

# Wireshark

## hash ntlmv2(smb2)

SMB es un protocolo cuyo principal cometido es el compartir ficheros en la red con otros equipos.

[illegible]

Formato de hash nltm:

```
[user Name]::[domain name]:[NTLM server challenge]:[NtproofStr]:[Rest  
of NTLMv2 Response]
```

## Paquete donde se localiza el NTLM server challenge

filtrar por smb2

|      |                                                                                       |
|------|---------------------------------------------------------------------------------------|
| SMB2 | 220 Session Setup Request, NTLMSSP_NEGOTIATE                                          |
| SMB2 | 359 Session Setup Response, Error: STATUS_MORE_PROCESSING_REQUIRED, NTLMSSP_CHALLENGE |
| SMB2 | 610 Session Setup Request, NTLMSSP_AUTH_Hmac-SHA1-NTLMSSP_inauth                      |

## Paquete donde se localiza el resto de elementos

en este paquete se realiza la autenticación

|    |           |              |              |      |     |                                                                |
|----|-----------|--------------|--------------|------|-----|----------------------------------------------------------------|
| 13 | 14.557099 | 192.168.1.5  | 192.168.1.22 | SMB2 | 359 | Session Setup Response, Error: STATUS_MORE_PROCESSING_REQUIRED |
| 14 | 14.557284 | 192.168.1.22 | 192.168.1.5  | SMB2 | 643 | Session Setup Request, NTLMSSP_AUTH, User: CLITM03\joseph      |
| 15 | 14.561743 | 192.168.1.5  | 192.168.1.22 | SMB2 | 159 | Session Setup Response                                         |

## User name and Domain name

## NTLM server Challenge

## NTProofStr

## Resto del hash

Copiar valor desde wireshark haciendo clic derecho en el campo del hash, después borrar la parte de NTPProofStr

```

13 14.557099      192.168.1.5      192.168.1.22      SMB2      359 Session Setup Response, Error: STATUS_MORE_PROCESSING_REQUIRED, NTLMSSP_CHA
14 14.557284      192.168.1.22      192.168.1.5      SMB2      643 Session Setup Request, NTLMSSP_AUTH, User: CLITM003\joseph
15 14.561743      192.168.1.5      192.168.1.22      SMB2      159 Session Setup Response
16 14.561903      192.168.1.22      192.168.1.5      SMB2      166 Tree Connect Request Tree: \\192.168.1.5\IPC$
17 14.562299      192.168.1.5      192.168.1.22      SMB2      138 Tree Connect Response
18 14.562368      192.168.1.22      192.168.1.5      SMB2      190 Create Request File: srvsvc
19 14.562690      192.168.1.5      192.168.1.22      SMB2      210 Create Response File: srvsvc
20 14.562752      192.168.1.22      192.168.1.5      SMB2      162 GetInfo Request FILE_INFO/SMB2_FILE_STANDARD_INFO File: srvsvc
21 14.563029      192.168.1.5      192.168.1.22      SMB2      154 GetInfo Response
22 14.563179      192.168.1.22      192.168.1.5      DCERPC    330 Bind: call_id: 2, Fragment: Single, 3 context items: SRVSVC V3.0 (32bit NDR
23 14.563482      192.168.1.5      192.168.1.22      SMB2      138 Write Response

responseToken: 4e544c4d535350000300000018001800840000001c011c019c0000001000100058000000...
  NTLM Secure Service Provider
    NTLMSSP identifier: NTLMSSP
    NTLM Message Type: NTLMSSP_AUTH (0x00000003)
  Lan Manager Response: 0000000000000000000000000000000000000000000000000000000000000000
    LmV2 Client Challenge: 0000000000000000000000000000000000000000000000000000000000000000
  NTLM Response: 1ed65b74625b531660b02f6abb621f8801010000000000001108aef86cfd501989ff10c
    Length: 284
    MaxLen: 284
    Offset: 156
  NTLMv2 Response: 1ed65b74625b531660b02f6abb621f8801010000000000001108aef86cfd501989
    NTProofStr: 1ed05b74625b531660b02f6abb621f88
    Response Version: 1

```

```
010100000000000001108eaeef86cfd501989ff10daee222e1000000000200160053004f
004d00420053004800410052004500300032000100160053004f004d00420053004800
410052004500300032000400160053004f004d00420053004800410052004500300032
000300160053004f004d0042005300480041005200450030003200070008001108eaeef
86cfd501060004000200000000800300030000000000000000100000000200000d7f171
06c90c6679fdca1774bbbb9c481f29fc711a43970a2d500d86020b97ca0a0010000000
00000000000000000000000000000000900200063006900660073002f003100390032002e
003100360038002e0031002e003500000000000000000000000000
```

## crackear hash

```
echo 'joseph::CLITMW03:9c6ee2011a330fef:1ed65b74625b531666b02f6abb621f
88:010100000000000001108eaf86cfd501989ff10daee222e10000000002001600530
04f004d00420053004800410052004500300032000100160053004f004d00420053004
800410052004500300032000400160053004f004d00420053004800410052004500300
032000300160053004f004d0042005300480041005200450030003200070008001108e
af86cfd50106000400020000000080030003000000000000000100000000200000d7f
17106c90c6679fdca1774bbbb9c481f29fc711a43970a2d500d86020b97ca0a0010000
00000000000000000000000000000000900200063006900660073002f0031003900320
02e003100360038002e0031002e003500000000000000000000' > hash
```

```
john --wordlist=/usr/share/wordlists/rockyou.txt hash
```

## Telnet

Telnet es un protocolo por el que se puede realizar una conexión de forma remota entre dos equipos, es un protocolo inseguro ya que estas conexiones no van cifradas como en ssh.

hacer clic derecho en un paquete telnet y darle a follow>tcp, nos mostrara el intercambio de datos entre cliente servidor y podremos ver las credenciales.

|    |          |             |             |        |                                                                           |
|----|----------|-------------|-------------|--------|---------------------------------------------------------------------------|
| 3  | 0.001741 | 192.168.0.2 | 192.168.0.1 | TCP    | 66 1254 → 23 [ACK] Seq=1 Ack=1 Win=32120 Len=0 TSval=1444389 TSecr=346979 |
| 4  | 0.013173 | 192.168.0.2 | 192.168.0.1 | TELNET | 93 Telnet Data ..                                                         |
| 5  | 0.150283 | 192.168.0.1 | 192.168.0.2 | TELNET | 69 Telnet Data ..                                                         |
| 6  | 0.150351 | 192.168.0.2 | 192.168.0.1 | TCP    | 66 1254 → 23 [ACK]                                                        |
| 7  | 0.150528 | 192.168.0.2 | 192.168.0.1 | TELNET | 69 Telnet Data ..                                                         |
| 8  | 0.151908 | 192.168.0.1 | 192.168.0.2 | TCP    | 66 23 → 1254 [ACK]                                                        |
| 9  | 0.153602 | 192.168.0.1 | 192.168.0.2 | TELNET | 91 Telnet Data ..                                                         |
| 10 | 0.153816 | 192.168.0.2 | 192.168.0.1 | TELNET | 130 Telnet Data ..                                                        |
| 11 | 0.154904 | 192.168.0.1 | 192.168.0.2 | TCP    | 66 23 → 1254 [ACK]                                                        |
| 12 | 0.155418 | 192.168.0.1 | 192.168.0.2 | TELNET | 84 Telnet Data ..                                                         |
| 13 | 0.155496 | 192.168.0.2 | 192.168.0.1 | TELNET | 75 Telnet Data ..                                                         |
| 14 | 0.156474 | 192.168.0.1 | 192.168.0.2 | TCP    | 66 23 → 1254 [ACK]                                                        |
| 15 | 0.158758 | 192.168.0.1 | 192.168.0.2 | TELNET | 90 Telnet Data ..                                                         |
| 16 | 0.159498 | 192.168.0.2 | 192.168.0.1 | TELNET | 151 Telnet Data ..                                                        |
| 17 | 0.160654 | 192.168.0.1 | 192.168.0.2 | TCP    | 66 23 → 1254 [ACK]                                                        |
| 18 | 0.181170 | 192.168.0.1 | 192.168.0.2 | TELNET | 69 Telnet Data ..                                                         |
| 19 | 0.181250 | 192.168.0.2 | 192.168.0.1 | TELNET | 69 Telnet Data ..                                                         |
| 20 | 0.182445 | 192.168.0.1 | 192.168.0.2 | TCP    | 66 23 → 1254 [ACK]                                                        |
| 21 | 0.196092 | 192.168.0.1 | 192.168.0.2 | TELNET | 78 Telnet Data ..                                                         |
| 22 | 0.196205 | 192.168.0.2 | 192.168.0.1 | TELNET | 72 Telnet Data ..                                                         |
| 23 | 0.197390 | 192.168.0.1 | 192.168.0.2 | TCP    | 66 23 → 1254 [ACK]                                                        |

Mark/Unmark Packet Ctrl+M  
Ignore/Unignore Packet Ctrl+D  
Set/Unset Time Reference Ctrl+T  
Time Shift... Ctrl+Shift+T  
Packet Comments  
Edit Resolved Name  
Apply as Filter  
Prepare as Filter  
Conversation Filter  
Colorize Conversation  
SCTP  
Follow  
Copy  
Protocol Preferences


Frame 4: 93 bytes on wire (744 bits), 93 bytes captured (744 bits)

## FTP

Instalar librería descargándola del siguiente enlace:

<https://pypi.org/project/pyftplib/#files>

### Distribución fuente

 [pyftplib-1.5.9.tar.gz](#) (204.8 kB [ver hashes](#))  
Uploaded 25 oct 2023 [Source](#)

```
# en Downloads, descomprimos
tar -xvf pyftplib-1.5.9.tar.gz

# entramos en el directorio generado
# instalamos
sudo python setup.py install

# levantamos servidor ftp
sudo python -m pyftplib -u user -P 123 -p 21

-u --> indicamos usuario del servidor ftp
-P --> indicamos una contraseña
-p --> indicamos el puerto
```

iniciamos wireshark para capturar los paquetes de conexión con el servidor ftp.

Desde ubuntu server nos conectamos al server ftp.

```
ftp IPKALI
usuario:user
password:123
```

En wireshark filtramos por ftp

| No. | Time          | Source     | Destination | Protocol | Length | Info                                      |
|-----|---------------|------------|-------------|----------|--------|-------------------------------------------|
| 270 | 320.839616470 | 10.0.10.5  | 10.0.10.20  | FTP      | 94     | Response: 220 pyftplib 1.5.9 ready.       |
| 272 | 323.238480301 | 10.0.10.20 | 10.0.10.5   | FTP      | 77     | Request: USER user                        |
| 274 | 323.239281271 | 10.0.10.5  | 10.0.10.20  | FTP      | 99     | Response: 331 Username ok, send password. |
| 278 | 330.372984818 | 10.0.10.20 | 10.0.10.5   | FTP      | 76     | Request: PASS 123                         |
| 279 | 330.373957536 | 10.0.10.5  | 10.0.10.20  | FTP      | 89     | Response: 230 Login successful.           |
| 281 | 330.376770714 | 10.0.10.20 | 10.0.10.5   | FTP      | 72     | Request: SYST                             |
| 282 | 330.377474486 | 10.0.10.5  | 10.0.10.20  | FTP      | 85     | Response: 215 UNIX Type: L8               |

## HTTP

Con wireshark sniffando accedemos al Login de la siguiente web.

<http://testphp.vulnweb.com/login.php>

introducimos unas credenciales cualquiera.

En wireshark filtramos por http.

Clic derecho en un paquete http y a continuación follow>tcp

```
Referer: http://testphp.vulnweb.com/login.php
Upgrade-Insecure-Requests: 1
uname=test&pass=test HTTP/1.1 200 OK
Server: nginx/1.19.0
Date: Sat, 27 Apr 2024 15:34:25 GMT
```

## ICMP

Realizamos un ping a nuestra maquina o desde nuestra maquina y filtramos por icmp

| No. | Time        | Source       | Destination  | Protocol | Length | Info                                                               |
|-----|-------------|--------------|--------------|----------|--------|--------------------------------------------------------------------|
| 81  | 7.200142546 | 192.168.1.40 | 192.168.1.36 | ICMP     | 74     | Echo (ping) request id=0x0001, seq=9/2304, ttl=128 (reply in 82)   |
| 82  | 7.200168808 | 192.168.1.36 | 192.168.1.40 | ICMP     | 74     | Echo (ping) reply id=0x0001, seq=9/2304, ttl=64 (request in 81)    |
| 100 | 8.108723355 | 192.168.1.40 | 192.168.1.36 | ICMP     | 74     | Echo (ping) request id=0x0001, seq=10/2560, ttl=128 (reply in 101) |
| 101 | 8.108756916 | 192.168.1.36 | 192.168.1.40 | ICMP     | 74     | Echo (ping) reply id=0x0001, seq=10/2560, ttl=64 (request in 100)  |
| 112 | 9.020431822 | 192.168.1.40 | 192.168.1.36 | ICMP     | 74     | Echo (ping) request id=0x0001, seq=11/2816, ttl=128 (reply in 113) |
| 113 | 9.020458705 | 192.168.1.36 | 192.168.1.40 | ICMP     | 74     | Echo (ping) reply id=0x0001, seq=11/2816, ttl=64 (request in 112)  |
| 114 | 9.931879848 | 192.168.1.40 | 192.168.1.36 | ICMP     | 74     | Echo (ping) request id=0x0001, seq=12/3072, ttl=128 (reply in 115) |
| 115 | 9.931905777 | 192.168.1.36 | 192.168.1.40 | ICMP     | 74     | Echo (ping) reply id=0x0001, seq=12/3072, ttl=64 (request in 114)  |