użyjemy tego polecenia aby wyfiltrować z logów suricata wszystkie adresy oraz informacje ile razy się powtarzają i zapiszemy je do pliku out1.txt może się nam to przydać w późniejszym czasie.

```
cat fast.log | cut -d " " -f 21 | sort | uniq -c | sort > out1.txt
```

```
nichal@linux:~/Desktop/log$ cat out1.txt | grep -i 192.  
11 192.168.112.128:443  
1 192.168.112.139:110  
1 192.168.112.139:59491  
1 192.168.112.2:137  
27 192.168.112.128:80  
3 192.168.112.139:0  
nichal@linux:~/Desktop/log$
```

Jeden z adresów wysyłał podejrzanie dużo ruchu do ofiary oraz wielokrotnie był uważany przez sygnatury jako potencjalna exploitacja.

Answer:192.168.112.128

### 4. What is the IPv4 address of the DNS server used by the victim machine?

Wyfiltrujmy ruch dns związany z adresem ip ofiary:

```
nichal@linux:~/Desktop$ tshark -r e3.pcap -Y 'ip.addr == 192.168.112.139 and dns'  
2687 35.783752 192.168.112.139 → 192.168.112.2 DNS 88 Standard query 0x54ed PTR 128.112.168.192.in-addr.arpa  
2688 35.812488 192.168.112.139 → 192.168.112.2 DNS 88 Standard query 0x54ed PTR 128.112.168.192.in-addr.arpa  
2690 35.879660 192.168.112.2 → 192.168.112.139 DNS 88 Standard query response 0x54ed No such name PTR 128.112.168.192.in-addr.arpa  
2691 35.879660 192.168.112.2 → 192.168.112.139 DNS 88 Standard query response 0x54ed No such name PTR 128.112.168.192.in-addr.arpa  
2727 38.109430 192.168.112.139 → 192.168.112.2 DNS 93 Standard query 0x93a5 A v10.vortex-win.data.microsoft.com  
2728 38.140609 192.168.112.139 → 192.168.112.2 DNS 93 Standard query 0x93a5 A v10.vortex-win.data.microsoft.com  
2729 38.148931 192.168.112.2 → 192.168.112.139 DNS 161 Standard query response 0x93a5 A v10.vortex-win.data.microsoft.com CNAME v10-win.vortex.data.trafficmanager.net A 40.77.226.250  
2730 38.148931 192.168.112.2 → 192.168.112.139 DNS 161 Standard query response 0x93a5 A v10.vortex-win.data.microsoft.com CNAME v10-win.vortex.data.trafficmanager.net A 40.77.226.250  
5616 159.769708 192.168.112.139 → 192.168.112.2 DNS 86 Standard query 0xf891 A templateservice.office.com  
5617 159.797022 192.168.112.139 → 192.168.112.2 DNS 86 Standard query 0xf891 A templateservice.office.com  
5618 159.859611 192.168.112.2 → 192.168.112.139 DNS 188 Standard query response 0xf891 A templateservice.office.com CNAME templateservice.office.com.edgekey.net CNAME e16253.d.akamaet  
6  
5619 159.859839 192.168.112.2 → 192.168.112.139 DNS 188 Standard query response 0xf891 A templateservice.office.com CNAME templateservice.office.com.edgekey.net CNAME e16253.d.akamaet  
6  
5648 160.333104 192.168.112.139 → 192.168.112.2 DNS 92 Standard query 0xfdd5 A onextemplates.content.office.net  
5649 160.359491 192.168.112.139 → 192.168.112.2 DNS 92 Standard query 0xfdd5 A onextemplates.content.office.net  
5650 160.371776 192.168.112.2 → 192.168.112.139 DNS 195 Standard query response 0xfdd5 A onextemplates.content.office.net CNAME onextemplates.content.office.net.edgekey.net CNAME e584.  
23.205.48.23  
5651 160.371939 192.168.112.2 → 192.168.112.139 DNS 195 Standard query response 0xfdd5 A onextemplates.content.office.net CNAME onextemplates.content.office.net.edgekey.net CNAME e584.  
23.205.48.23  
35922 247.813544 192.168.112.139 → 192.168.112.2 DNS 91 Standard query 0x26df A settings-win.data.microsoft.com  
35923 247.844153 192.168.112.139 → 192.168.112.2 DNS 91 Standard query 0x26df A settings-win.data.microsoft.com  
35924 247.854118 192.168.112.2 → 192.168.112.139 DNS 154 Standard query response 0x26df A settings-win.data.microsoft.com CNAME settingsfd-geo.trafficmanager.net A 51.104.136.2  
35925 247.854118 192.168.112.2 → 192.168.112.139 DNS 154 Standard query response 0x26df A settings-win.data.microsoft.com CNAME settingsfd-geo.trafficmanager.net A 51.104.136.2  
nichal@linux:~/Desktop$
```

Jak można zaobserwować adres ofiary odpytywał tylko jeden serwer DNS

Answer:192.168.112.2

## 5. What domain is the victim looking up in packet 5648?

```
micha@linux:~/Desktop$ tshark -r e3.pcap -Y 'ip.addr == 192.168.112.139 and dns'
2687 35.783752 192.168.112.139 → 192.168.112.2 DNS 88 Standard query 0x54ed PTR 128.112.168.192.in-addr.arpa
2688 35.812488 192.168.112.139 → 192.168.112.2 DNS 88 Standard query 0x54ed PTR 128.112.168.192.in-addr.arpa
2690 35.879660 192.168.112.2 → 192.168.112.139 DNS 88 Standard query response 0x54ed No such name PTR 128.112.168.192.in-addr.arpa
2691 35.879660 192.168.112.2 → 192.168.112.139 DNS 88 Standard query response 0x54ed No such name PTR 128.112.168.192.in-addr.arpa
2727 38.109430 192.168.112.139 → 192.168.112.2 DNS 93 Standard query 0x93a5 A v10.vortex-win.data.microsoft.com
2728 38.140609 192.168.112.139 → 192.168.112.2 DNS 93 Standard query 0x93a5 A v10.vortex-win.data.microsoft.com
2729 38.148931 192.168.112.2 → 192.168.112.139 DNS 161 Standard query response 0x93a5 A v10.vortex-win.data.microsoft.com
2730 38.148931 192.168.112.2 → 192.168.112.139 DNS 161 Standard query response 0x93a5 A v10.vortex-win.data.microsoft.com
5616 159.769708 192.168.112.139 → 192.168.112.2 DNS 86 Standard query 0xf891 A templateservice.office.com
5617 159.797022 192.168.112.139 → 192.168.112.2 DNS 86 Standard query 0xf891 A templateservice.office.com
5618 159.859611 192.168.112.2 → 192.168.112.139 DNS 188 Standard query response 0xf891 A templateservice.office.com CNAME to
5619 159.859611 192.168.112.2 → 192.168.112.139 DNS 188 Standard query response 0xf891 A templateservice.office.com CNAME to
5648 160.333104 192.168.112.139 → 192.168.112.2 DNS 92 Standard query 0xfdd5 A omextemplates.content.office.net
5649 160.359491 192.168.112.139 → 192.168.112.2 DNS 92 Standard query 0xfdd5 A omextemplates.content.office.net
5650 160.371776 192.168.112.2 → 192.168.112.139 DNS 195 Standard query response 0xfdd5 A omextemplates.content.office.net CNAME to
5651 160.371939 192.168.112.2 → 192.168.112.139 DNS 195 Standard query response 0xfdd5 A omextemplates.content.office.net CNAME to
35922 247.813544 192.168.112.139 → 192.168.112.2 DNS 91 Standard query 0x26df A settings-win.data.microsoft.com
35923 247.844153 192.168.112.139 → 192.168.112.2 DNS 91 Standard query 0x26df A settings-win.data.microsoft.com
35924 247.854118 192.168.112.2 → 192.168.112.139 DNS 154 Standard query response 0x26df A settings-win.data.microsoft.com CNAME to
35925 247.854118 192.168.112.2 → 192.168.112.139 DNS 154 Standard query response 0x26df A settings-win.data.microsoft.com CNAME to
micha@linux:~/Desktop$
```

Answer: omextemplates.content.office.net

## 6. What is the server certificate public key that was used in TLS session:

731300002437c17bdfa2593dd0e0b28d391e680f764b5db3c4059f7abadbb28e

The image shows a Wireshark packet capture of a TLS handshake. The left pane displays the packet list, with packet 2739 (969 bytes on wire) selected. The middle pane shows the packet details for this packet, highlighting the 'Handshake Protocol: Server Key Exchange' section. The right pane shows the raw packet data in hexadecimal and ASCII.

Packet 2739: 969 bytes on wire (7752 bits), 969 bytes captured (7752 bits) on interface 0

Ethernet II, Src: VMware\_f9:ee:df (00:50:56:f9:ee:df), Dst: Destination: VMWare\_b7:ca:91 (00:0c:29:b7:ca:91)

Source: VMWare\_f9:ee:df (00:50:56:f9:ee:df)

Type: IPv4 (0x0800)

Internet Protocol Version 4, Src: 40.77.226.250, Dst: 192.168.112.2

Transmission Control Protocol, Src Port: 443, Dst Port: 80

[3 Reassembled TCP Segments (3715 bytes): #2736(1400), #2737(1400), #2738(1400)]

Transport Layer Security

Session ID Length: 32

Session ID: 731300002437c17bdfa2593dd0e0b28d391e680f764b5db3c4059f7abadbb28e

Cipher Suite: TLS\_ECDHE\_RSA\_WITH\_AES\_128\_GCM\_SHA256 (0xc02f)

Compression Method: null (0)

Extensions Length: 13

Extension: extended\_master\_secret (len=0)

Extension: renegotiation\_info (len=1)

Extension: server\_name (len=0)

Handshake Protocol: Certificate

Handshake Protocol: Server Key Exchange

Handshake Type: Server Key Exchange (12)

Length: 296

EC Diffie-Hellman Server Params

Curve Type: named\_curve (0x03)

Named Curve: x25519 (0x001d)

Pubkey Length: 32

Pubkey: 64089e29f386356f1ffbd64d7056ca0f1d489a09cd7ebda630f2b7394e319406

Answer: 64089e29f386356f1ffbd64d7056ca0f1d489a09cd7ebda630f2b7394e319406

## 7. What domain is the victim connected to in packet 4085?

```
2739 38.329874 40.77.226.250 → 192.168.112.139 TLSv1.2 969 Server Hello, Certificate, Server Key Exchange, Server Hello Done
2740 38.334348 192.168.112.139 → 40.77.226.250 TLSv1.2 147 Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
2741 38.334732 40.77.226.250 → 192.168.112.139 TCP 60 443 → 50073 [ACK] Seq=3716 Ack=306 Win=64240 Len=0
2742 38.420633 40.77.226.250 → 192.168.112.139 TLSv1.2 105 Change Cipher Spec, Encrypted Handshake Message
2743 38.424137 192.168.112.139 → 40.77.226.250 TLSv1.2 824 Application Data
2744 38.424512 40.77.226.250 → 192.168.112.139 TCP 60 443 → 50073 [ACK] Seq=3767 Ack=1076 Win=64240 Len=0
2745 38.424748 192.168.112.139 → 40.77.226.250 TLSv1.2 656 Application Data
2746 38.425037 40.77.226.250 → 192.168.112.139 TCP 60 443 → 50073 [ACK] Seq=3767 Ack=1678 Win=64240 Len=0
2747 38.693254 40.77.226.250 → 192.168.112.139 TLSv1.2 349 Application Data
2748 38.701642 192.168.112.139 → 40.77.226.250 TLSv1.2 824 Application Data
2749 38.701912 40.77.226.250 → 192.168.112.139 TCP 60 443 → 50073 [ACK] Seq=4062 Ack=2448 Win=64240 Len=0
2750 38.702287 192.168.112.139 → 40.77.226.250 TLSv1.2 466 Application Data
2751 38.702492 40.77.226.250 → 192.168.112.139 TCP 60 443 → 50073 [ACK] Seq=4062 Ack=2860 Win=64240 Len=0
2760 39.017047 40.77.226.250 → 192.168.112.139 TLSv1.2 349 Application Data
2761 39.023882 192.168.112.139 → 40.77.226.250 TLSv1.2 825 Application Data
2762 39.024232 40.77.226.250 → 192.168.112.139 TCP 60 443 → 50073 [ACK] Seq=4357 Ack=3631 Win=64240 Len=0
2763 39.024453 192.168.112.139 → 40.77.226.250 TLSv1.2 3411 Application Data
2764 39.024767 40.77.226.250 → 192.168.112.139 TCP 60 443 → 50073 [ACK] Seq=4357 Ack=5091 Win=64240 Len=0
2765 39.024767 40.77.226.250 → 192.168.112.139 TCP 60 443 → 50073 [ACK] Seq=4357 Ack=6551 Win=64240 Len=0
2766 39.024767 40.77.226.250 → 192.168.112.139 TCP 60 443 → 50073 [ACK] Seq=4357 Ack=6988 Win=64240 Len=0
2767 39.342050 40.77.226.250 → 192.168.112.139 TLSv1.2 350 Application Data
2768 39.359062 192.168.112.139 → 40.77.226.250 TCP 54 50073 → 443 [ACK] Seq=6988 Ack=4653 Win=63944 Len=0
4085 39.328418 192.168.112.139 → 40.77.226.250 TCP 54 50073 → 443 [FIN, ACK] Seq=6988 Ack=4653 Win=63944 Len=0
4086 39.328826 40.77.226.250 → 192.168.112.139 TCP 60 443 → 50073 [ACK] Seq=4653 Ack=6989 Win=64239 Len=0
4089 39.412866 40.77.226.250 → 192.168.112.139 TCP 60 443 → 50073 [FIN, PSH, ACK] Seq=4653 Ack=6989 Win=64239 Len=0
4090 39.412959 192.168.112.139 → 40.77.226.250 TCP 54 50073 → 443 [ACK] Seq=6989 Ack=4654 Win=63944 Len=0
```

W pakiecie 4085 nasza ofiara komunikuje się z adresem 40.77.226.250 czyli z adresem logicznym ip nie mamy podanej tutaj domeny jednak kiedy wrócimy do screena z pytania 4 i 5 zauważmy że ofiara odpytywała już serwer DNS w poszukiwaniu adresu 40.77.226.250:

```
4090 39.412959 192.168.112.139 → 40.77.226.250 TCP 54 50073 → 443 [ACK] Seq=6989 Ack=4654 Win=63944 Len=0
michal@linux:~/Desktop$ tshark -r e3.pcap -Y "ip.addr == 192.168.112.139 and dns"
2687 35.783752 192.168.112.139 → 192.168.112.2 DNS 88 Standard query 0x54ed PTR 128.112.168.192.in-addr.arpa
2688 35.812488 192.168.112.139 → 192.168.112.2 DNS 88 Standard query 0x54ed PTR 128.112.168.192.in-addr.arpa
2690 35.879660 192.168.112.2 → 192.168.112.139 DNS 88 Standard query response 0x54ed No such name PTR 128.112.168.192.in-addr.arpa
2691 35.879660 192.168.112.2 → 192.168.112.139 DNS 88 Standard query response 0x54ed No such name PTR 128.112.168.192.in-addr.arpa
2727 38.094330 192.168.112.139 → 192.168.112.2 DNS 93 Standard query 0x93a5 A v10.vortex-win.data.microsoft.com
2728 38.140609 192.168.112.139 → 192.168.112.2 DNS 93 Standard query 0x93a5 A v10.vortex-win.data.microsoft.com
2729 38.148931 192.168.112.2 → 192.168.112.139 DNS 161 Standard query response 0x93a5 A v10.vortex-win.data.microsoft.com CNAME v10-win.vortex.data.trafficmanager.net A 40.77.226.250
2730 38.148931 192.168.112.2 → 192.168.112.139 DNS 161 Standard query response 0x93a5 A v10.vortex-win.data.microsoft.com CNAME v10-win.vortex.data.trafficmanager.net A 40.77.226.250
5616 159.797082 192.168.112.139 → 192.168.112.2 DNS 86 Standard query 0xf891 A templateservice.office.com
5617 159.797082 192.168.112.139 → 192.168.112.2 DNS 86 Standard query 0xf891 A templateservice.office.com
5618 159.859611 192.168.112.2 → 192.168.112.139 DNS 188 Standard query response 0xf891 A templateservice.office.com CNAME templateservice.office.com.edgekey.net CNAME e16253.d.akamaiedge.net A 2.23.28.8
5619 159.859639 192.168.112.2 → 192.168.112.139 DNS 188 Standard query response 0xf891 A templateservice.office.com CNAME templateservice.office.com.edgekey.net CNAME e16253.d.akamaiedge.net A 2.23.28.8
5648 160.333104 192.168.112.139 → 192.168.112.2 DNS 92 Standard query 0xfdd5 A onextemplates.content.office.net
5649 160.359491 192.168.112.139 → 192.168.112.2 DNS 92 Standard query 0xfdd5 A onextemplates.content.office.net
5650 160.371776 192.168.112.2 → 192.168.112.139 DNS 195 Standard query response 0xfdd5 A onextemplates.content.office.net CNAME onextemplates.content.office.net.edgekey.net CNAME e584.g.akamaiedge.net A 23.205.48.23
5651 160.371939 192.168.112.2 → 192.168.112.139 DNS 195 Standard query response 0xfdd5 A onextemplates.content.office.net CNAME onextemplates.content.office.net.edgekey.net CNAME e584.g.akamaiedge.net A 23.205.48.23
35922 247.813544 192.168.112.139 → 192.168.112.2 DNS 91 Standard query 0x26df A settings-win.data.microsoft.com
35923 247.844153 192.168.112.139 → 192.168.112.2 DNS 91 Standard query 0x26df A settings-win.data.microsoft.com
35924 247.854118 192.168.112.2 → 192.168.112.139 DNS 154 Standard query response 0x26df A settings-win.data.microsoft.com CNAME settingsfd-geo.trafficmanager.net A 51.104.136.2
35925 247.854118 192.168.112.2 → 192.168.112.139 DNS 154 Standard query response 0x26df A settings-win.data.microsoft.com CNAME settingsfd-geo.trafficmanager.net A 51.104.136.2
```

Answer: v10.vortex-win.data.microsoft.com

## 8. The attacker conducted a port scan on the victim machine. How many open ports did the attacker find?

Atakujący skanował serwer / adres ofiary w poszukiwaniu otwartych portów na których działają usługi. W skrócie polega to na tym że narzędzie skanujące wysyła do określonego zakresu portów flagę SYN tcp oraz oczekuje od danej usługi odpowiedzi w postaci flagi SYN, ACK

tcp.ack and ip.addr == 192.168.112.128 and ip.src_host == 192.168.112.139						
No.	Time	Source	Destination	Protocol	Length	Info
2279	7.392589	192.168.112.139	192.168.112.128	TCP	54	58000 → 46506 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
2281	7.392883	192.168.112.139	192.168.112.128	TCP	54	3800 → 43626 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
2283	7.393197	192.168.112.139	192.168.112.128	TCP	54	50006 → 36364 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
2286	7.393391	192.168.112.139	192.168.112.128	TCP	54	1185 → 37168 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
2287	7.393376	192.168.112.139	192.168.112.128	TCP	54	2009 → 54562 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
2289	7.393597	192.168.112.139	192.168.112.128	TCP	54	5960 → 50472 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
2291	7.393612	192.168.112.139	192.168.112.128	TCP	54	32773 → 46704 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
2293	7.393677	192.168.112.139	192.168.112.128	TCP	54	19842 → 41304 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
2296	7.398684	192.168.112.139	192.168.112.128	TCP	74	25 → 59192 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1460 WS=
2297	7.398752	192.168.112.139	192.168.112.128	TCP	74	119 → 56902 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1460 WS=
2301	7.398880	192.168.112.139	192.168.112.128	TCP	74	135 → 38694 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1460 WS=
2302	7.398965	192.168.112.139	192.168.112.128	TCP	74	139 → 45162 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1460 WS=
2307	7.399165	192.168.112.139	192.168.112.128	TCP	74	143 → 38264 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1460 WS=
2311	7.399406	192.168.112.139	192.168.112.128	TCP	74	445 → 45624 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1460 WS=
2312	7.399565	192.168.112.139	192.168.112.128	TCP	74	587 → 37866 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1460 WS=
2315	7.402400	192.168.112.139	192.168.112.128	SMTP	93	S: 220 WIN-D2TSDME6NN ESMTF
Frame 2296: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface 0 Ethernet II, Src: VMware_b7:ca:91 (00:0c:29:b7:ca:91), Dst: VMware_61:f9:84 (00:0c:29:61:f9:84) Destination: VMware_61:f9:84 (00:0c:29:61:f9:84) Address: VMware_61:f9:84 (00:0c:29:61:f9:84) ....0. .... = LG bit: Globally unique address (factory default) ....0. .... = IG bit: Individual address (unicast) Source: VMware_b7:ca:91 (00:0c:29:b7:ca:91) Address: VMware_b7:ca:91 (00:0c:29:b7:ca:91) ....0. .... = LG bit: Globally unique address (factory default)						
0000	00 0c 29 61 f9 84 00 0c 29 b7 ca 91 00 00 45 00	[...].E				
0010	00 3c 37 ba 40 00 00 06 00 00 c0 a8 70 8b c0 a8	<7.0...p				
0020	70 00 00 19 e7 38 8e b7 2c 3c 7c 78 1d b7 a0 12	p...8...< x...				
0030	20 00 62 8b 00 00 02 04 05 b4 01 03 03 08 04 02	.b... ..				
0040	08 0a 00 61 69 d3 29 09 33 68	...ai). 3h				

Answer:7

9. Analyze the pcap using the provided rules. What is the CVE number falsely alerted by Suricata?

Wracamy do logów wygenerowanych przez suricate tzn fast.log oraz filtrujemy słowa kluczowego „CVE”:

```

michal@linux:~/Desktop/log$ cat fast.log | grep -i CVE
10/01/2021-14:31:54.593627  [**] [1:2030387:1] ET EXPLOIT Possible CVE-2020-11899 Multicast out-of-bound read [**] [Classification: At
000:0000:d4aa:8d54:3230:720b:52484 -> ff02:0000:0000:0000:0000:0000:0001:0003:5355
michal@linux:~/Desktop/log$

```

Mamy jeden wynik i zawiera on poprawną odpowiedź.

Answer: CVE-2020-11899



10. What is the command parameter sent by the attacker in packet number 2650?

The image shows a Wireshark packet capture. The filter is set to 'ip.addr == 192.168.112.128'. Packet 2650 is selected, showing details for Ethernet II, Internet Protocol Version 4, Transmission Control Protocol, and Simple Mail Transfer Protocol. The SMTP command is 'EHELO kali\r\n' and the request parameter is 'kali'.

Answer: kali

11. What is the stream number which contains email traffic?

Answer:1183

12. What is the victim's email address?

Filtrujemy tcp.stream == 1183 i szukamy adresu docelowego SMTP RCPT TO:

No.	Time	Source	Destination	Protocol	Length	Info
2645	35.699982	192.168.112.128	192.168.112.139	TCP	74	59216 → 25 [SYN] Seq=0 Win=64240 Len=0 M
2646	35.700154	192.168.112.139	192.168.112.128	TCP	74	25 → 59216 [SYN, ACK] Seq=0 Ack=1 Win=81
2647	35.700835	192.168.112.128	192.168.112.139	TCP	66	59216 → 25 [ACK] Seq=1 Ack=1 Win=64256 L
2648	35.704634	192.168.112.139	192.168.112.128	SMTP	93	S: 220 WIN-D2TSEME6NN ESMTP
2649	35.705314	192.168.112.128	192.168.112.139	TCP	66	59216 → 25 [ACK] Seq=1 Ack=28 Win=64256
2650	35.705539	192.168.112.128	192.168.112.139	SMTP	77	C: EHELO kali
2651	35.706073	192.168.112.139	192.168.112.128	SMTP	132	S: 250-WIN-D2TSEME6NN   SIZE 20480000
2652	35.706672	192.168.112.128	192.168.112.139	TCP	66	59216 → 25 [ACK] Seq=12 Ack=94 Win=64256
2653	35.706896	192.168.112.128	192.168.112.139	SMTP	106	C: MAIL FROM:<support@cyberdefenders.org
2654	35.715348	192.168.112.139	192.168.112.128	SMTP	74	S: 250 OK
2655	35.715822	192.168.112.128	192.168.112.139	TCP	66	59216 → 25 [ACK] Seq=52 Ack=102 Win=6425
2656	35.716019	192.168.112.128	192.168.112.139	SMTP	103	C: RCPT TO:<joshua@cyberdefenders.org>
2657	35.719764	192.168.112.139	192.168.112.128	SMTP	74	S: 250 OK

Answer: [joshua@cyberdefenders.org](mailto:joshua@cyberdefenders.org)

13. What was the time attacker sent the email?

Filtrujemy strumień po raz kolejny i odczytujemy godzinę z e-maila

Answer: 12:31:54

#### 14. What is the version of the program used to send the email?

W pakiecie 2660 znajduje się zawartość wiadomości z informacji warstwy 7 można odczytać narzędzie za pomocą którego prawdopodobnie wysłano wiadomość

Urgent pointer: 0  
Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps  
[SEQ/ACK analysis]  
[Timestamps]  
TCP payload (380 bytes)  
Simple Mail Transfer Protocol  
Line-based text data (9 lines)  
Message-ID: <595903.006239922-sendEmail@kali>\r\n  
From: "support@cyberdefenders.org" <support@cyberdefenders.org>\r\n  
To: "joshua@cyberdefenders.org" <joshua@cyberdefenders.org>\r\n  
Subject: Immediate responses\r\n  
Date: Fri, 1 Oct 2021 12:31:54 +0000\r\n  
X-Mailer: sendEmail-1.56\r\n  
Content-Type: multipart/mixed; boundary="----MIME delimiter for sendEmail-803805.438959077"\r\n  
Reassembled DATA in frame: 2666

0070 0a 46 72 6f 6d 3a 20 22 73 75 70 70 6f 72 74 40 .From: " support@  
0080 63 79 62 65 72 64 05 66 65 66 64 05 72 73 2e 6f cyberdef enders.o  
0090 72 67 22 20 3c 73 75 70 70 6f 72 74 40 63 79 62 rg" <sup port@c  
00a0 65 72 64 65 66 65 66 64 65 72 73 2e 6f 72 67 3e erdefend ers.org  
00b0 0d 0a 54 6f 3a 20 22 6a 6f 73 68 75 61 40 63 79 .To: "j oshua@c  
00c0 62 65 72 64 65 66 65 66 64 65 72 73 2e 6f 72 67 berdefend ers.org  
00d0 22 20 3c 6a 6f 73 68 75 61 40 63 79 62 65 72 64 " <joshu a@c  
00e0 65 72 64 65 66 65 66 64 65 72 73 2e 6f 72 67 3e eferdef ers.org  
00f0 75 62 6a 65 63 74 3a 20 49 6d 6d 65 64 69 61 74 bject: Immediat  
0100 65 20 72 65 73 70 6f 6e 65 73 6d 6a 44 61 74 65 e respon es..Date  
0110 3a 20 46 72 69 2c 20 31 20 4f 63 74 20 32 30 32 : Fri, 1 Oct 202  
0120 31 20 31 32 3a 33 31 3a 35 34 20 2b 30 30 30 30 1 12:31: 54 +0000  
0130 0d 0a 58 2d 4d 01 69 6c 65 72 3a 20 73 65 66 64 .X-Mail er: Send  
0140 45 6d 61 69 6c 2d 31 2e 35 36 6d 6a 4d 49 40 45 Email-1.56..MIME  
0150 20 50 65 72 73 69 6f 6e 3a 20 31 2e 30 6d 0a 43 -Version: 1.0..C  
0160 6f 6e 74 65 6e 74 2d 54 79 70 65 3a 20 6d 75 6f ontent-T ype: mul  
0170 74 69 70 61 72 74 2f 6d 69 70 65 64 3b 20 62 6f tupart/m ixed; bo  
0180 75 6e 64 61 72 79 3d 22 2d 2d 2d 4d 49 40 45 undary=" ----MIME  
0190 20 64 65 6c 69 6d 69 74 65 72 20 66 6f 72 20 73 delimit er for s  
01a0 65 6e 64 45 6d 61 69 6c 2d 30 30 33 38 30 35 2e endEmail -803805.  
01b0 34 33 30 39 35 39 30 37 37 22 0d 0a 0d 0a 43895907 7".....

Answer: 1.56

#### 15. What is the MD5 hash of the email attachment?

Robimy rzut pakietu odpowiedzi ofiary na pakiet SMTP i zapisujemy go z rozszerzeniem .eml

220 WIN-D2TSDME6NN ESMT  
EHLO kali  
250-WIN-D2TSDME6NN  
250-SIZE 20480000  
250-AUTH LOGIN  
250 HELP  
MAIL FROM:<support@cyberdefenders.org>  
103 C:  
RCPT TO:<joshua@cyberdefenders.org>  
72 C:  
250 OK  
81 S:  
DATA  
354 OK, send.  
Message-ID: <595903.006239922-sendEmail@kali>  
From: "support@cyberdefenders.org" <support@cyberdefenders.org>  
To: "joshua@cyberdefenders.org" <joshua@cyberdefenders.org>  
Subject: Immediate responses  
Date: Fri, 1 Oct 2021 12:31:54 +0000  
X-Mailer: sendEmail-1.56  
MIME-Version: 1.0  
Content-Type: multipart/mixed; boundary="----MIME delimiter for sendEmail-803805.438959077"  
  
This is a multi-part message in MIME format. To properly display this message you need a  
MIME-Version 1.0 compliant Email program.  
  
-----MIME delimiter for sendEmail-803805.438959077  
Content-Type: text/plain;  
charset="iso-8859-1"  
Content-Transfer-Encoding: 7bit  
  
Hi Joshua,  
  
There has been an issue with our web server that need fixing ASAP.  
Please find the web server details in the attached document.  
  
23 client pkts, 7 server pkts, 12 turns.  
Entire conversation (19 kB) Show and save data as ASCII Stream 1183  
Find: Filter Out This Stream Print Save as... Back Close

Answer: 55e7660d9b21ba07fc34630d49445030

16. What is the CVE number the attacker tried to exploit using the malicious document?

Zrzuty pliku wrzucamy na stronę virustotal jest to sandbox zawierający silniki wielu antywirusów.

Answer: CVE-2021-40444

17. The malicious document file contains a URL to a malicious HTML file. Provide the URL for this file.

Filtrujemy w poszukiwaniu ruchu http z adresem docelowym atakującego:

ip.dst == 192.168.112.128 and http						
No.	Time	Source	Destination	Protocol	Length	Info
2940	60.364064	192.168.112.139	192.168.112.128	HTTP	229	OPTIONS / HTTP/1.1
2949	60.403141	192.168.112.139	192.168.112.128	HTTP	217	HEAD /word.html HTTP/1.1
2959	60.419163	192.168.112.139	192.168.112.128	HTTP	229	OPTIONS / HTTP/1.1
2968	60.438714	192.168.112.139	192.168.112.128	HTTP	381	GET /word.html HTTP/1.1
2985	60.449673	192.168.112.139	192.168.112.128	HTTP	200	HEAD /word.html HTTP/1.1
2996	60.457949	192.168.112.139	192.168.112.128	HTTP	200	HEAD /word.html HTTP/1.1
3007	60.470053	192.168.112.139	192.168.112.128	HTTP	229	OPTIONS / HTTP/1.1
3016	60.498931	192.168.112.139	192.168.112.128	HTTP	217	HEAD /word.html HTTP/1.1
3026	60.510041	192.168.112.139	192.168.112.128	HTTP	229	OPTIONS / HTTP/1.1
3035	60.521039	192.168.112.139	192.168.112.128	HTTP	444	GET /word.html HTTP/1.1
3046	60.529126	192.168.112.139	192.168.112.128	HTTP	200	HEAD /word.html HTTP/1.1
3057	60.536577	192.168.112.139	192.168.112.128	HTTP	200	HEAD /word.html HTTP/1.1
3076	61.461908	192.168.112.139	192.168.112.128	HTTP	423	GET /word.cab HTTP/1.1
GET /word.html HTTP/1.1\r\n						
Accept: */*\r\n						
User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.2; Win64; x64; Trident/7.0; .NET4.0C; .NET4.0E; .NET						
UA-CPU: AMD64\r\n						
Accept-Encoding: gzip, deflate\r\n						
Host: 192.168.112.128\r\n						
Connection: Keep-Alive\r\n						
\r\n						
[Full request URI: http://192.168.112.128/word.html]						
[HTTP request 1/1]						
[Response in frame: 2977]						

Answer: <http://192.168.112.128/word.html>

20. The malicious HTML contains a js code that points to a malicious CAB file. Provide the URL to the CAB file?

Na Screenshot z zadania 17 w pakiecie 3076 widzimy plik /word.cab jego lokalizacja jest bardzo podobna:

Answer: <http://192.168.112.128/word.cab>

21. The exploit takes advantage of a CAB vulnerability. Provide the vulnerability name?

Trochę googlowania i odpowiedzią jest podatność ZipSlip jest to krytyczna podatność która umożliwia wykonanie kodu umieszczonego w pliku html.

Answer: ZipSlip

22. The CAB file contains a malicious dll file. What is the tool used to generate the dll?

Funkcje generowania złośliwych bibliotek dll lub innych revshelli daje np. pakiet metasploit

Answer: metasploit



23. What is the path of the dropped malicious dll file? Replace your username with IEUser

Po analizie pliku word.html i deobfuskacji możemy znaleźć lokalizację gdzie złośliwy plik został zapisany:

Answer: C:\Users\IEUser\AppData\Local\Temp\msword.inf

24. Analyzing the dll file what is the API used to write the shellcode in the process memory?

Możemy się tego dowiedzieć poprzez wypakowanie pliku z archiwum word.cap za pomocą polecenia: 7z e word.cab następnie sprawdzamy stringi wyodrębnionego pliku gdzie znajdujemy odpowiedź:

```
michal@linux:~/Desktop$ strings msword.inf -n 8
!This program cannot be run in DOS mode.
.text$mn
.idata$5
.rdata$zzzdbg
.idata$2
.idata$3
.idata$4
.idata$6
CloseHandle
ReleaseSemaphore
WaitForSingleObject
CreateEventA
OpenEventA
ExitThread
ResumeThread
CreateProcessA
GetThreadContext
SetThreadContext
VirtualAllocEx
WriteProcessMemory
CreateSemaphoreA
KERNEL32.dll
DSS$F5~x70
```

Answer: WriteProcessMemory

25. Extracting the shellcode from the dll file. What is the name of the library loaded by the shellcode

tcp.stream eq 1262

No.	Time	Source	Destination	Protocol	Length	Info
3552	86.048816	192.168.112.128	192.168.112.139	TCP	1514	443 → 50102
3553	86.048816	192.168.112.128	192.168.112.139	TCP	1514	443 → 50102
3554	86.048816	192.168.112.128	192.168.112.139	TCP	1514	443 → 50102
3555	86.048816	192.168.112.128	192.168.112.139	TCP	1514	443 → 50102
3556	86.048816	192.168.112.128	192.168.112.139	TCP	1514	443 → 50102
3557	86.048816	192.168.112.128	192.168.112.139	TCP	1514	443 → 50102
3558	86.048816	192.168.112.128	192.168.112.139	TCP	1514	443 → 50102
3559	86.048816	192.168.112.128	192.168.112.139	TCP	1514	443 → 50102
3560	86.048816	192.168.112.128	192.168.112.139	TCP	1514	443 → 50102
3561	86.048816	192.168.112.128	192.168.112.139	TCP	1514	443 → 50102
3562	86.048816	192.168.112.128	192.168.112.139	TCP	1514	443 → 50102
3563	86.048816	192.168.112.128	192.168.112.139	TCP	1514	443 → 50102
3564	86.048816	192.168.112.128	192.168.112.139	TCP	1514	443 → 50102

Frame 3561: 1514 bytes on wire (12112 bits), 1514 bytes captured (12112 bits) on 0  
Ethernet II, Src: VMware\_61:f9:84 (00:0c:29:61:f9:84), Dst: VMware\_b7:ca:91 (00:0c:29:b7:ca:91)  
Internet Protocol Version 4, Src: 192.168.112.128, Dst: 192.168.112.139  
Transmission Control Protocol, Src Port: 443, Dst Port: 50102, Seq: 149405, Ack: 327, Len: 1514

0000 00 0c 29 b7 ca 91 00 0c 29 61 f9 84 00 00 45 00 .....)ja...E  
0010 05 dc f4 1d 40 00 40 06 de a1 c8 a8 70 80 c8 a8 ...@...p...  
0020 70 0b 01 bb c3 b6 da 44 7a a3 bd 0b db b6 50 10 .....Dz...P  
0030 01 f5 ff 31 00 00 49 0e 06 6f 57 00 00 57 49 4e ...1..In fow..WIN  
0040 49 4e 45 54 2e 04 0c 00 09 00 57 69 0e 48 74 ..NET.dll...WinHT  
0050 74 70 43 72 61 03 6b 55 72 6c 09 0f 00 57 69 6e tpCrackU r...WinHT  
0060 48 74 74 70 4f 70 65 0e 00 07 00 57 69 6e 48 74 httpOpen...WinHT

Se...t.WSADuplicateSocketA...getaddrinfo...freeaddrinfo...WS2\_32.dll...CryptDecodeObject  
Ex...CryptImportPublicKeyInfo...F.CertGetCertificateContextProperty.CRYPT32.dll...InternetCr  
ackURL...InternetOpenW...k.InternetCloseHandle...r.InternetConnectW...InternetReadFile...Int  
ernetSetOptionW...X.HttpOpenRequestW...HttpSendRequestW...Z.HttpQueryInfoW...WININET.dll  
...WinHttpCrackURL...WinHttpOpen...WinHttpCloseHandle...WinHttpConnect...WinHttpReadData...  
WinHttpRequestOption...WinHttpSetOption...WinHttpOpenRequest...WinHttpSendRequest...WinHT  
pReceiveResponse...WinHttpQueryHeaders...WinHttpGetProxyForUrl...  
WinHttpGetIEProxyConfigForCurrentUser.WINHTTP.dll...VirtualAllocEx...OpenProcess...GetCur  
rentProcess...GetLastError...WriteProcessMemory...R.CloseHandle...DuplicateHandle...CreateE  
ventW...b.FreeLibrary...E.GetProcAddress...VirtualAlloc...VirtualFree...VirtualQueryEx...Op  
enThread...s.GetLastError...SuspendThread...ResumeThread...Sleep...LoadLibraryA...GetVers  
ion...client pkt, 123 server pkts, 1 turn.  
Entire conversation (176 kb) Show and save data as ASCII Stream 1262

Adres atakującego generował bardzo dużo ruchu z portu 443 do ofiary w pakiecie 3561 można odczytać bibliotekę .dll która prawdopodobnie była załadowana do pamięci

26. Which port was configured to receive the reverse shell?

W dalszym etapie ataku adres ofiary wysyła bardzo dużo flag tcp.ack tak jakby wysyłał jakieś dane jest to typowe zachowanie revershella, Oprócz tego wysyła żądania do serwera http na adres atakującego z którego zostały wcześniej pobrane dwa złośliwe pliki

ip.src == 192.168.112.139 and ip.dst == 192.168.112.128						
No.	Time	Source	Destination	Protocol	Length	Info
5142	137.765661	192.168.112.139	192.168.112.128	TCP	64294	50103 → 443 [ACK] Seq=11151514 Ack=259637 Win=524032 Len=6424...
5143	137.765911	192.168.112.139	192.168.112.128	TCP	64294	50103 → 443 [ACK] Seq=11215754 Ack=259637 Win=524032 Len=6424...
5144	137.766163	192.168.112.139	192.168.112.128	TCP	64294	50103 → 443 [ACK] Seq=11279994 Ack=259637 Win=524032 Len=6424...
5145	137.766405	192.168.112.139	192.168.112.128	TCP	64294	50103 → 443 [ACK] Seq=11344234 Ack=259637 Win=524032 Len=6424...
5146	137.766639	192.168.112.139	192.168.112.128	TCP	64294	50103 → 443 [ACK] Seq=11406474 Ack=259637 Win=524032 Len=6424...
5147	137.766853	192.168.112.139	192.168.112.128	TCP	17574	50103 → 443 [ACK] Seq=11472714 Ack=259637 Win=524032 Len=1752...
5161	137.767086	192.168.112.139	192.168.112.128	TCP	64294	50103 → 443 [ACK] Seq=11490234 Ack=259637 Win=524032 Len=6424...
5162	137.767375	192.168.112.139	192.168.112.128	TCP	64294	50103 → 443 [ACK] Seq=11554474 Ack=259637 Win=524032 Len=6424...
5163	137.767605	192.168.112.139	192.168.112.128	HTTP	3446	POST /Trm_Qwz3GPappqinyPBWSgeXk3iW8b0Mcj1yj -CAk3zd2WRSqc9ItyY...
5177	138.117526	192.168.112.139	192.168.112.128	HTTP	362	GET /Trm_Qwz3GPappqinyPBWSgeXk3iW8b0Mcj1yj -CAk3zd2WRSqc9ItyY...
5179	138.141122	192.168.112.139	192.168.112.128	HTTP	362	GET /Trm_Qwz3GPappqinyPBWSgeXk3iW8b0Mcj1yj -CAk3zd2WRSqc9ItyY...
5181	138.145380	192.168.112.139	192.168.112.128	HTTP	362	GET /Trm_Qwz3GPappqinyPBWSgeXk3iW8b0Mcj1yj -CAk3zd2WRSqc9ItyY...
5183	138.147620	192.168.112.139	192.168.112.128	HTTP	362	GET /Trm_Qwz3GPappqinyPBWSgeXk3iW8b0Mcj1yj -CAk3zd2WRSqc9ItyY...
5185	138.155842	192.168.112.139	192.168.112.128	TCP	388	50103 → 443 [PSH, ACK] Seq=11623338 Ack=260357 Win=525056 Len=...
5186	138.156458	192.168.112.139	192.168.112.128	TCP	64294	50103 → 443 [ACK] Seq=11623672 Ack=260357 Win=525056 Len=6424...
5187	138.156862	192.168.112.139	192.168.112.128	TCP	64294	50103 → 443 [ACK] Seq=11687912 Ack=260357 Win=525056 Len=6424...
5188	138.157110	192.168.112.139	192.168.112.128	TCP	64294	50103 → 443 [ACK] Seq=11752152 Ack=260357 Win=525056 Len=6424...
5190	138.157378	192.168.112.139	192.168.112.128	TCP	64294	50103 → 443 [ACK] Seq=11816392 Ack=260357 Win=525056 Len=6424...
5192	138.157614	192.168.112.139	192.168.112.128	TCP	64294	50103 → 443 [ACK] Seq=11880632 Ack=260357 Win=525056 Len=6424...
5195	138.157865	192.168.112.139	192.168.112.128	TCP	64294	50103 → 443 [ACK] Seq=11944872 Ack=260357 Win=525056 Len=6424...
5197	138.158122	192.168.112.139	192.168.112.128	TCP	64294	50103 → 443 [ACK] Seq=12009112 Ack=260357 Win=525056 Len=6424...
▶ Frame 3547: 54 bytes on wire (432 bits), 54 bytes captured (432 bits)						
▶ Ethernet II, Src: VMware_b7:ca:91 (00:0c:29:b7:ca:91), Dst: VMware_61:f9:84 (00:0c:29:61:f9:84)						
▶ Internet Protocol Version 4, Src: 192.168.112.139, Dst: 192.168.112.128						
▶ Transmission Control Protocol, Src Port: 50103, Dest Port: 443, Seq: 337, Ack: 138435, Len: 0						

Answer: 443