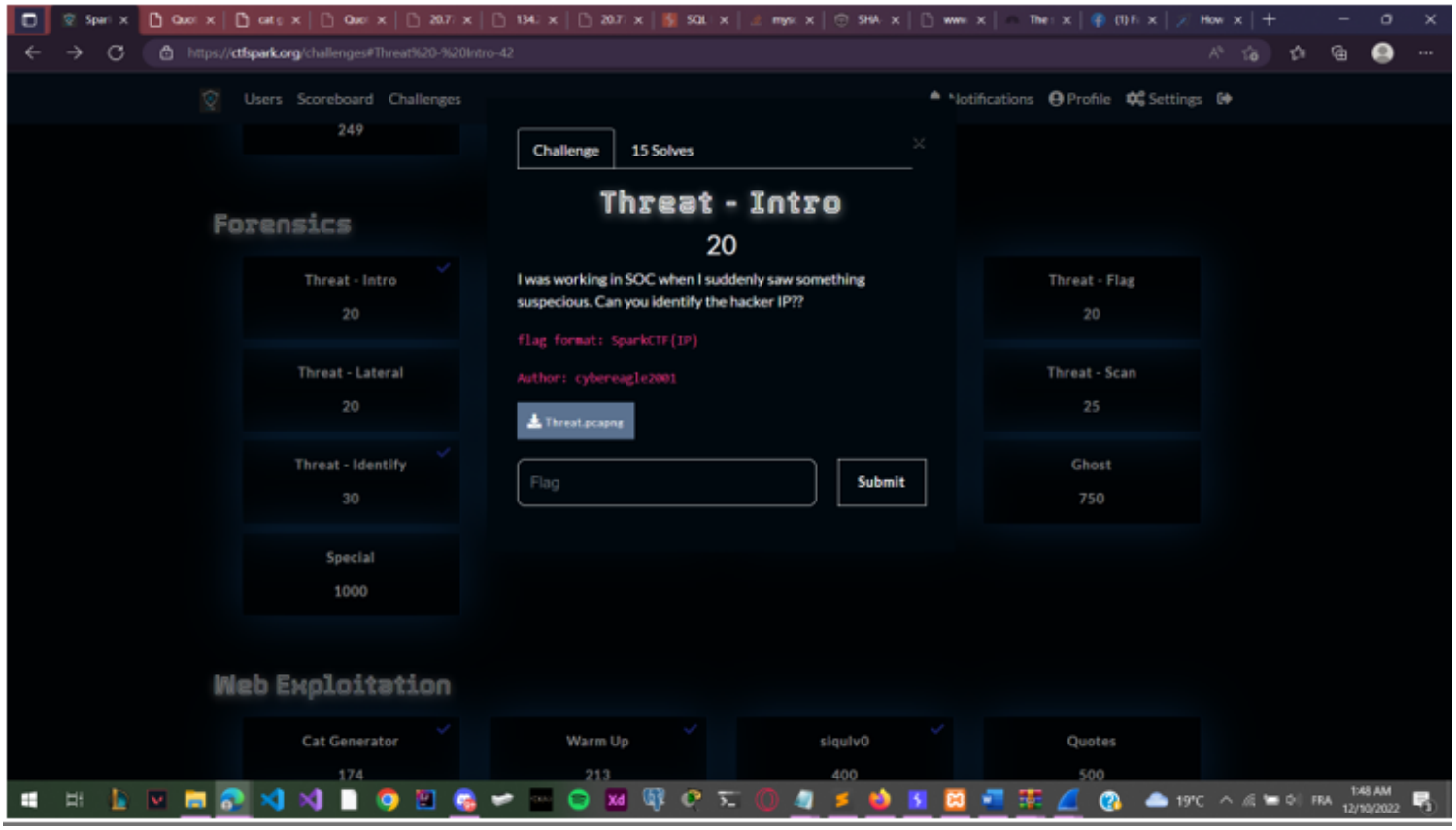
Threat - Forensics Series - Writeup

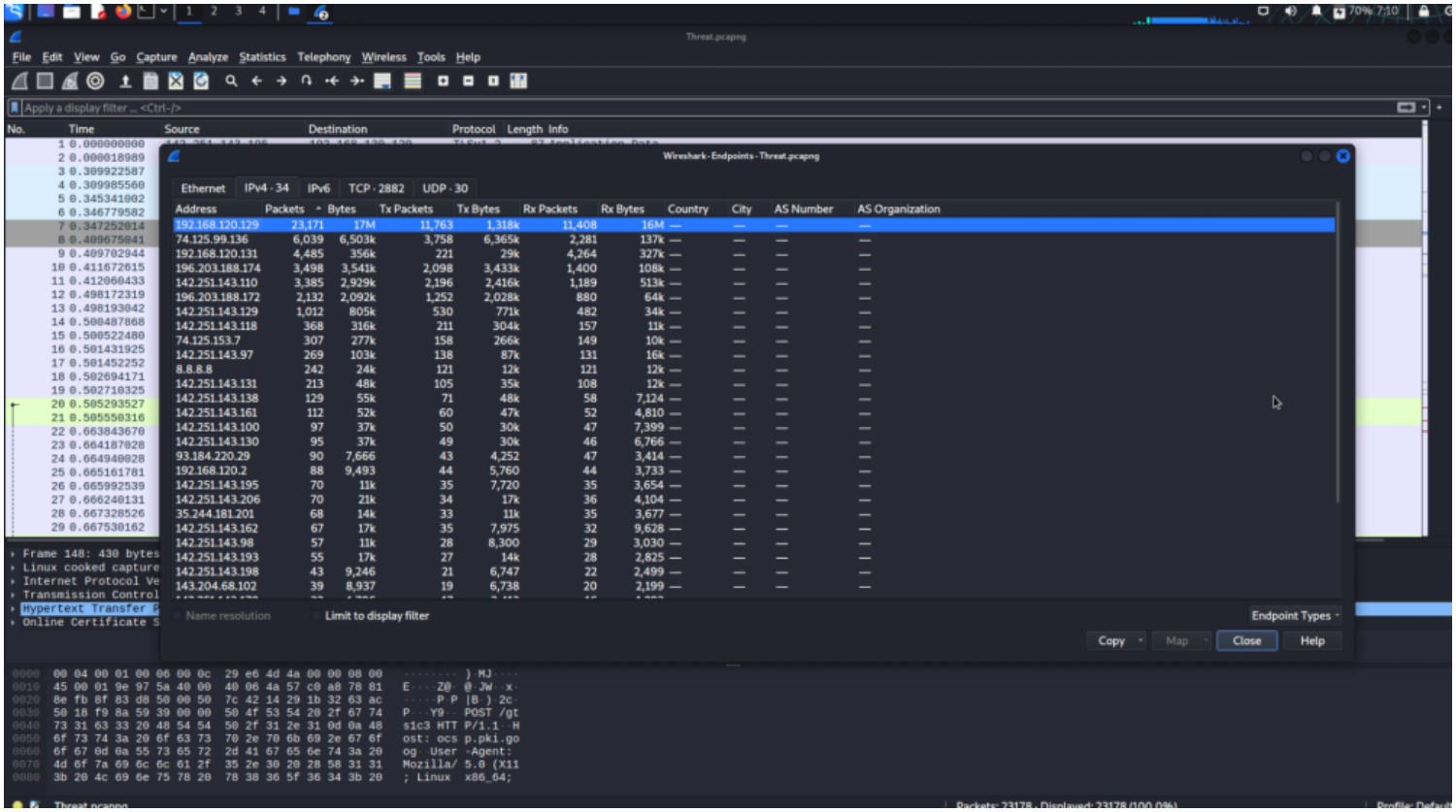
These are the writeups for my Threat forensics series tasks that were published during SparkCTF 2022. These tasks are based on real life scenarios where the CTF players should have basic understanding of network analysis. To create the tasks I used two VM's (kali linux for the attacker server and CentOS 7 for the victim machine). I used a simple reverse shell written in python and some known utilities like Nmap.

This Event was Organised by **Engineers Spark Community** :

Threat-Intro

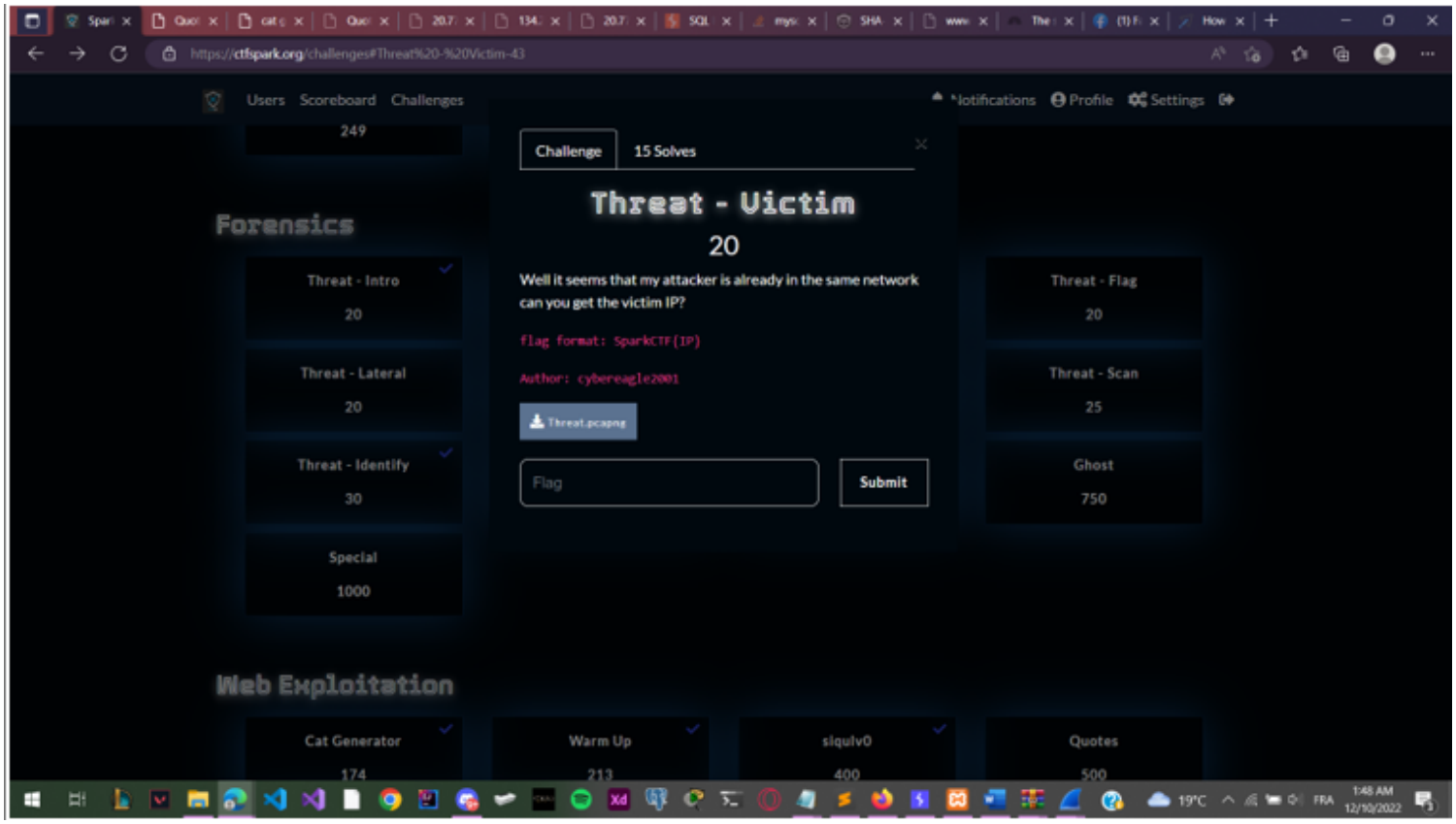


In all the tasks of the Threat Series the palyers will use the same capture file.
To identify the IP address of the hacker the player must identify the most suspicous IP in the network. It is the most redundant IP in the capture file.
To do so you will need to analyse the endpoints of the capture. You can visit > statistics > endpoints in wireshark menu. then we will choose IPv4 - 34 and make sure that the number of the packets is decreasing.
We can easly identify the most active IP in the network:

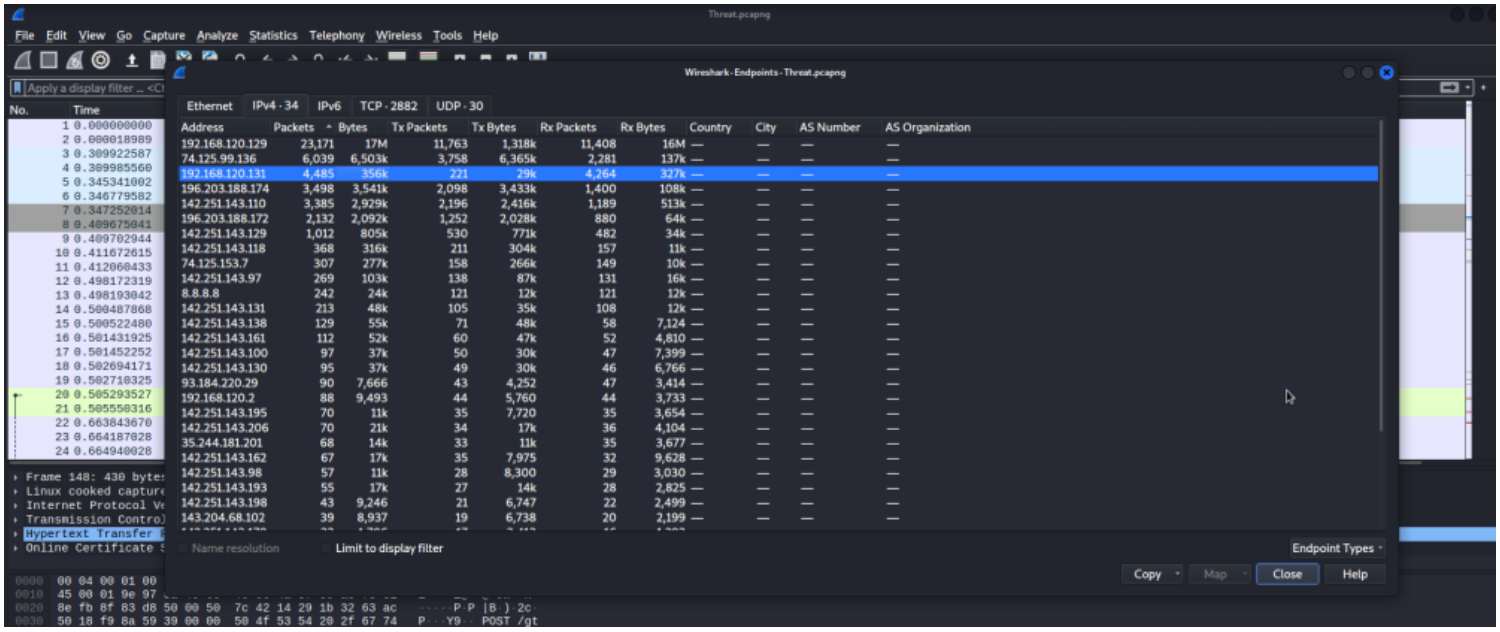


our flag is : SparkCTF{192.168.120.129}

Threat-Victim

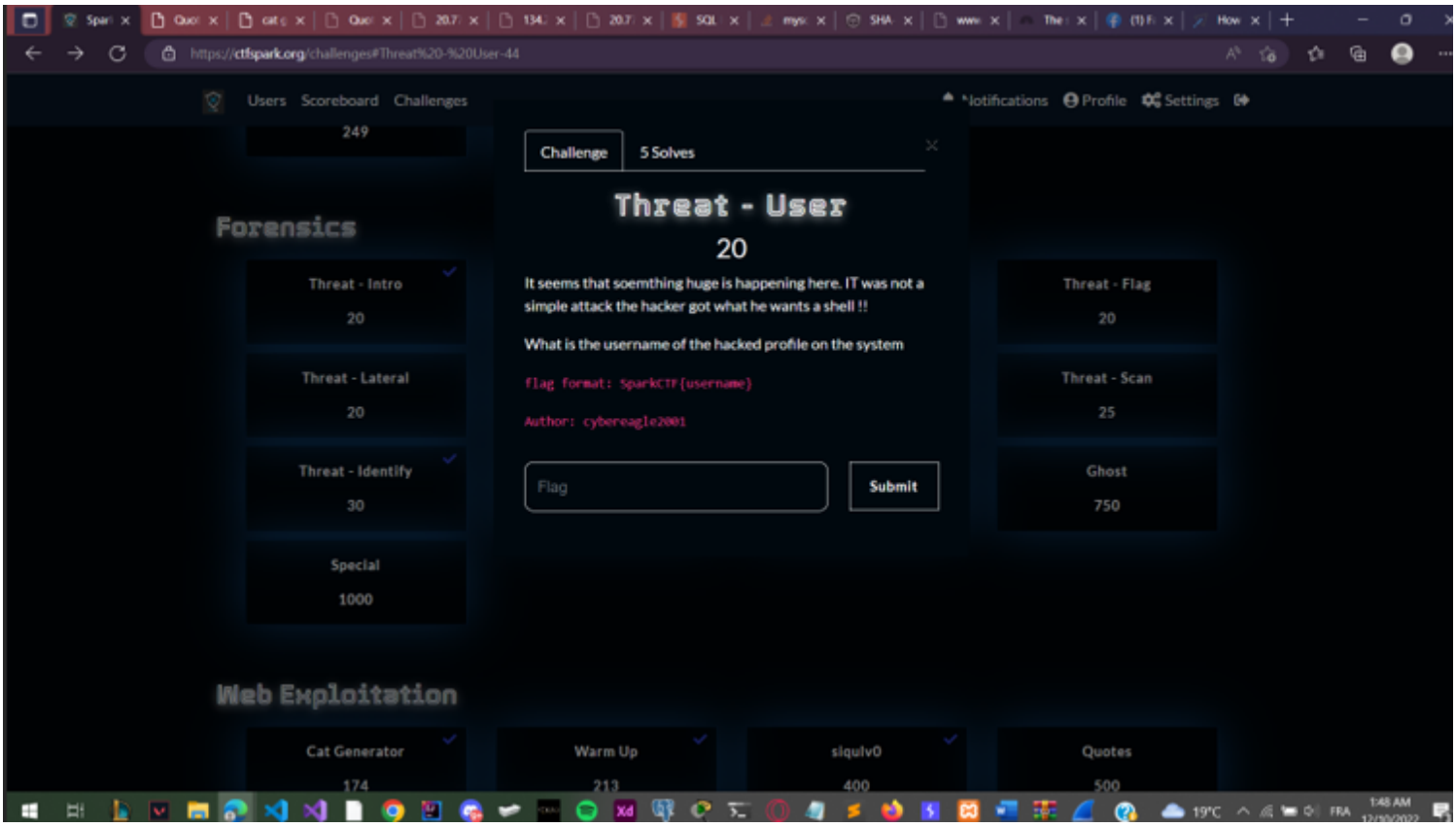


Using the same techniques in the first task the players will identify two Other IP's which are the second and third most active in our network.
To identify the victim we will use the hint available in the task description. “my attacker is already in the same network”.
that means that the victim IP should have the same IP class as our attacker's IP this is why we can neglect the second most active IP adress in the capture:

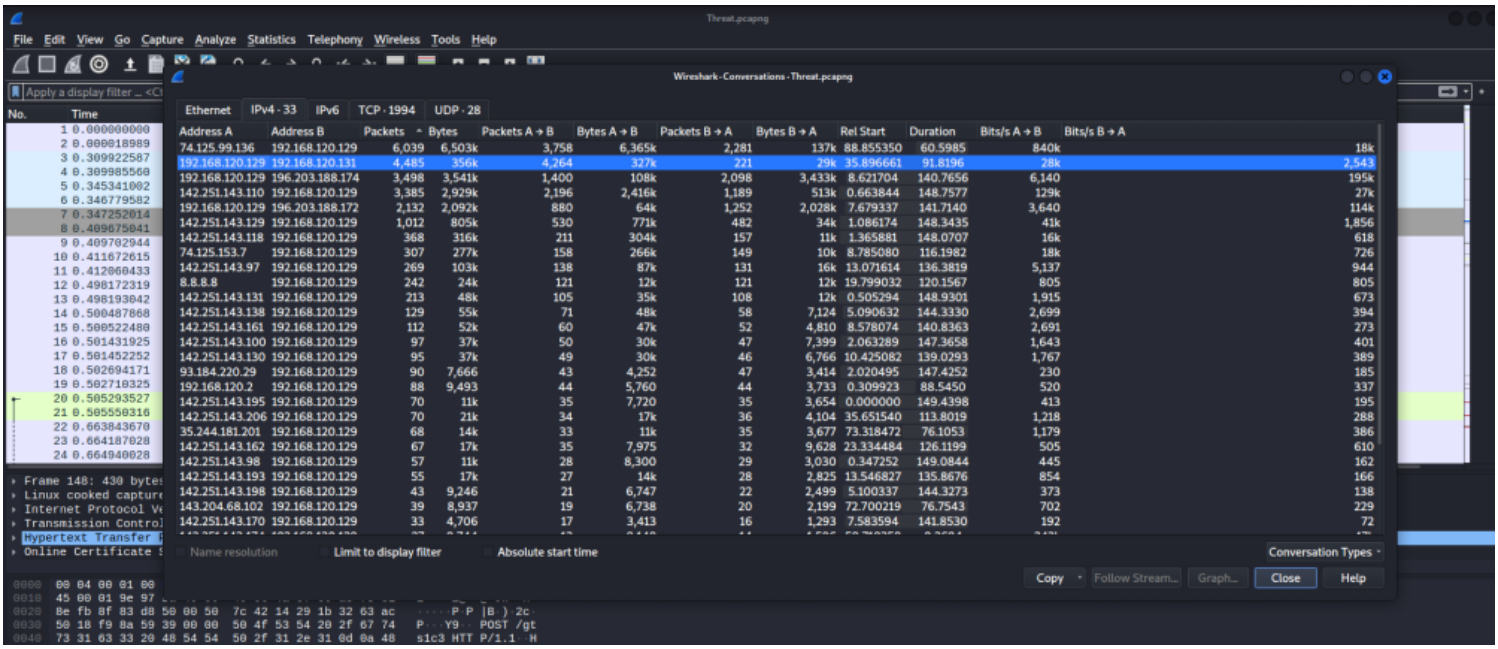


our flag is : SparkCTF{192.168.120.131}

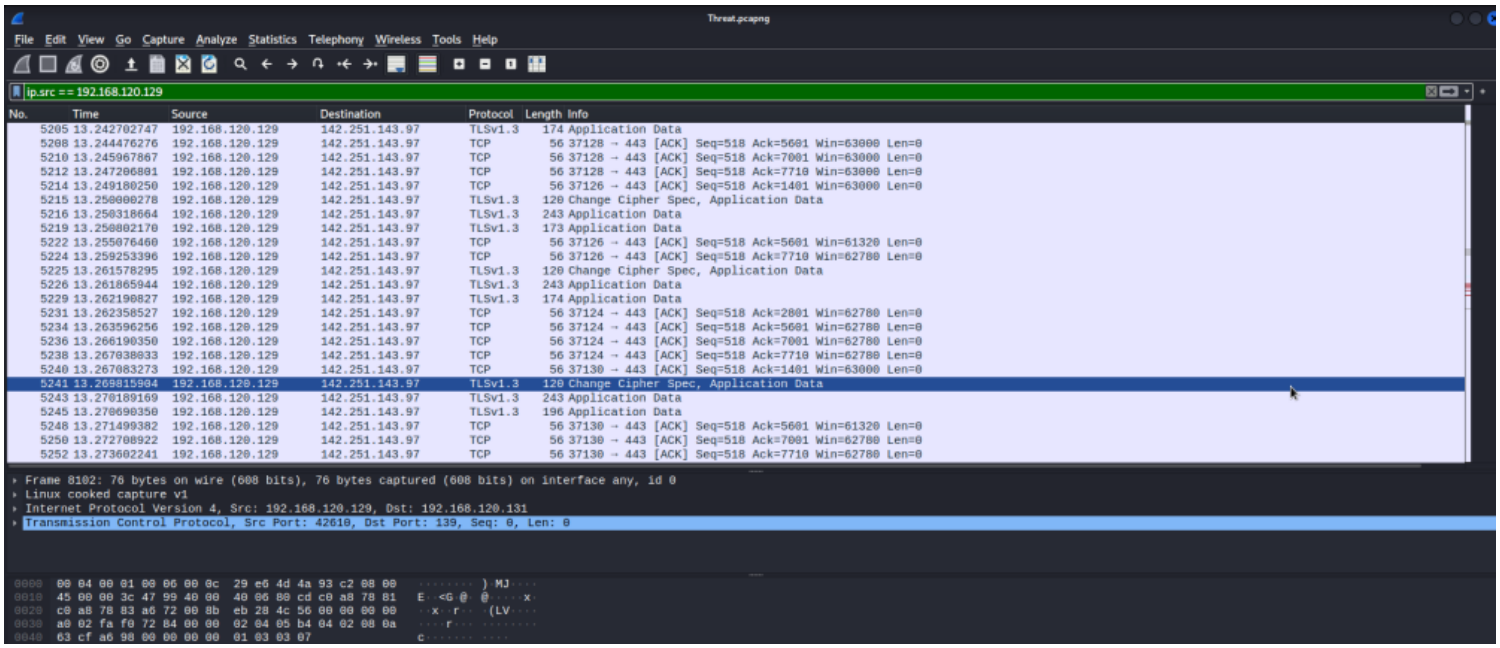
Threat-user



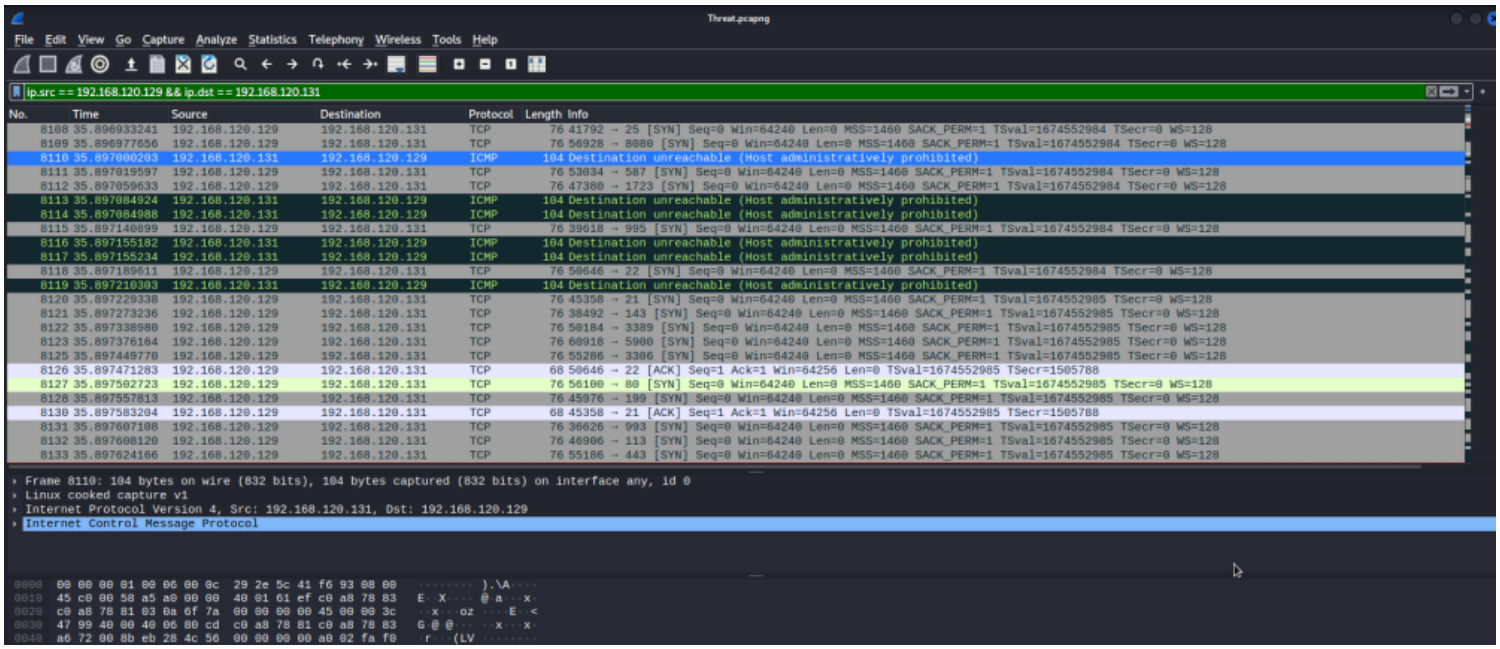
It seems that something huge is happening here, It was not a simple attack. The hacker got what he wanted, a shell!!
The players must identify the victim's username. To do so we need to find the packets whith the IP's found in the previous tasks.
We already knew that 192.168.120.129 is the hacker's IP and 192.168.120.131 is the victim's IP. Let's give a look to statistics again and try and understand the conversations we have captured.
> statistics > conversations



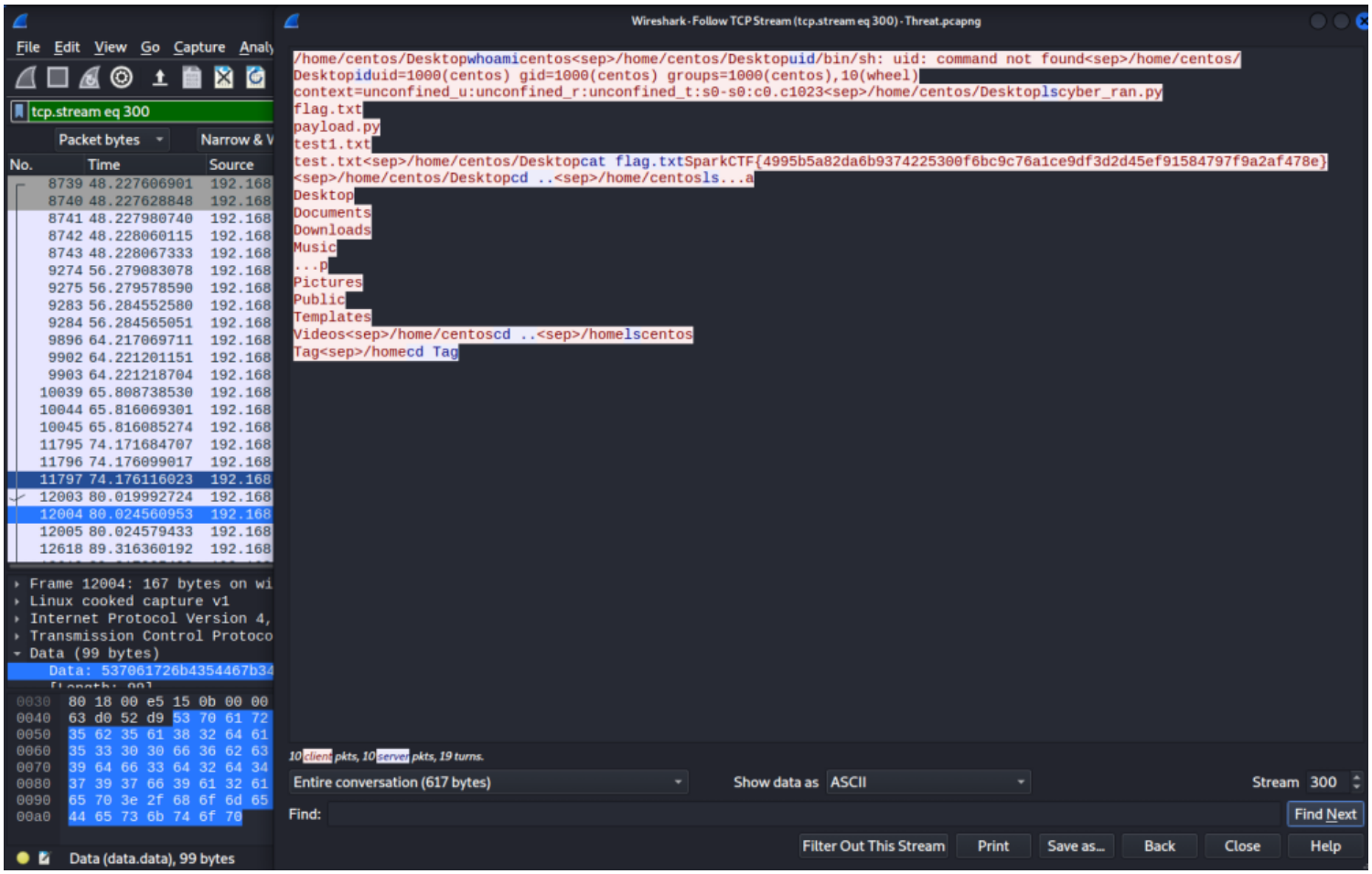
the second biggest number of packets is sent from 192.168.120.129 to 192.168.120.131 which is quite logica if we will tie it to the previous tasks.
Let's filter and see the packets where the IP source is 192.168.120.129. we can see that we still have a lot of packets to analyse.



this is why we will need to combine our filter with the destination IP in order to retrieve the only communication we have between the hacker and the victim:

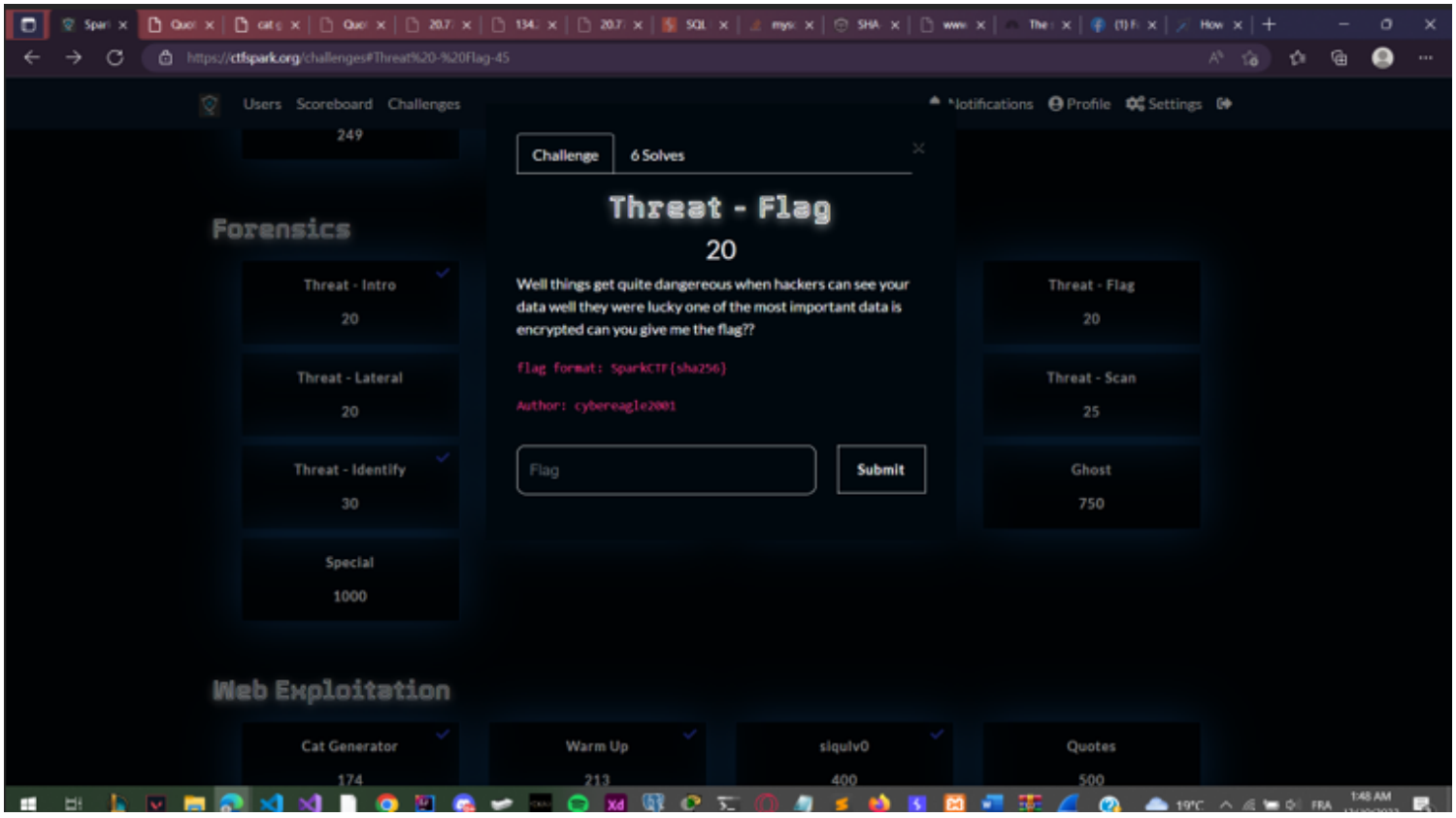


Things are getting more and more promising with fiewer packets and less protocols.
Let's follow the TCP streams and try to find if we captured a shell communication.

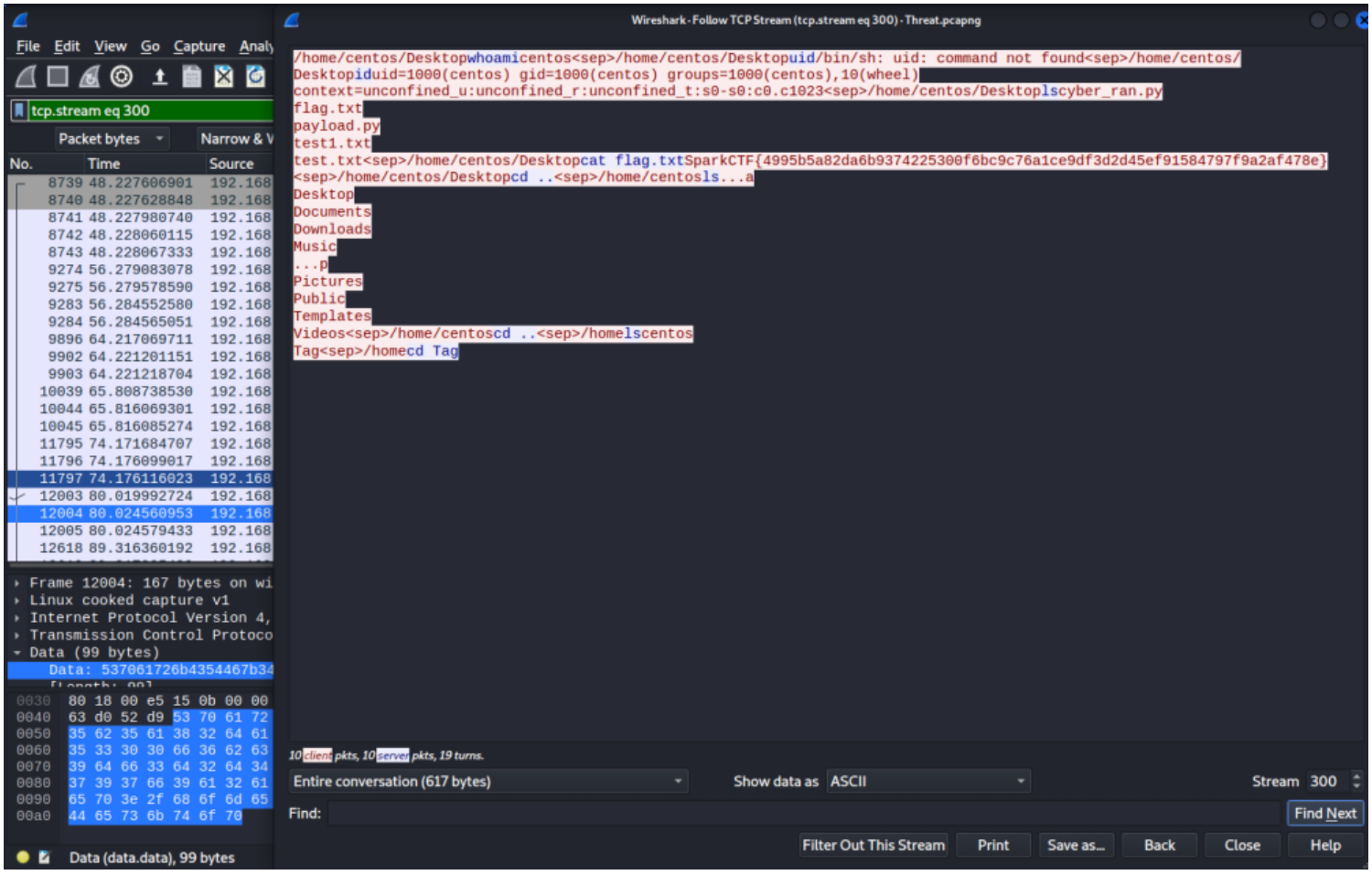


you can see that the hacker executed “whoami” on the victims machine and got "centos" as response.
The flag is : SparkCTF{centos}

Threat-Flag

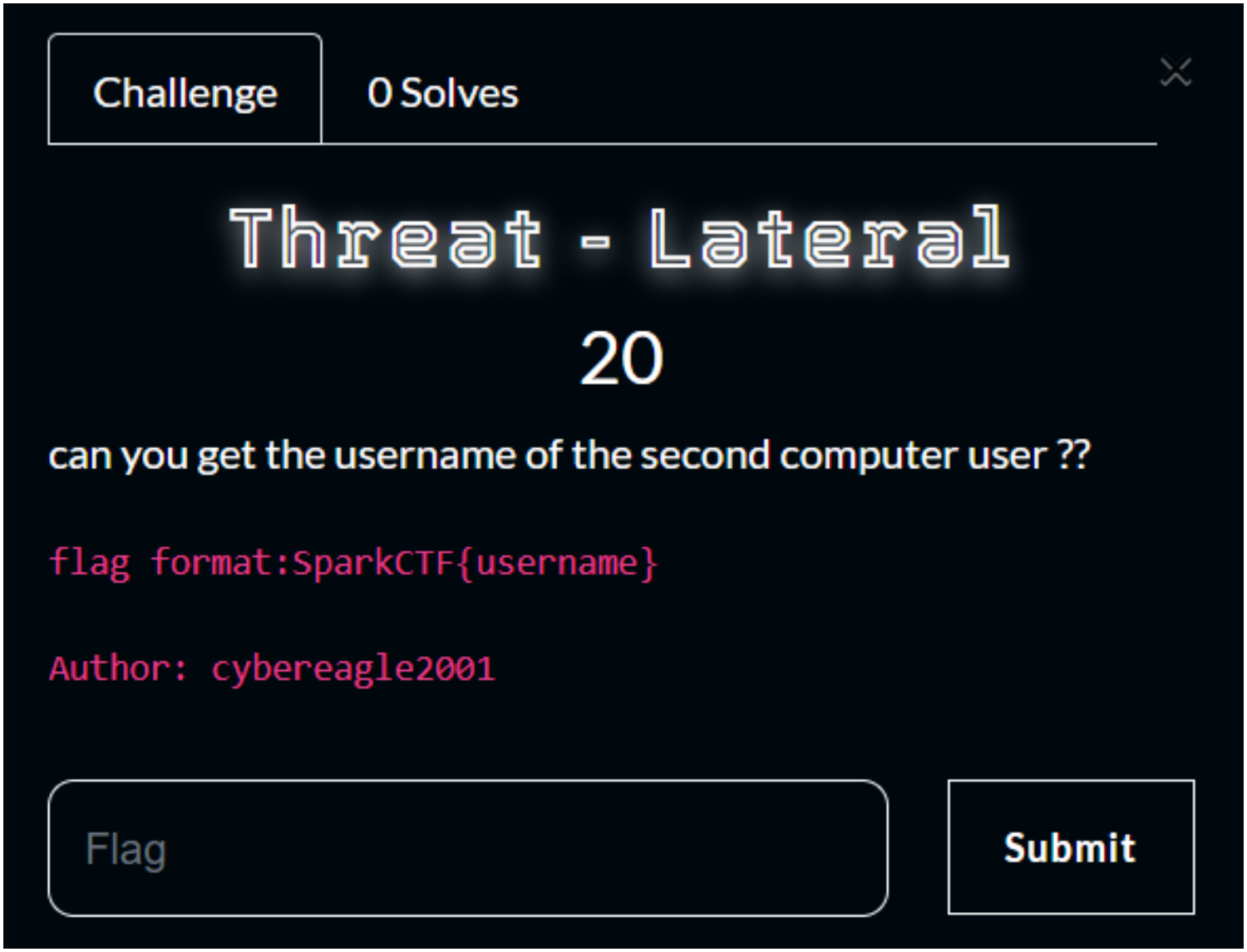


This task is quite easy if we have already solved the user task.
Examining the shell commands we will see that the hacker run the follwing : cat flag.txt

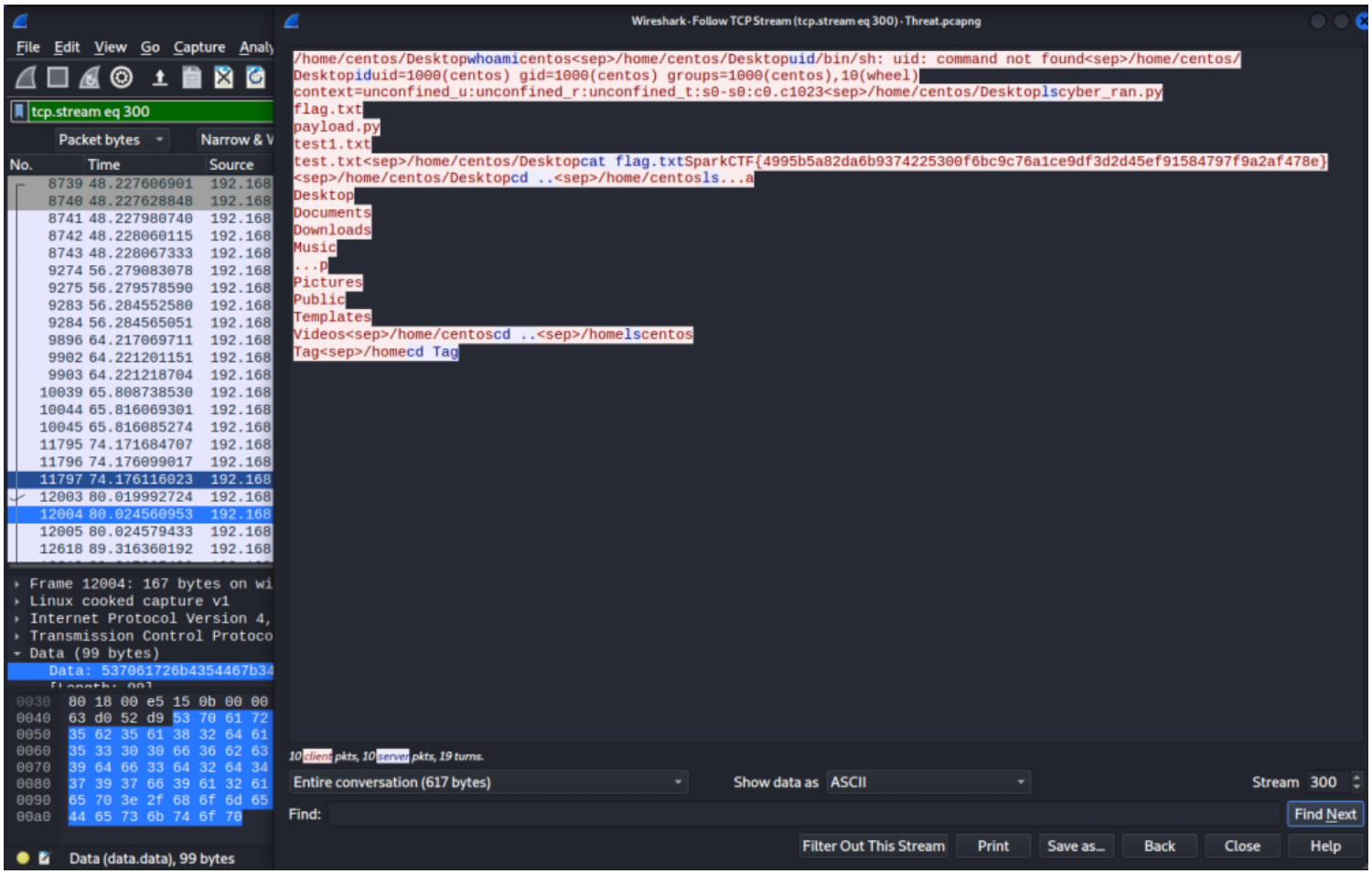


the flag is :
SparkCTF{4995b5a82da6b9374225300f6bc9c76a1ce9df3d2d45ef91584797f9a2af478e}

Threat-Lateral



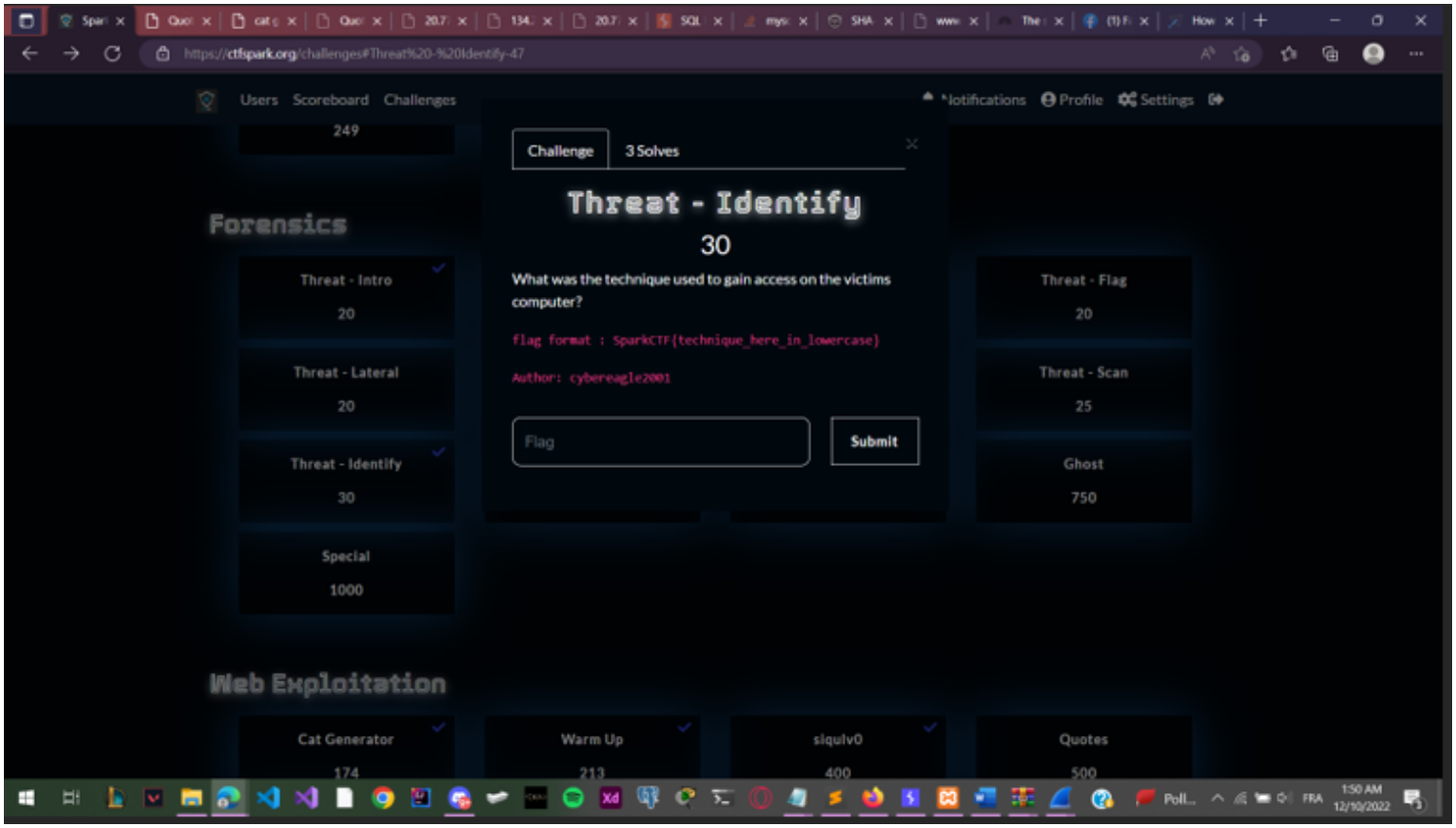
Lateral movement refers to the techniques that a cyberattacker uses, after gaining initial access, to move deeper into a network in search of sensitive data and other high-value assets. So let's see if our hacker
tried to reach another user account.
This task is also easy if we have already solved the previous tasks.



we can see that our hacker wanted to gain access to Tag /home directory.

Flag : SparkCTF{Tag}

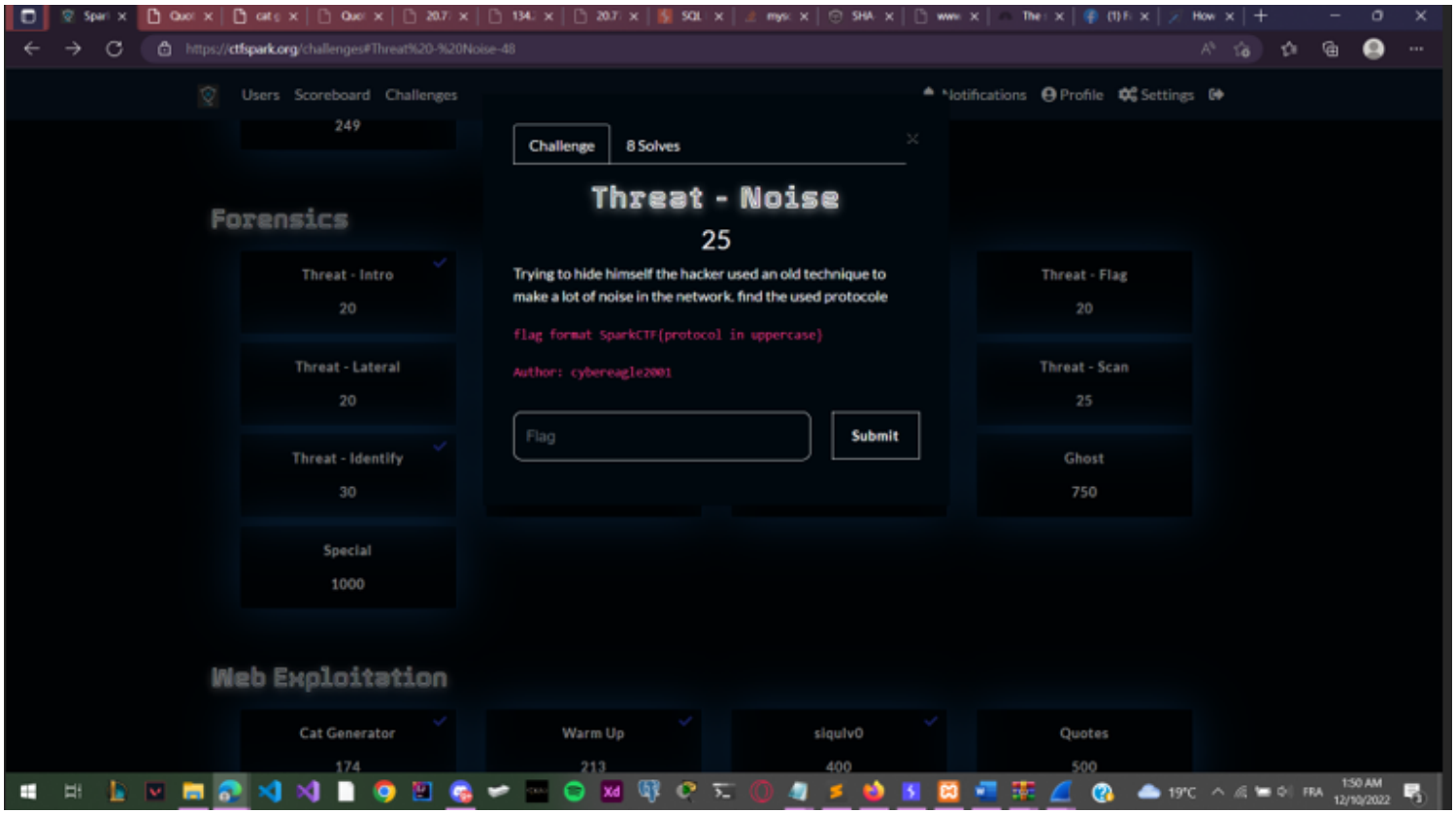
Threat-Identify



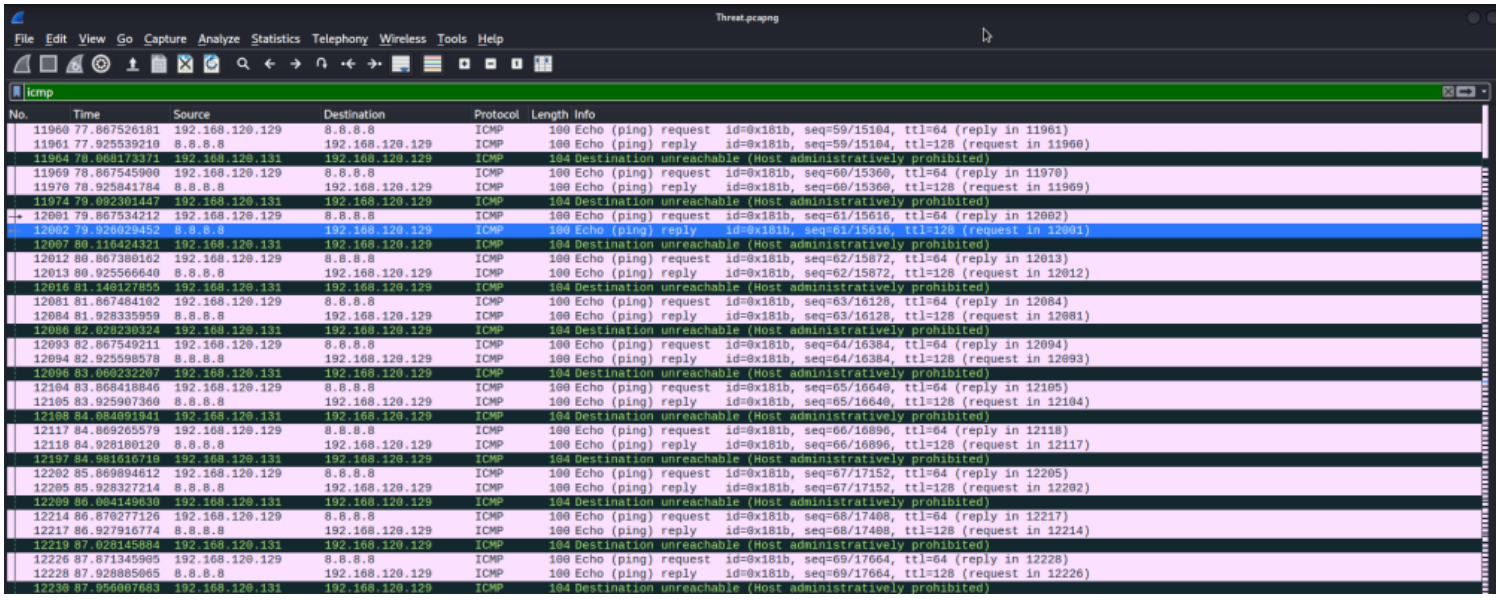
This is not a tachnical challenge, it's more about your knowledge. What was the technique used to gain access.
The hacker actually had the chance to execute commands on the victim computer but we can see that the communication was establsihed from the victim IP.
this is what it's called revere shell

Flag: SparkCTF{reverse_shell}

Threat-Noise

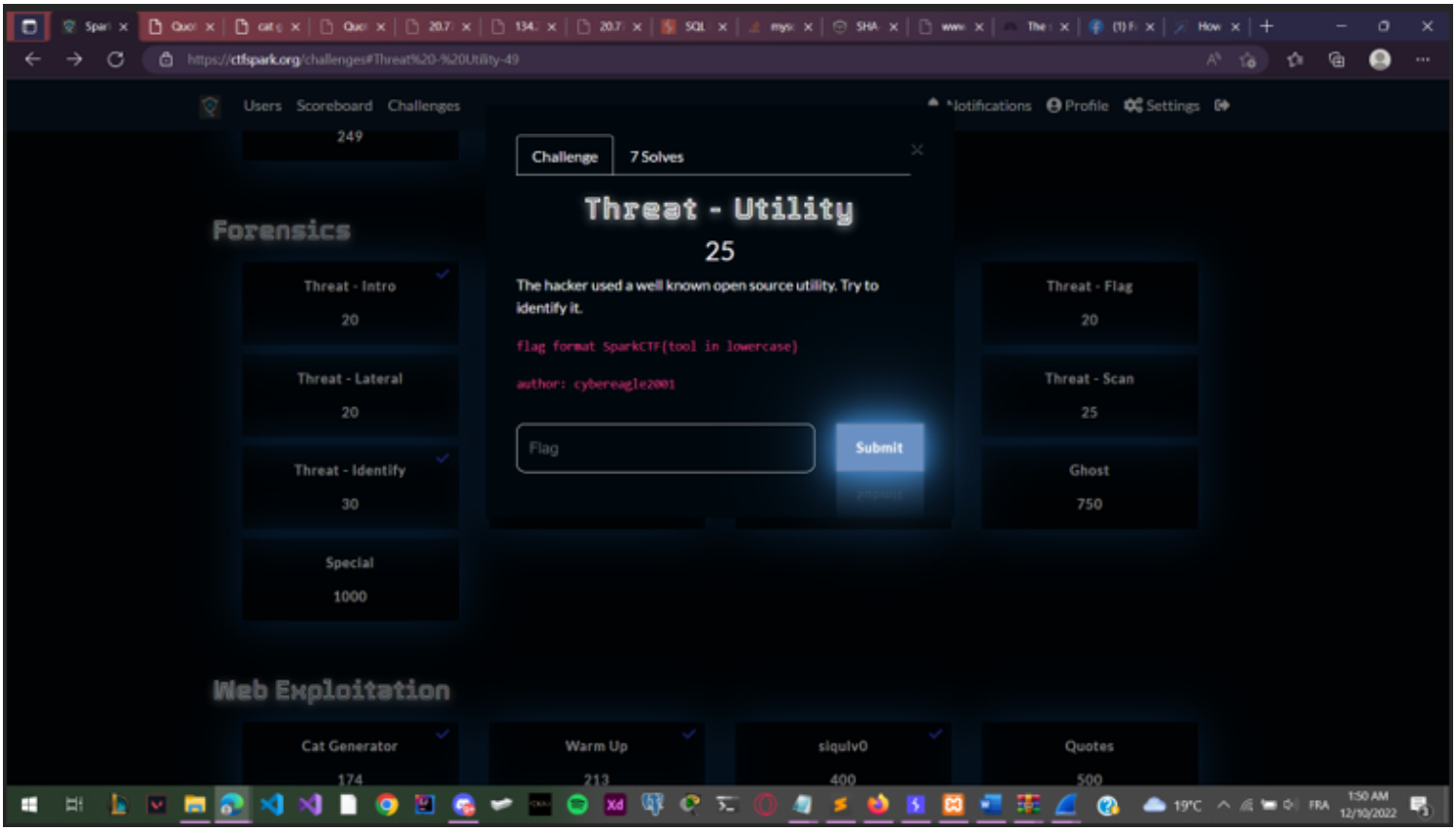


To hide himself the hacker used an old technique.
This is what the description says. But let's go back and see why our network capture is quite big.
If you will analyse the file precisely you can easly identify the huge ammount of ping request sent from the hacker ip.



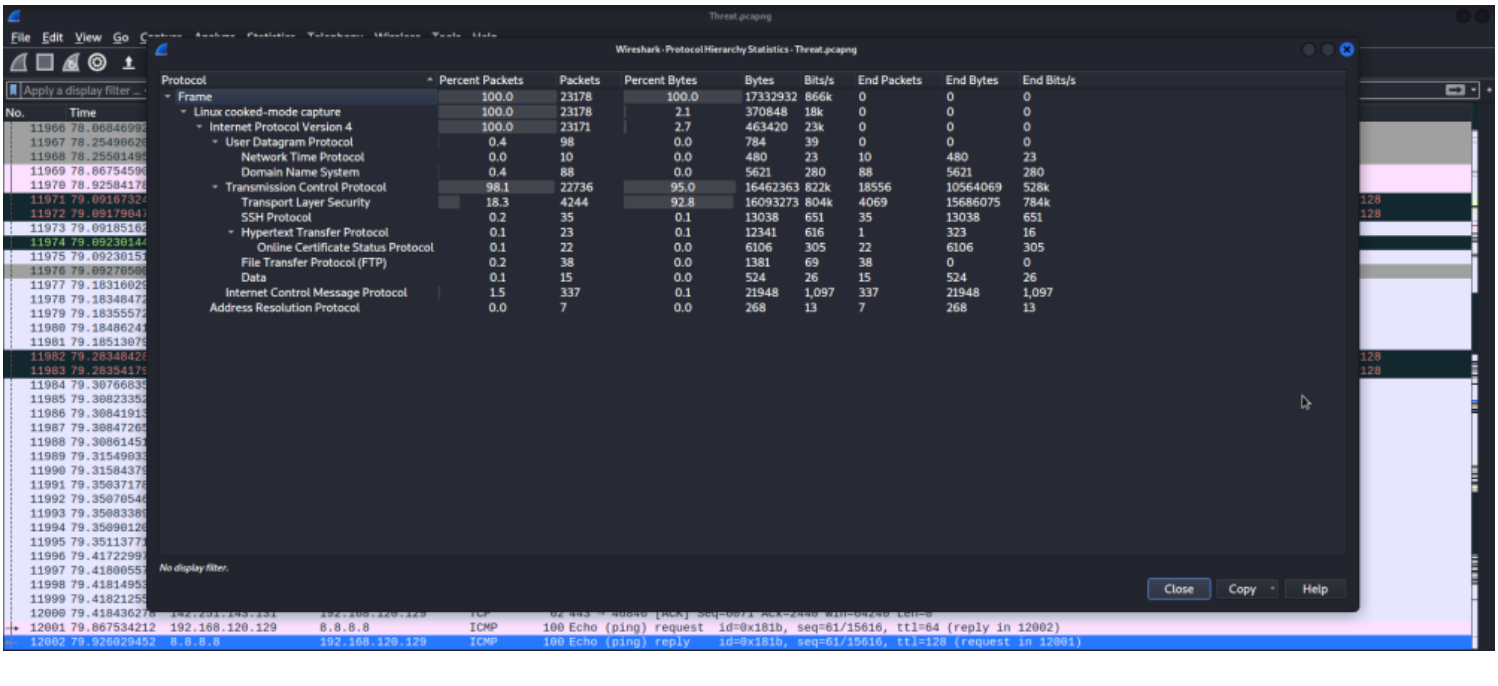
flag: SparkCTF{ICMP}

Threat-Utility

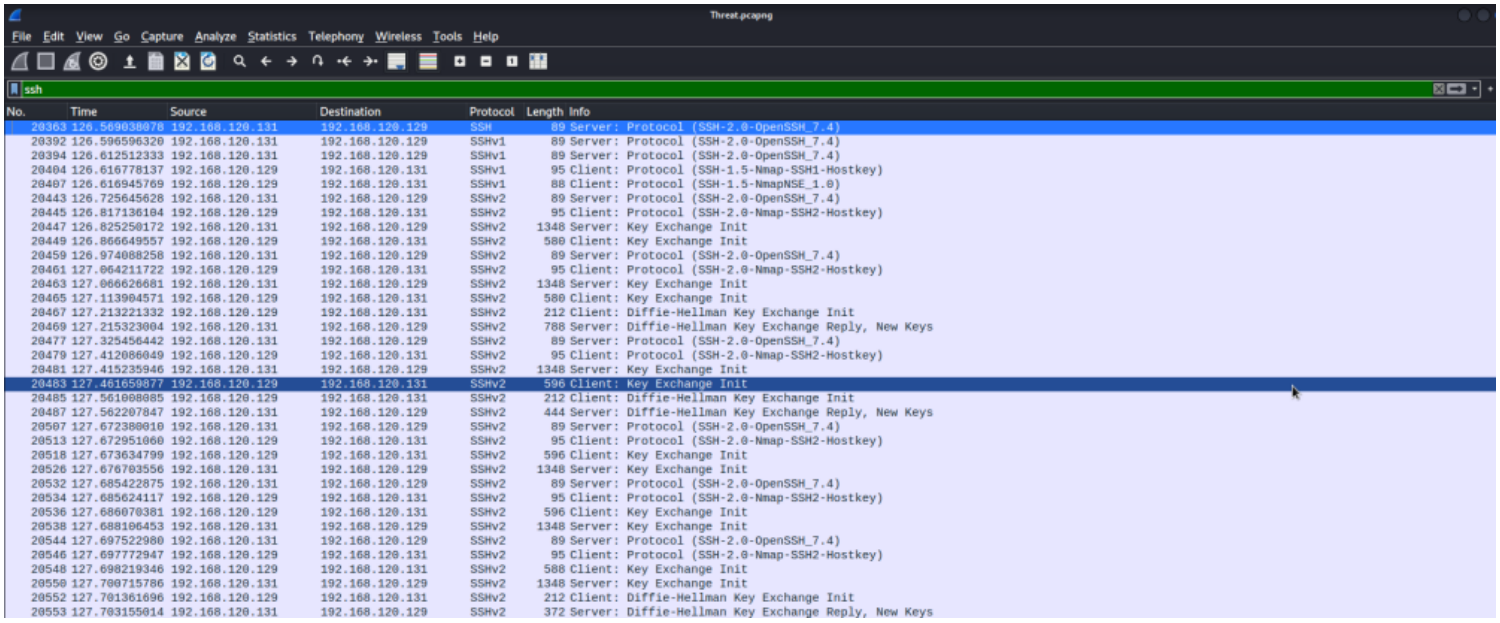


A well known utility used during the attacks???? Le's see what the network capture is hiding!

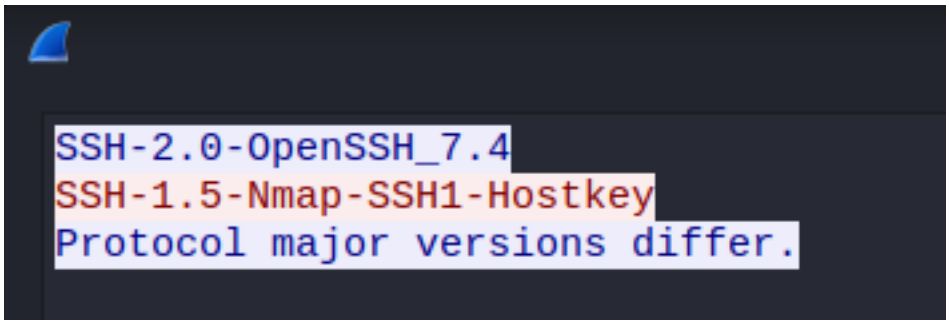
first we will analyse all the possible protocols that are availible at our capture.



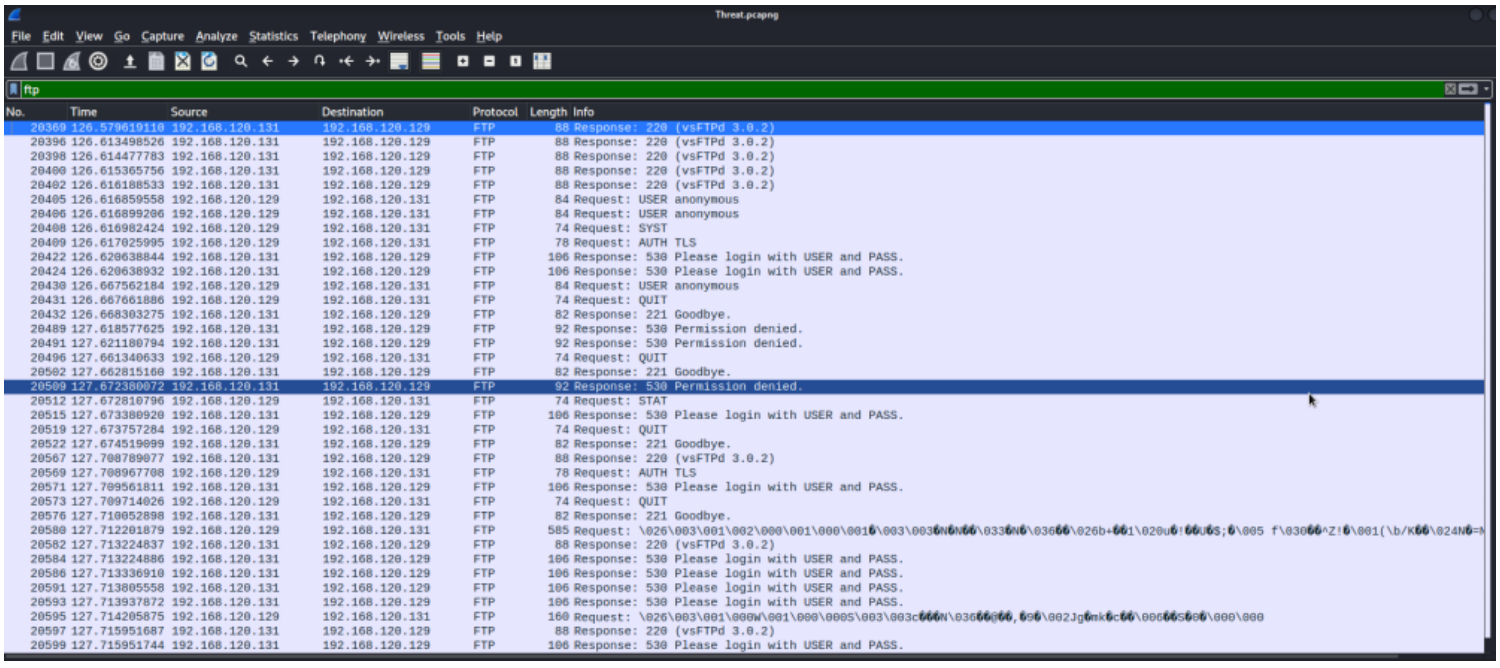
the most attractive ones are the FTP and the SSH. Why?? Well because the attacker didn't use SSH as a protocol to gain access to the shell and he didn't use the FTP in order to tranfert any type of files. Let's analyse them one by one to try and understand why they are present in our capture:



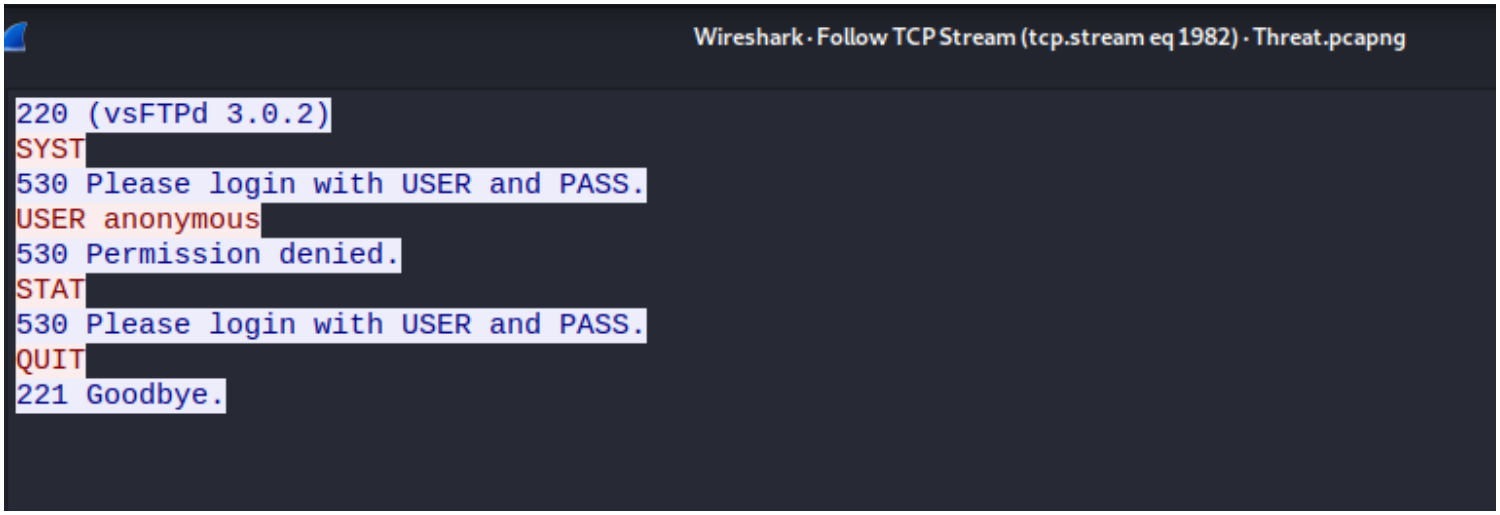
Well it seems that SSH was obviously so present and well used !! if we examin the info of the packets we can see the version used which is SSH-2.0-OpenSSH-7.4.
Let's follow the streams and see what's happening .



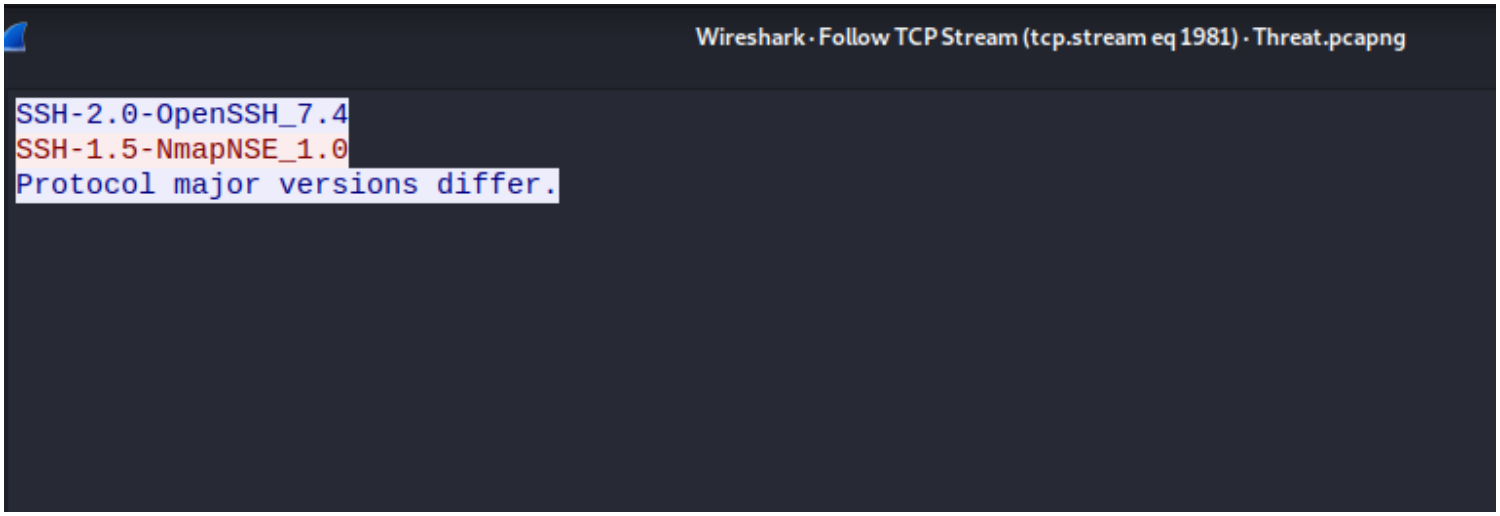
It seems that someone is scanning the network using Nmap. Let's verify our theory by analysing the ftp service.



this one looks like a brute force attack where someone is trying to guess the password and the username of the FTP service running on the 192.168.120.131 host.
Let's follow the FTP and try to verify the tool used :

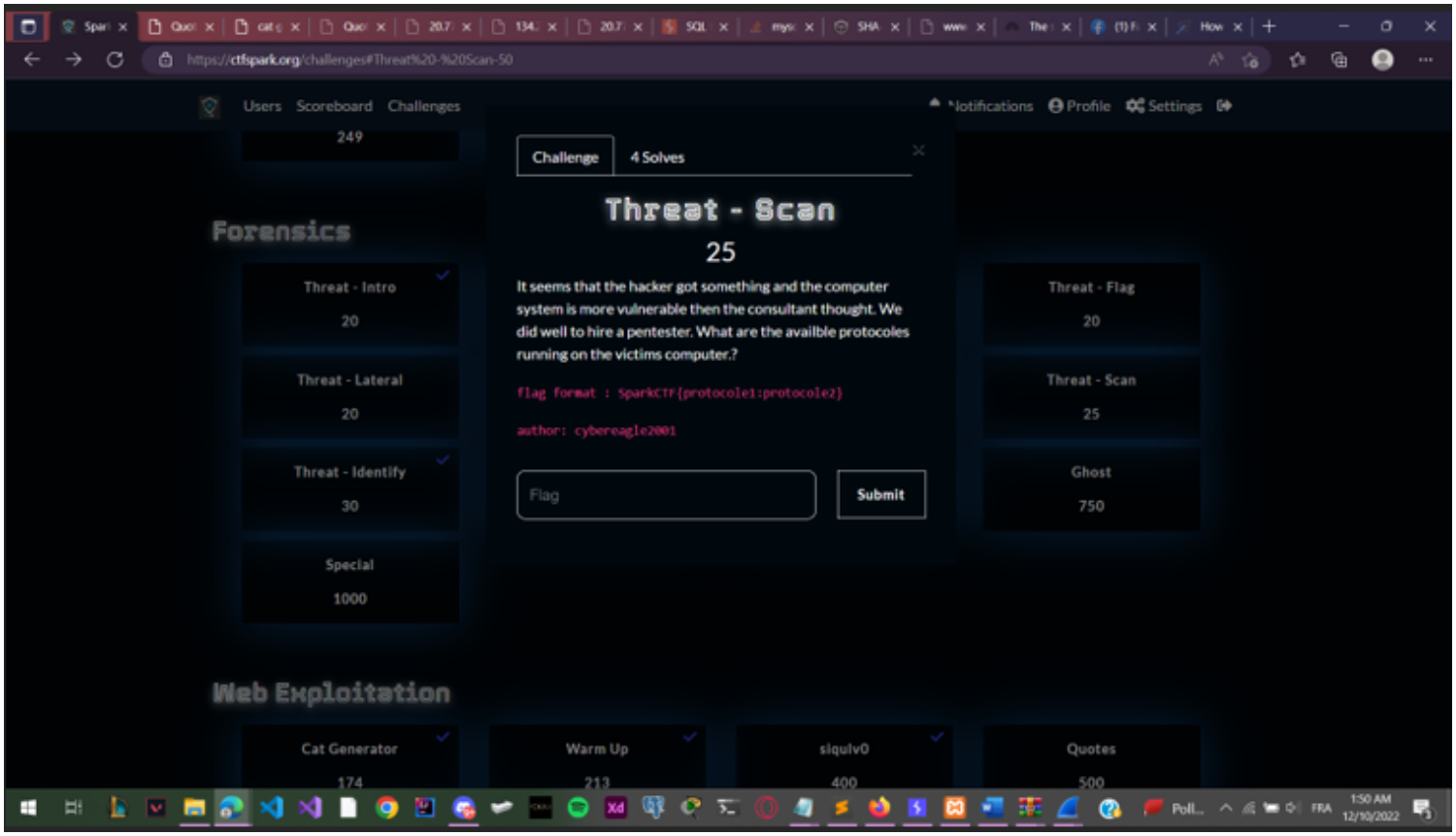


The vsFTPd 3.0.2 version of FTP is quite known to be vulnerable to a well known CVE.
CVE-2015-1419: Unspecified vulnerability in vsftpd 3.0.2 and earlier allows remote attackers to bypass access restrictions via unknown vectors, related to deny_file parsing.
This CVE is used by the NSE script engine of Nmap. We can obviously relate because we have already found some Nmaps fingerprints in SSH.



our flag is : SparkCTF{nmap}

Threat-scan results



This is A gift challenge if you have solved the Utility task.

flag: SparkCTF{ftp:ssh}

author: **cybereagle2001 (Oussama Ben Hadj Dahman)**