HTB- Sherlocks: Noxious – Level: Very Easy
Amit Persky

Table of Contents

Task 1: 2
Task 2: 3
Task 3: 4
Task 4: 5
Task 5: 5
Task 6: 6
Task 7: 7
Task 8: 7
Task 9: 8

Intro:

The IDS device alerted us to a possible rogue device in the internal Active Directory network. The Intrusion Detection System also indicated signs of LLMNR traffic, which is unusual. It is suspected that an LLMNR poisoning attack occurred. The LLMNR traffic was directed towards Forela-WKstn002, which has the IP address 172.17.79.136. A limited packet capture from the surrounding time is provided to you, our Network Forensics expert. Since this occurred in the Active Directory VLAN, it is suggested that we perform network threat hunting with the Active Directory attack vector in mind, specifically focusing on LLMNR poisoning.

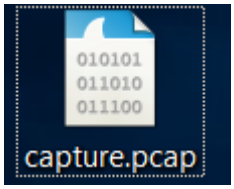
Tools that I used in this Sherlock:

1. Wireshark
2. Network Miner
3. Hashcat

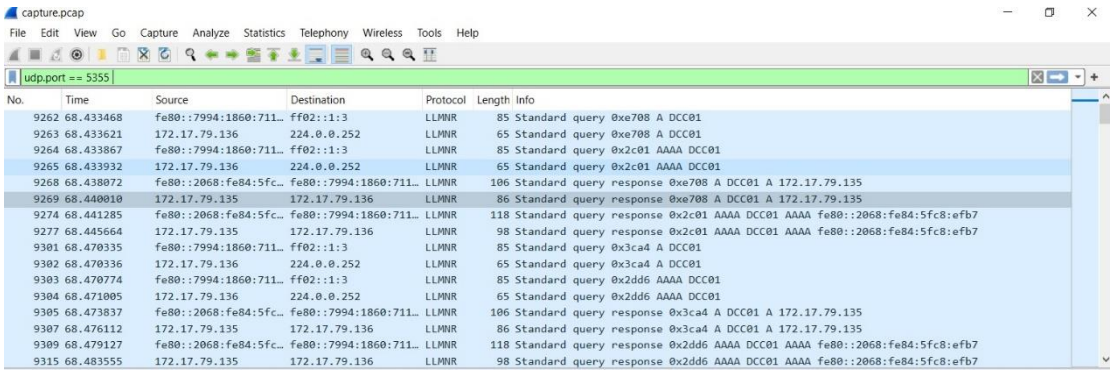
Task 1:

Its suspected by the security team that there was a rogue device in Forela's internal network running responder tool to perform an LLMNR Poisoning attack. Please find the malicious IP Address of the machine.

Ok so basically when we are talking about LLMNR we need to remember that this service running on UDP port 5355 and we know that 172.17.79.136 is a legitimate IP of the network. We got a pcap file that we are going to work with:



I decided to run Wireshark and to filter exactly to this LLMNR traffic:



If we look more closely we can see that the IP 172.17.79.136 made queries for the name DCC01 its a typo since it should be DC01 (most of the time)

Then we can see that the IP 172.17.79.135 is response back. Because of seeing this, we can say that the LLMNR Poisoning attack happened through these addresses, and we know that only one of this addresses is a legitimate IP, and one is not legitimate IP, and we already from the intro who is the legitimate IP.

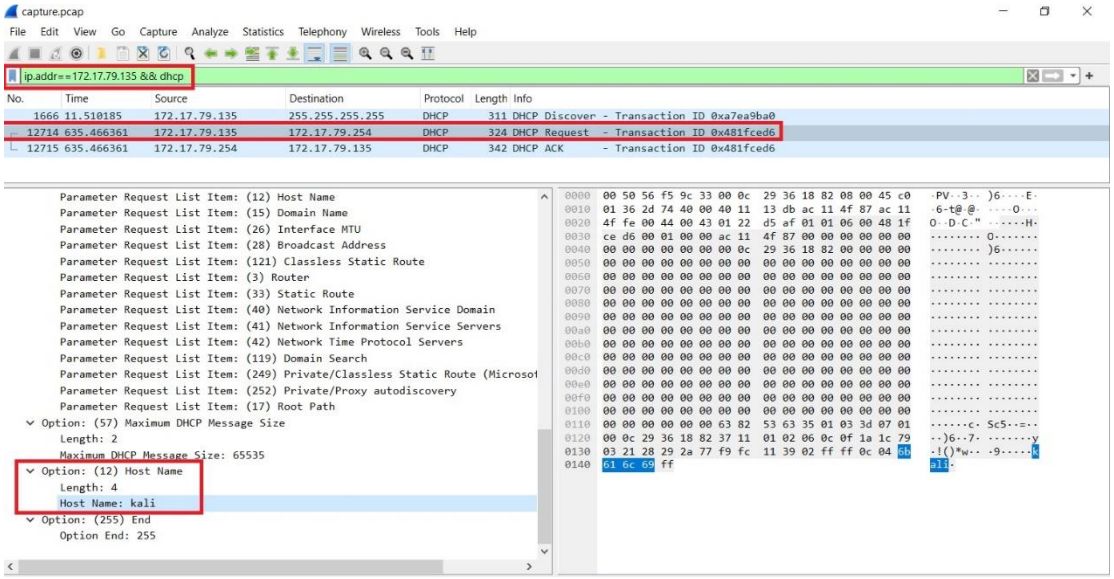
Answer:

172.17.79.135

Task 2:

What is the hostname of the rogue machine?

ok so for this question we need to filter 2 things on wireshark. The first one is the IP of the rogue machine (the answer from the last question) and the DHCP protocol we can to do that like that:



As we can see that I filterd 2 things, then I inspected the DHCP request, then I opend the Host Name tab and then you will can see the Host Name.

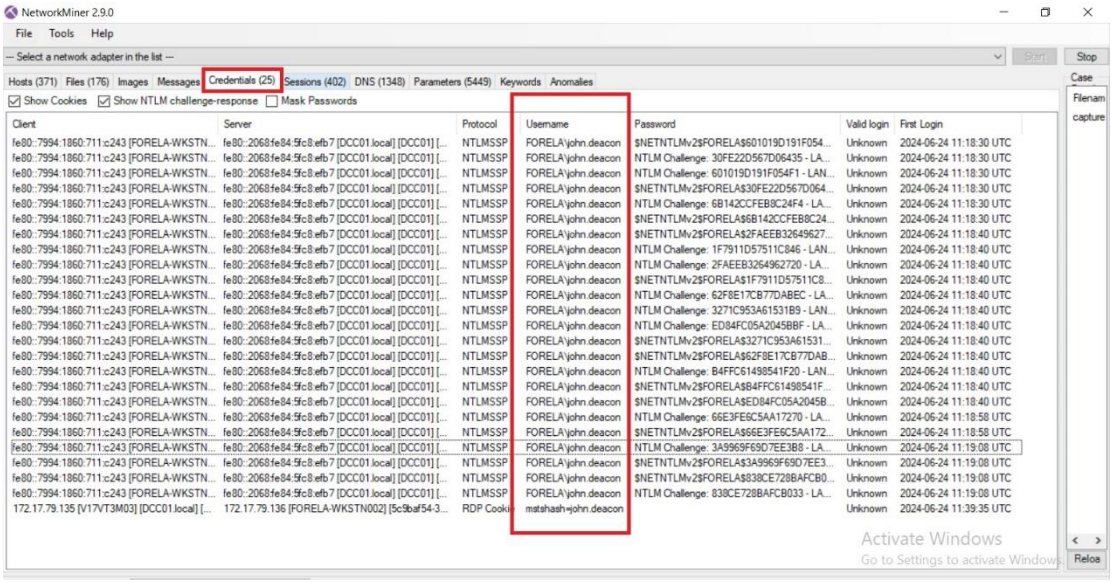
Answer:

Kali

Task 3:

Now we need to confirm whether the attacker captured the user's hash and it is crackable!! What is the username whose hash was captured?

Ok so for this question I opened my Network Miner and loaded our pcap.



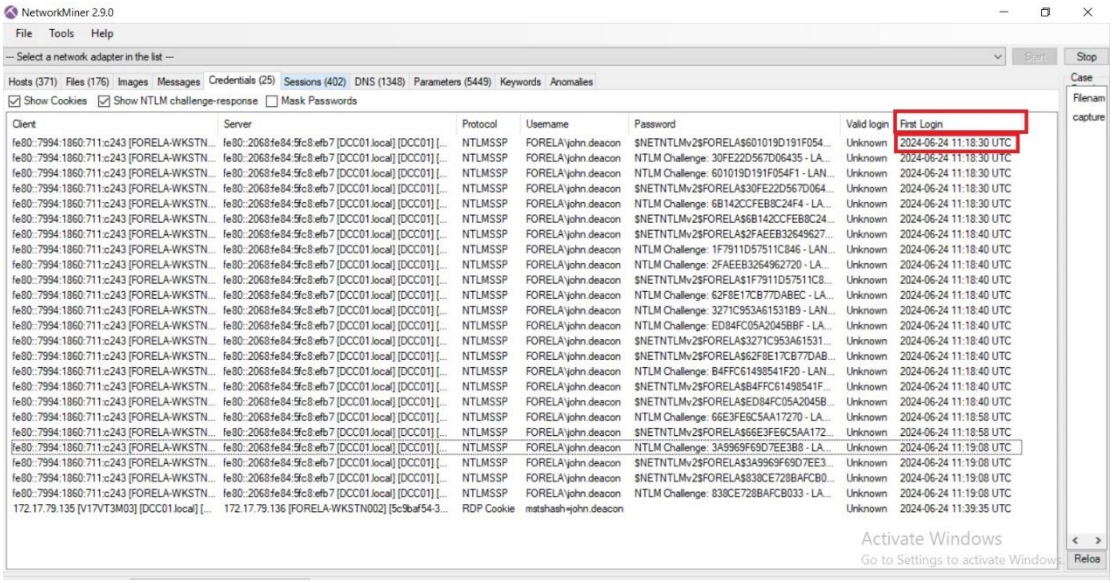
We can see the I moved to the tab Credentials. We can see one the Username Colum the username that we are looking for.

Answer:

john.deacon

Task 4:

Like the last question, we are at the same tab of Credentials. We are inside the tab and we need to navigate to First Login colum, and pressing on this in aim to sort. As we can see the first answer is the answer ae are looking for.



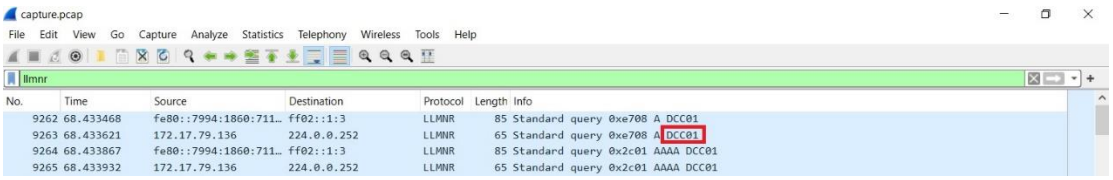
Answer:

2024-06-24 11:18:30

Task 5:

<u>What was the typo made by the victim when navigating to the file share that caused his credentials to be leaked?</u>

In the Wireshark capture, I looked at the LLMNR queries again being sent out by the victim's machine. These queries contained the hostname that the victim tried to resolve, which included the typo, as we saw the before and this is the answer.



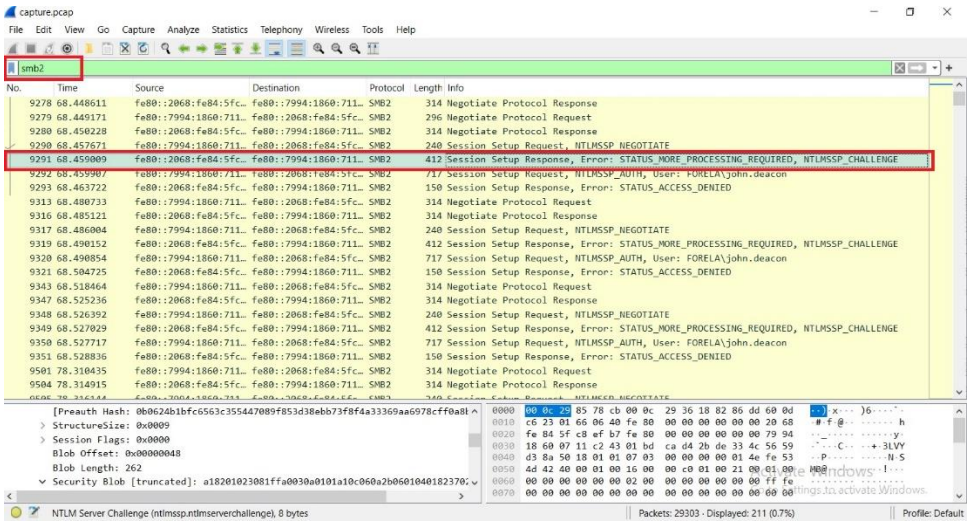
Answer:

DCC01

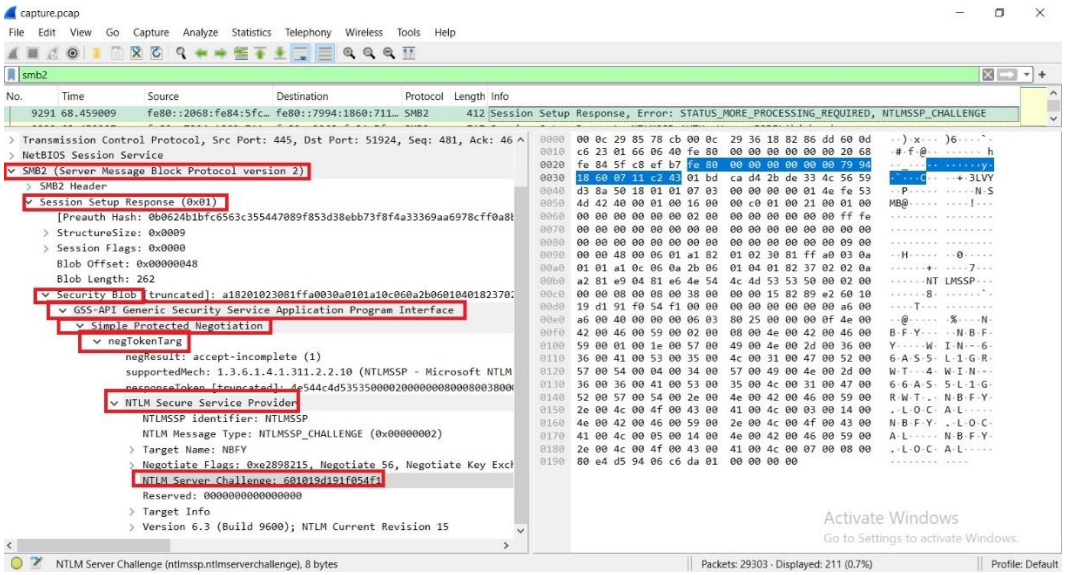
Task 6:

To get the actual credentials of the victim user we need to stitch together multiple values from the ntlm negotiation packets. What is the NTLM server challenge value?

Ok to answer this question I opened Wireshark again and I searched smb (smb2 on the wireshark) packets. As you can see I inspected the first NTLMSSP_CHALLENGE packet that I saw.



If we are going deep inside, we can see the NTLM server challenge value:



As we can observer, we found the NTLM server challenge value.

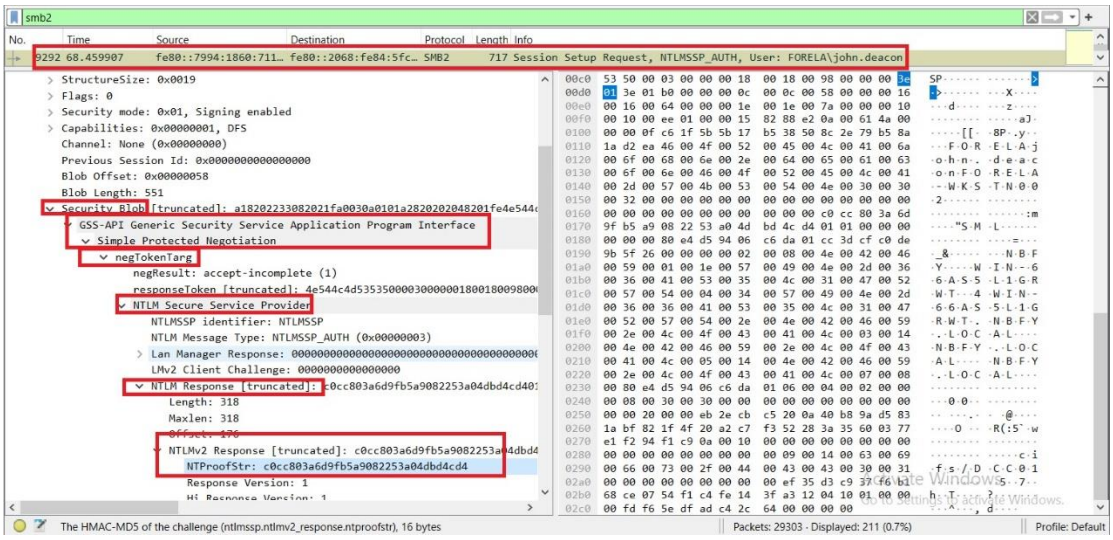
Answer:

601019d191f054f1

Task 7:

Now doing something similar find the NTProofStr value.

Ok for this question I scrolled 1 packet down to 9292 inspected deep inside at the same way, and I found the value:



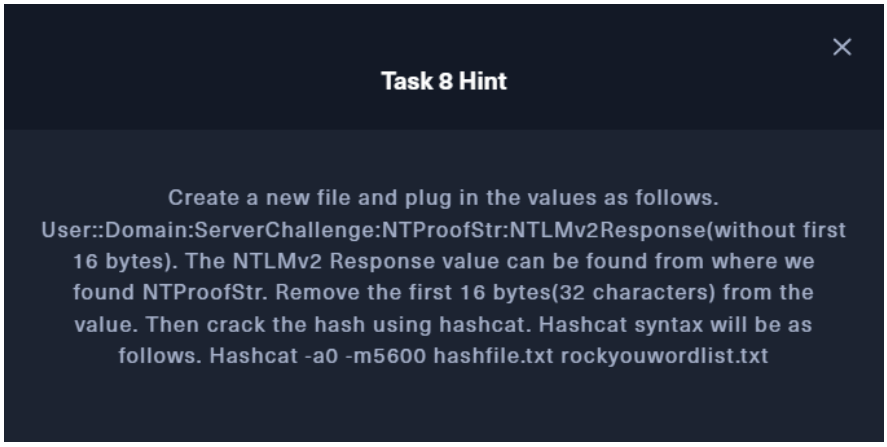
Answer:

c0cc803a6d9fb5a9082253a04dbd4cd4

Task 8:

To test the password complexity, try recovering the password from the information found from packet capture. This is a crucial step as this way we can find whether the attacker was able to crack this and how quickly.

Ok so for answering this question I checked the HTB hint to understand how to crack the password:



I checked the packets exactly how the hint tried to help and got this:

john.deacon::FORELA:601019d191f054f1:c0cc803a6d9fb5a9082253a04dbd4cd4:01010000000000080e4d59406c6da01cc3dcfc0de9b5f26000000000200008004e004200460059000100 1e00570049004e002d00360036004100530035004c00310047005200570054000400340057004 9004e002d00360036004100530035004c00310047005200570054002e004e004200460059002e004c004f00430041004c00030014004e004200460059002e004c004f00430041004c00050014004e004200460059002e004c004f00430041004c000700080080e4d59406c6da01060004000200000008003000300000000000000000000000000200000eb2ecbc5200a40b89ad5831abf821f4f20a2c7f352283a35600377e1f294f1c90a0010000000000000000000000000000000000000900140063006900660073002f00440043004300300031000000000000000000

now all I have to do is to hashcat that:

```
┌──(kali㉿kali)-[~/Desktop]
└─$ hashcat -a0 -m5600 crackme rockyou.txt
hashcat (v6.2.6) starting
```

```
┌──(kali㉿kali)-[~/Desktop]
└─$ hashcat -a0 -m5600 crackme rockyou.txt --show > youaregood.txt

┌──(kali㉿kali)-[~/Desktop]
└─$ cat youaregood.txt
JOHN.DEACON::FORELA:601019d191f054f1:c0cc803a6d9fb5a9082253a04dbd4cd4:01010000000000080e4d59406c6da01cc3dcfc0de9b5f260000000
0020008004e004200460059000100 1e00570049004e002d00360036004100530035004c00310047005200570054000400340057004 9004e002d00360036004
100530035004c00310047005200570054002e004e004200460059002e004c004f00430041004
c00050014004e004200460059002e004c004f00430041004c000700080080e4d59406c6da01060004000200000008003000300000000000000000000000000
200000eb2ecbc5200a40b89ad5831abf821f4f20a2c7f352283a35600377e1f294f1c90a0010000000000000000000000000000000000000900140063006900
0660073002f00440043004300300031000000000000000000:NotMyPassword0k?
```

Answer:

NotMyPassword0k?

Task 9:

Just to get more context surrounding the incident, what is the actual file share that the victim was trying to navigate to?

Ok so for answering this question I looked in the hint:

Task 9 Hint

Filter for SMB traffic. In Wireshark, it is "smb2". Then keep scrolling until you see a tree connect/disconnect of a NON-DEFAULT File share name. This will make sense if you keep in mind the typo the victim made in Question 5.

I understood that we are looking for smb (smb2 on Wireshark) packet with connect/disconnect with some details that we already know. I started scrolling down after the filtering and at first look I saw that in some time:

10189	115.465183	172.17.79.136	172.17.79.4	SMB2	152	Tree Connect Request Tree: \\DC01\IPC$

But, we know that the share name IPC$ is a **default** share in Windows environments so its not the answer. But know we can imagine how the real answer need to look like with unique path. I scrolled down and say this:

10214	115.957127	172.17.79.136	172.17.79.4	SMB2	174	Tree Connect Request Tree: \\DC01\DC-Confidential

Unlike the previous IPC$ share, which is a default administrative share, DC-Confidential appears to be a custom, non-default file share. It might be a share created specifically for sensitive or confidential data on the domain controller.

So this is the answer that we looked for:

Answer:

\\DC01\DC-Confidential