Lab 4: Metasploit (Scanning/Enumeration)

Aim:

The aim of this lab is to provide a foundation in enumerating Windows instances on a network in which usernames and infomation on groups, shares, and services of Windows computer are retrieved. This should not be confused with network mapping which only retrieves information about which computers/servers are connected to a specific network or what operating system runs on them.

Quick tool introduction:

Throwback to last week's lab: Metasploit framework is an open source penetration tool used for developing and executing exploit code against a remote target machine. The framework has the world's largest database of public and tested exploits. In simple words, Metasploit can be used to test the vulnerability of computer systems in order to protect them and on the other hand it can also be used to break into remote systems. It's a powerful tool used for penetration testing.

Time to Complete:

2-3 hours

Activities:

- **Complete Lab 4:** Windows scanning and enumeration using Metasploit.

Learning activities:

At the end of this lab, you should understand:

- How to use ***auxiliary modules*** in Metasploit with particular focus on *scanner auxiliary modules*

References:

- Offensive Security Training, Certifications and Services – Metasploit – Auxiliary modules, available at: https://www.offensive-security.com/metasploit-unleashed/auxiliary-module-reference
- Agarwal, M., & Singh, A. (2013). Metasploit penetration testing cookbook. Packt Publishing Ltd.

1 Lab Overview

Aim: To provide a foundation in enumerating Windows instances on a network.

	The demo of this lab is at: http://youtu.be/5JdxGRt7QN0

With your group allocation, setup your host with the IP addresses using **Allocation C**, as defined in: **http://asecuritysite.com/csn10107/prep**

Table 1: Your challenges summary checklist

Challenge	Description	How will I do this?	Completed?
1	You should be able to know how to enumerate any SMB shares available on a remote system using Metasploit	Use smb_enumshares *auxiliary modules*	
2	You should be able to know how to attempt to login via SMB across a provided IP address using Metasploit	Use smb_login *auxiliary modules*	
3	You should be able to know how to brute-forces SID lookups on a range of targets using Metasploit	Use smb_lookupsid *auxiliary modules*	
4	You should be able to know how to determines the version of the SMB service that is running on remote hosts using Metasploit	Use smb_version *auxiliary modules*	
5	You should be able to know how to do port scanning including: TCP, TCP SYN and XMAS using Metasploit	Use portscan *auxiliary modules in Metasploit*	
6	You should be able to know how to scan a subnet and fingerprint any Telnet servers that are running on a remote host using Metasploit	Use telnet *auxiliary modules in Metasploit*	

SMB Shares

Microsoft Windows uses the Server Message Block (SMB) Protocol, one version of which was also known as Common Internet File System (CIFS), operates as an application-layer network protocol mainly used for providing shared access to files, printers, and serial ports and miscellaneous communications between nodes on a network.

In today's lab, we use ***auxiliary modules*** in Metasploit. The Metasploit Framework includes hundreds of auxiliary modules that perform scanning, fuzzing, sniffing, and much more. Although these modules will not give you a shell, they are extremely valuable when conducting a penetration test. Generally, they are grouped in three categories: Admin, Scanner and Server. Following is a reference to the auxiliary modules from each group in Metasploit.

***Admin* auxiliary module:**
Admin HTTP Modules Admin MSSQL Modules Admin MySQL Modules

Admin Postgres Modules Admin VMWare Modules

***Scanner* auxiliary module:**

DCERPC	Discovery	FTP	HTTP	IMAP	MSSQL
MySQL	NetBIOS	POP3	SMB	SMTP	SNMP
SSH	Telnet	TFTP	VMWare	VNC	

***Server* auxiliary module:**
Server Capture Modules

You should complete in a group of two, and only use one of your folders. One of the group will be the SCANNER, and the other will be the TARGET. For your target (Windows 2008, Windows 2003 or Windows 7). Determine the details of the target:

User name	[USER]:
Password	[PASSWORD]:
Target IP address	[W.X.Y.Z]:

1 Microsoft Windows uses the Server Message Block (SMB) Protocol to share files and folders over a network. Setup your Kali and [TARGET] instance to be on the same network. Now one person will setup a **new share** on the target machine, but do not say what the name of it is to the person who will scan it.

2 Now scan the target computer for SMB shares with (also run Wireshark on your Kali instance and capture your network traffic). On your Kali DMZ:

```
msfconsole
msf > use auxiliary/scanner/smb/smb_enumshares
msf auxiliary(smb_enumshares) > set RHOSTS [W.X.Y.Z]
RHOSTS => [W.X.Y.Z]
msf auxiliary(smb_enumshares) > set SMBUser [USER]
SMBUser => Administrator
msf auxiliary(smb_enumshares) > set SMBPass [PASSWORD]
SMBPass => [PASSWORD]
msf auxiliary(smb_enumshares) > run
```

As would be expected, smb_enumshares module enumerates any SMB shares that are available on a remote system.

What is the name of the folder they created?

From the Wireshark trace, which TCP port SMB uses to connect?

3 Now get your lab partner to create a **new user** on the target instance. Then scan the Windows computer for SMB users with.

```
msf > use auxiliary/scanner/smb/smb_enumusers
msf auxiliary(smb_enumusers) > set RHOSTS [W.X.Y.Z]
RHOSTS => [W.X.Y.Z]
msf auxiliary(smb_enumusers) > set SMBUser [USER]
SMBUser => Administrator
msf auxiliary(smb_enumusers) > set SMBPass [PASSWORD]
SMBPass => [PASSWORD]
msf auxiliary(smb_enumusers) > run
```

What was the user name they created?

Is there a password lock-out set?

Is there a minimum password length set?

SMB Login

4 Metasploit's smb_login module will attempt to login via SMB across a provided IP address (es). If you have a database plugin loaded, successful logins will be stored in it for future reference and usage.

```
msf > use auxiliary/scanner/smb/smb_login
msf auxiliary(smb_login) > set RHOSTS [W.X.Y.Z]
RHOSTS => [W.X.Y.Z]
msf auxiliary(smb_login) > set SMBUser [USER]
SMBUser => Administrator
msf auxiliary(smb_login) > set SMBPass [PASSWORD]
SMBPass => napier
msf auxiliary(smb_login) > run
```

5 Using the **show options** command in Metasploit, you can clearly see that this module has many more options that other auxiliary modules and is quite versatile. The smb_login module can also be passed a username and password list in order to attempt to brute-force login attempts across a range of machines.

6 Create a **User name file (users.txt)** and a **Password file (passwords.txt)** with all the following using "nano" command in Linux:

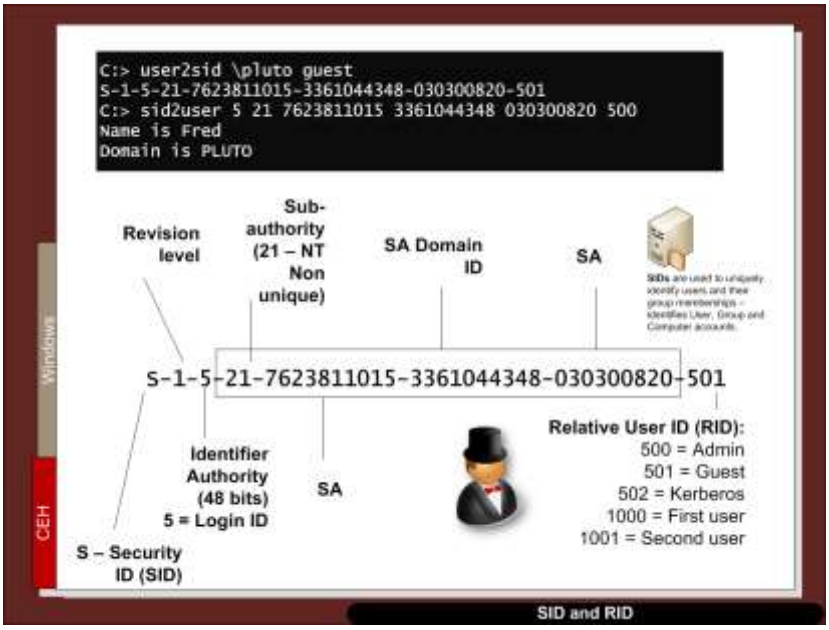
users.txt: Administrator, napier, root, Guest, test, default, [USER]
passwords.txt: napier, test, guest, password, changeme, [PASSWORD]

msf > **use auxiliary/scanner/smb/smb_login**
msf auxiliary(smb_login) > **show options**
Check if you can see PASS_FILE and USER_FILE
msf auxiliary(smb_login) > **set PASS_FILE /root/passwords.txt**
set USER_FILE /root/users.txt
msf auxiliary(smb_login) > **set USER_FILE /root/users.txt**
SMBPass => napier
msf auxiliary(smb_login) > **run**

Which user names and passwords did it detect?

Each Windows computer in a domain (or workgroup) has a unique identifier. For example:

- 5-1-5-21-7623811015-3361044348-030300820-501
- 1 Revision level
- 5- Identified authority (5 - SECURITY_NT_AUTHORITY).
- 21-7623811015-3361044348-030300820 - Unique domain or local computer ID.
- 501 – RID (Relative ID) defines the unique ID for the given SID.



SMB Lookup

7 The smb_lookupsid module brute-forces SID lookups on a range of targets to determine what local users exist the system. Knowing what users exist on a system can greatly speed up any further brute-force log-on attempts later on.

```
msf > use auxiliary/scanner/smb/smb_lookupsid
msf auxiliary(smb_lookupsid) > show options
msf auxiliary(smb_lookupsid) > set RHOSTS [W.X.Y.Z]
RHOSTS => [W.X.Y.Z]
msf auxiliary(smb_lookupsid) > set SMBUser Administrator
SMBUser => Administrator
msf auxiliary(smb_lookupsid) > set SMBPass [PASSWORD]
SMBPass => [PASSWORD]
msf auxiliary(smb_lookupsid) > run
```

What is the SID of the Windows 8 computer?

Ask another group for their SID. For the Administrator account, is the SID different from yours?

What does an RID of 500 identify?

What is special about the RID values of 1,000 and above?

SMB Version

8 The smb_version scanner connects to each workstation in a given range of hosts and determines the version of the SMB service that is running (you can use "–" in order to identify a range of IP address e.g. 192.168.1.150-165):

```
msf > use auxiliary/scanner/smb/smb_version
msf auxiliary(smb_lookupsid) > show options
msf auxiliary(smb_lookupsid) > set RHOSTS [W.X.Y.Z]
RHOSTS => [W.X.Y.Z]
msf auxiliary(smb_version) > set SMBUser Administrator
SMBUser => Administrator
msf auxiliary(smb_version) > set SMBPass [PASSWORD]
SMBPass => [PASSWORD]
msf auxiliary(smb_version) > run
```

What information gained from the scan?

- *net share* manages shared resources. Used without parameters, **net share** displays information about all of the resources that are shared on the local computer.

- *net view* displays a list of domains, computers, or resources that are being shared by the specified computer.

Syntax: net view [\ \ComputerName] [/domain[:DomainName]

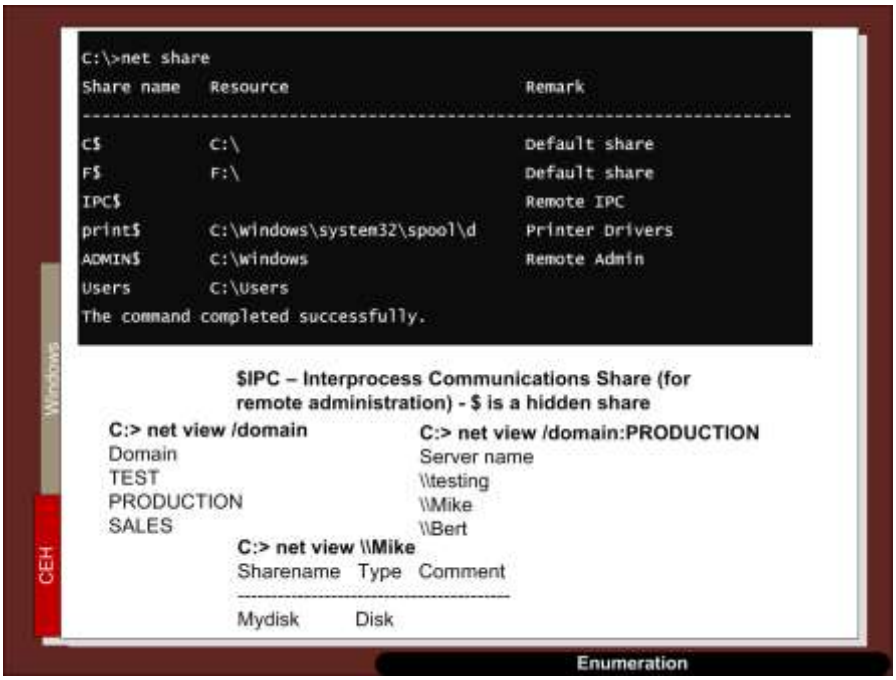
9 On the target instance, use the following commands and observe the output:

C:> **net share**

Output:

C:> net view \\[W.X.Y.Z]

Output:



10 Get your lab partner to **add a file** on the target host (and put in a secret message) to the **shared folder**, and mount the folder with (replace **admin_share** with the name of your share). On Kali DMZ type:

smbclient //[W.X.Y.Z]/admin_share -U Administrator

What is the name of the file produced?

On the Windows 8 instance, which command would you use to show your network shares?

From the trace, can you see the signs of the user accessing the file?

NOW SWAP YOUR ROLE WITH YOUR LAB PARTNER, and use their folder for their instances.

11 Now create an SMB share.

Using Metasploit, determine the following:

Shares:

Users:

SID:

RID values (and their mapping to Users):

Lock-out tries:

Minimum password size:

Operating system (with Service Pack):

Scanning

12 We can use Metaploit to perform a scan. First we will search for the portscan module:

msf > **search portscan**

What do you observe from the run:

13 **Start Wireshark.** We can now perform a TCP port scan using Metasploit's auxiliary module:

msf > **use auxiliary/scanner/portscan/tcp**
msf auxiliary(smb_lookupsid) > **show options**

msf auxiliary(tcp) > **set RHOSTS [W.X.Y.Z]**
RHOSTS => [W.X.Y.Z]
msf auxiliary(tcp) > **run**

Which ports are open on Windows 8 and Window 2003:

From your Wireshark trace (using the filter in the form ip.addr==1.2.3.4), identify how Metasploit identifies an open port and a closed port:

14 We can now perform a SYN port scan and capture the traffic with Wireshark:

msf > **use auxiliary/scanner/portscan/syn**
msf auxiliary(smb_lookupsid) > **show options**
msf auxiliary(tcp) > **set RHOSTS [W.X.Y.Z]**
RHOSTS => [W.X.Y.Z]
msf auxiliary(tcp) > **run**

What is the main difference between the TCP SYN scan and the TCP port scan?

15 Run the sc_Uan again with a Xmas tree scan. What TCP flags are set? (hint: use: msf > use auxiliary/scanner/portscan/xmas):

If you get GATEWAY_PROBE_HOST error, type:

msf auxiliary(xmas) > **show advanced**
msf auxiliary(xmas) > **set GATEWAY_PROBE_HOST** ***your_Kali_IP***

16 Now we will discover the NetBios name of both the Windows 2003 and target instances (The "nbname" auxiliary module scans a range of hosts and determines their hostnames via NetBIOS.)?

use auxiliary/scanner/netbios/nbname

What are the NETBIOS names on your network (scan the range of your Windows 2003 and target instances)?

Telnet scanner

From a network security perspective, one would hope that Telnet would no longer be in use as everything, including credentials is passed in the clear but the fact is, you will still frequently encounter systems running Telnet, particularly on legacy systems.

The telnet_version auxiliary module will scan a subnet and fingerprint any Telnet servers that are running. We just need to pass a range of IPs to the module, set our THREADS value, and let it scan.

17 Now using the following :

msf > use auxiliary/scanner/telnet/telnet_version

msf auxiliary(telnet_version) > **show options**

msf auxiliary(telnet_version) > **set RHOSTS [W.X.Y.Z]**

msf auxiliary(telnet_version) > **run**

18 The telnet_login module will take a list a provided set of credentials and a range of IP addresses and attempt to login to any Telnet servers it encounters. Now using the following :

use auxiliary/scanner/telnet/telnet_login

Discover the username and passwords that allows for a Telnet login on the Windows 2003 instance (hint: use users.txt and passwrods.txt that you created recently. For this you need to set options – show options).

19 It seems that our scan has been successful and Metasploit has a few sessions open for us. Let's see if we can interact with one of them.

Type: msf auxiliary(telnet_login) > **sessions –l**

What do you observe?

Type:msf auxiliary(telnet_login) > **sessions -i 1**
id
uid
exit
logout

Appendix

User logins:

Ubuntu (User: napier, Password: napier123).
Windows2003 (User: Administrator, Password: napier).
Windows 2008 (User: Administrator, Password: Ankle123).
Pfsense (User: admin, Password: pfsense).
Windows 7 (User: EnCase, Password: napier).
Kali (User: root, Password: toor).