Ethical Hacking Tutorial Index

* [Ethical Hacking Tutorial](https://www.javatpoint.com/ethical-hacking)
* [Types of Hackers](https://www.javatpoint.com/types-of-hackers)
* [Famous Hackers](https://www.javatpoint.com/famous-hackers)
* [Environmental setup](https://www.javatpoint.com/ethical-hacking-environment-setup)

**Network Penetration**

* [Network Penetration Testing](https://www.javatpoint.com/network-penetration-testing)
* [Basic of Network](https://www.javatpoint.com/ethical-hacking-basic-network)

**Pre-connection Attacks**

* [Pre-connection Attacks](https://www.javatpoint.com/pre-connection-attacks)
* [Wireless Interface in Monitor mode](https://www.javatpoint.com/wireless-interface-in-monitor-mode)
* [About airodump-ng](https://www.javatpoint.com/airodump-ng)
* [Run airodump-ng](https://www.javatpoint.com/run-airodump-ng)
* [Deauthenticate](https://www.javatpoint.com/deauthenticate-wireless-client)

**Gaining Access**

* [Introduction](https://www.javatpoint.com/gaining-access-introduction)
* [Gaining Access](https://www.javatpoint.com/gaining-access)
* [WEP Introduction](https://www.javatpoint.com/wep-introduction)
* [WEP Cracking](https://www.javatpoint.com/wep-cracking)
* [Fake Authentication Attack](https://www.javatpoint.com/fake-authentication-attack)
* [ARP Request Replay Attack](https://www.javatpoint.com/arp-request-replay-attack)
* [WPA Theory](https://www.javatpoint.com/wpa-theory)
* [Handshake Theory](https://www.javatpoint.com/handshake-theory)
* [Capturing Handshakes](https://www.javatpoint.com/capturing-the-handshake)
* [Creating Wordlist](https://www.javatpoint.com/creating-wordlist)
* [Cracking Wordlist](https://www.javatpoint.com/cracking-wordlist)
* [Securing Network from Attacks](https://www.javatpoint.com/securing-network-from-attacks)

**Post Attacks**

* [Post-Connection Attacks](https://www.javatpoint.com/post-connection-attacks)
* [Netdiscover](https://www.javatpoint.com/netdiscover)
* [Zenmap](https://www.javatpoint.com/zenmap)

**MITM Attacks**

* [MITM Attacks](https://www.javatpoint.com/man-in-the-middle-attacks)
* [ARP spoofing using arpspoof](https://www.javatpoint.com/arp-spoofing-using-arpspoof)
* [ARP spoofing using MITMf](https://www.javatpoint.com/arp-spoofing-using-mitmf)
* [Bypassing HTTPS](https://www.javatpoint.com/bypassing-https)
* [DNS Spoofing](https://www.javatpoint.com/dns-spoofing)

**Server Attacks**

* [Server-side Attacks](https://www.javatpoint.com/server-side-attacks)
* [Server-side Attack Basics](https://www.javatpoint.com/server-side-attack-basics)
* [Attacks - Metasploit basics](https://www.javatpoint.com/server-side-attacks-metasploit-basics)
* [Exploiting a Code Execution Vulnerability](https://www.javatpoint.com/exploiting-a-code-execution-vulnerability)
* [Installing MSFC](https://www.javatpoint.com/installing-msfc)
* [MSFC Scan](https://www.javatpoint.com/msfc-scan)
* [MSFC Analysis](https://www.javatpoint.com/msfc-analysis)
* [Installing Nexpose](https://www.javatpoint.com/ethical-hacking-installing-nexpose)
* [Nexpose Scanning](https://www.javatpoint.com/ethical-hacking-nexpose-scanning)

Ethical Hacking Tutorial



Ethical Hacking tutorial provides basic and advanced concepts of Ethical Hacking. Our Ethical Hacking tutorial is developed for beginners and professionals.

Ethical hacking tutorial covers all the aspects associated with hacking. Firstly, we will learn how to install the needed software. After this, we will learn the 4 type of penetration testing section which is network hacking, gaining access, post exploitation, website hacking.

In network hacking section, we will learn how networks work, how to crack Wi-Fi keys and gain access the Wi-Fi networks. In Gaining access section, we will learn how to gain access to the servers and personal computers. In the post-exploitation section, we will learn what can we do with the access that we gained in the previous section. So we learn how to interact with the file system, how to execute a system command, how to open the webcam. In the website hacking section, we will learn how the website works, how to gather comprehensive information about website. In the end, we will learn how to secure our system from the discussed attacks.

What is Hacking?

Gaining access to a system that you are not supposed to have access is considered as hacking. For example: login into an email account that is not supposed to have access, gaining access to a remote computer that you are not supposed to have access, reading information that you are not supposed to able to read is considered as hacking. There are a large number of ways to hack a system.

In 1960, the first known event of hacking had taken place at MIT and at the same time, the term **Hacker** was organized.

What is Ethical hacking?

Ethical hacking is also known as **White hat Hacking or Penetration Testing**. Ethical hacking involves an authorized attempt to gain unauthorized access to a computer system or data. Ethical hacking is used to improve the security of the systems and networks by fixing the vulnerability found while testing.

Ethical hackers improve the security posture of an organization. Ethical hackers use the same tools, tricks, and techniques that malicious hackers used, but with the permission of the authorized person. The purpose of ethical hacking is to improve the security and to defend the systems from attacks by malicious users.

Types of Hacking

We can define hacking into different categories, based on what is being hacked. These are as follows:

1. Network Hacking
2. Website Hacking
3. Computer Hacking
4. Password Hacking
5. Email Hacking
6. **Network Hacking:** Network hacking means gathering information about a network with the intent to harm the network system and hamper its operations using the various tools like Telnet, NS lookup, Ping, Tracert, etc.
7. **Website hacking:** Website hacking means taking unauthorized access over a web server, database and make a change in the information.
8. **Computer hacking:** Computer hacking means unauthorized access to the Computer and steals the information from PC like Computer ID and password by applying hacking methods.
9. **Password hacking:** Password hacking is the process of recovering secret passwords from data that has been already stored in the computer system.
10. **Email hacking:** Email hacking means unauthorized access on an Email account and using it without the owner's permission.

Advantages of Hacking

There are various advantages of hacking:

1. It is used to recover the lost of information, especially when you lost your password.
2. It is used to perform penetration testing to increase the security of the computer and network.
3. It is used to test how good security is on your network.

Disadvantages of Hacking

There are various disadvantages of hacking:

1. It can harm the privacy of someone.
2. Hacking is illegal.
3. Criminal can use hacking to their advantage.
4. Hampering system operations.

## Prerequisite

There is nothing specific prerequisite for learning computer network.

## Audience

Our Ethical Hacking Tutorial is designed to help beginners and professionals.

## Problems

We assure that you will not find any problem in this Ethical Hacking Tutorial. But if there is any mistake, please post the problem in contact form.

# **Types of Hackers**

Hackers can be classified into three different categories:

1. Black Hat Hacker
2. White Hat Hacker
3. Grey Hat Hacker



## Black Hat Hacker

Black-hat Hackers are also known as an **Unethical Hacker or a Security Cracker**. These people hack the system illegally to steal money or to achieve their own illegal goals. They find banks or other companies with weak security and steal money or credit card information. They can also modify or destroy the data as well. Black hat hacking is illegal.



## White Hat Hacker

White hat Hackers are also known as **Ethical Hackers or a Penetration Tester**. White hat hackers are the good guys of the hacker world.

These people use the same technique used by the black hat hackers. They also hack the system, but they can only hack the system that they have permission to hack in order to test the security of the system. They focus on security and protecting IT system. White hat hacking is legal.



## Gray Hat Hacker

Gray hat Hackers are Hybrid between Black hat Hackers and White hat hackers. They can hack any system even if they don't have permission to test the security of the system but they will never steal money or damage the system.

In most cases, they tell the administrator of that system. But they are also illegal because they test the security of the system that they do not have permission to test. Grey hat hacking is sometimes acted legally and sometimes not.

# **Famous Hackers**

In this section, we will see some of the famous hackers and how they become famous.

## Jonathan James

Jonathan James was an American hacker. He is the first Juvenile who send to prison for cybercrime in the United States. He committed suicide on **18 May 2008**, of a self-inflicted gunshot wound.

In 1999, at the age of 16, he gained access to several computers by breaking the password of a NASA server and stole the source code of International Space Station, including control of the temperature and humidity within the living space.

## Kevin Mitnick

He is a computer security consultant, author, and hacker. He infiltrates his client's companies to expose their security strengths, weaknesses, and potential loopholes. In the history of the United state, he was formerly the most wanted computer criminal.

Advertisement

From the 1970s up until his last arrest in 1995, he skillfully bypassed corporate security safeguards and found his way into some of the most well-guarded systems like Sun Microsystems, Nokia, Motorola, Netcom, Digital Equipment Corporation.

## Mark Abene

Mark Abene is an American Infosec expert and Entrepreneur. He is known around the world by his pseudonym Phiber Optik. Once, he was a member of the hacker groups Legion of Doom and Master of Deception. He was a high profile hacker in the 1980s and early 1990s.

He openly debated and defended the positive merits of ethical hacking as a beneficial tool for the industry. He is also expert in penetration studies, security policy review and generation, on-site security assessments, systems administration, and network management, among many others.

## Robert Morris

Robert Morris was the creator of the Morris Worm. He was the first computer worm to be unleashed on the Internet. The Morris Worm had the capability to slow down computers and make them no longer usable. Due to this, he was sentenced to three years probation, 400 hours of community service and also had to pay a penalty amount of $10,500.

## Gary McKinnon

Gary McKinnon is a Scottish systems administrator and Hacker. In 2002, he was accused of the **"biggest military computer hack of all time"**. He has successfully hacked the network of Navy, Army, Air Force, NASA system of the United States Government.

In his statement to the media, he has often mentioned that his motivation was only to find evidence of UFOs and the suppression of "free energy" that could potentially be useful to the public.

## Linus Torvalds

Linus Torvalds is a Finnish-American software engineer and one of the best hackers of all the time. He is the developer of the very popular Unix-based operating system called as Linux. Linux operating system is open source, and thousands of developers have contributed to its kernel. However, he remains the ultimate authority on what new code is incorporated into the standard Linux kernel.

Torvalds just aspire to be simple and have fun by making the world's best operating system. Linus Torvalds has received honorary doctorates from University of Helsinki and Stockholm University.

## Kevin Poulsen

Kevin Poulsen is an American former Black-hat hacker. He is also known as Dark Dante. He took over all the telephone lines of radio station KIIS-FM of Los Angeles, guaranteeing that he would be the 102nd caller and win the prize of a Porsche 944 S2.

Poulsen also drew the ire of FBI, when he hacked into federal computers for wiretap information. As a result of this, he was sentenced for five years. He has reinvented himself as a journalist.

# **Environmental Setup**

To perform ethical hacking, we have to download the Kali Linux Operating System and we can download Kali Linux OS inside the Virtual box. Here are the basic steps to download the virtual box and Kali Linux.

## Step 1: Download Virtual Box

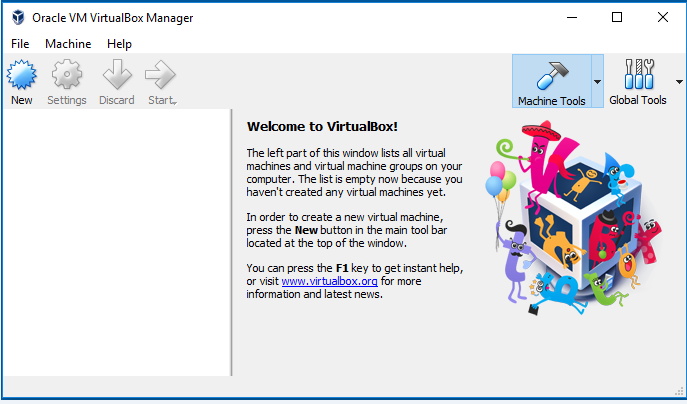
In step1, we download the Virtual box because the virtual box allows us to create a virtual machine inside our current operating system. After this, we will download the Kali Linux. A virtual machine is just like a completely separate working machine. You will lose nothing if you install an operating system inside the virtual machine. The operating system will perform just like the install on a separate laptop.

Now using the following link, you can download the virtual box according to your operating system and install it.

<https://www.virtualbox.org/wiki/Downloads>

Advertisement

After installation, the virtual box will be shown as follows:



## Step 2: Download Kali Linux

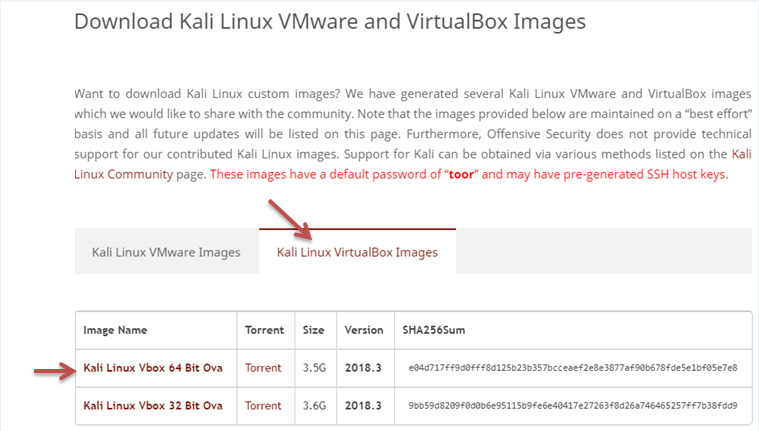
Now we will download the Kali Linux. It contained all the programs and application that we need to use pre-installed and preconfigured that means we just need to install this operating system and start hacking.

There are two ways to install Kali. You can install it as a virtual machine inside your current operating system, or you can download it as a main operating system. In this tutorial, I am going to use a virtual machine.

Use the following link to download the Kali Linux operating system.

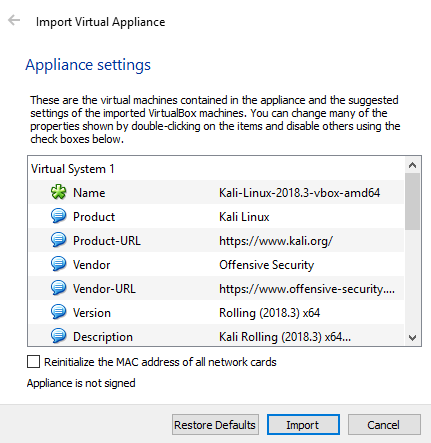
<https://www.offensive-security.com/kali-linux-vm-vmware-virtualbox-image-download/>

Now click on Kali Linux VirtualBox Images and download the Kali Linux according to the compatibility of your operating system.

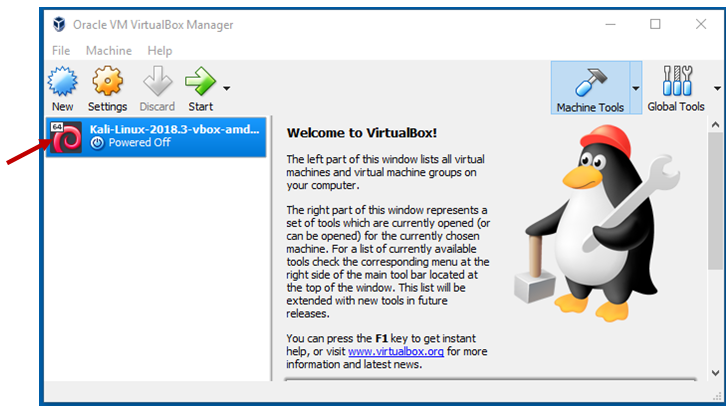


1. Download the 64-bit version if your computer is 64 bits otherwise, download the 32-bit version.
2. The downloaded file has a .ova extension. If the file doesn't have .ova extension that means you downloaded the wrong file.

After downloading, you will get a file with .ova extension. Now, to install the Kali Linux, you need to just double click on the file and click on the import button.

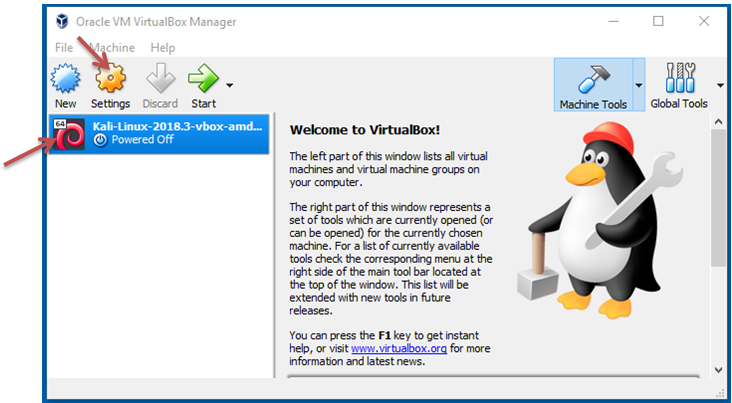


After installation, the Kali Linux is ready to use and will look like as follows:

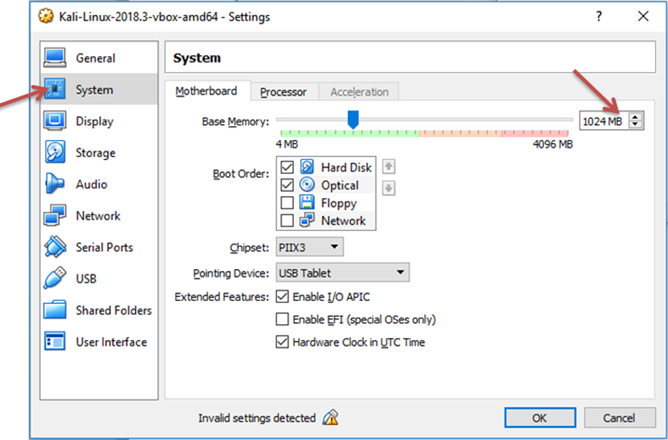


## Step 3: Modify some settings of Kali Linux

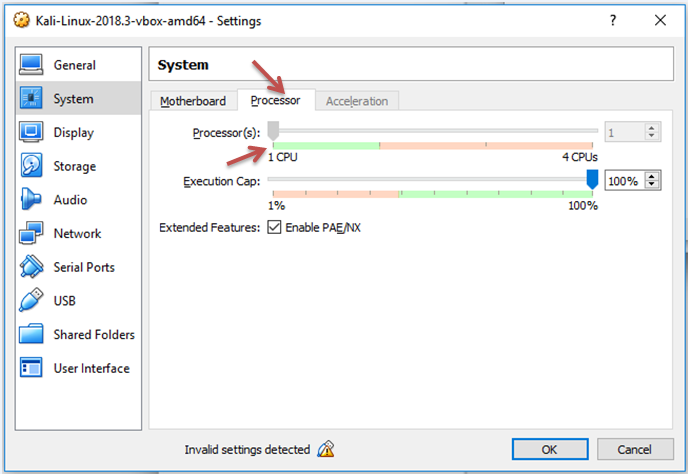
Before starting, we will modify some Settings. So just click on Kali Linux on the left side and then click on the Settings.



Now click on System and modify the amount of RAM depending on the amount of RAM on your computer. You can give it as 2GB if you want, but 1GB is enough for Kali.



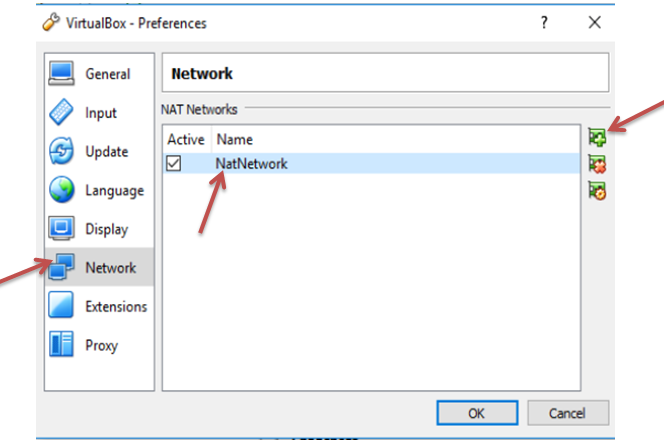
If you click on the Processor, then you can modify the amount of Processor as 2CPU, but 1 CPU is enough for Kali.



Now click on Network Settings and set "Attached to" as "NAT Network" but sometimes the network is automatically created by the virtual box, and sometimes the virtual box doesn't create this network automatically. If it is automatically created then click OK. If it is not created then the following screen will be shown:



If the virtual box is not automatically created the network, then just go to the VirtualBox → Preferences → Network → + sign. Now you can see that it creates another network.



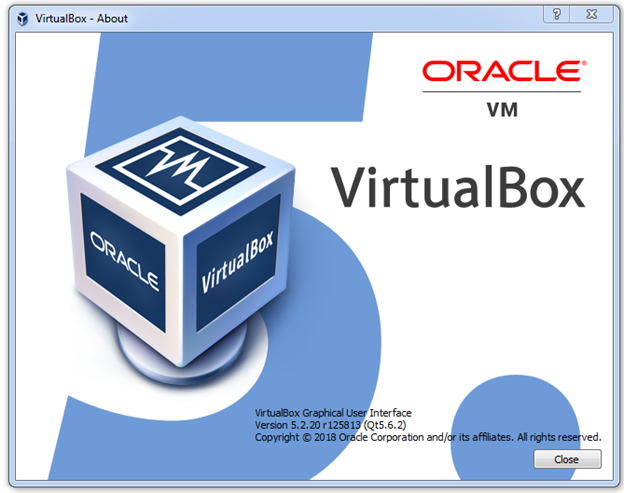
## Step 4: Starting of Kali Linux

Now we are starting the Kali Linux by clicking the start button. After clicking two cases will arise:

* Sometimes it will run successfully.
* Sometimes you will get an error like this:



To fix this error, you have to download the **Oracle VM VirtualBox Extension Pack of the same version of VirtualBox**. To find the version of Virtual Box just click on Help then click on About VirtualBox.



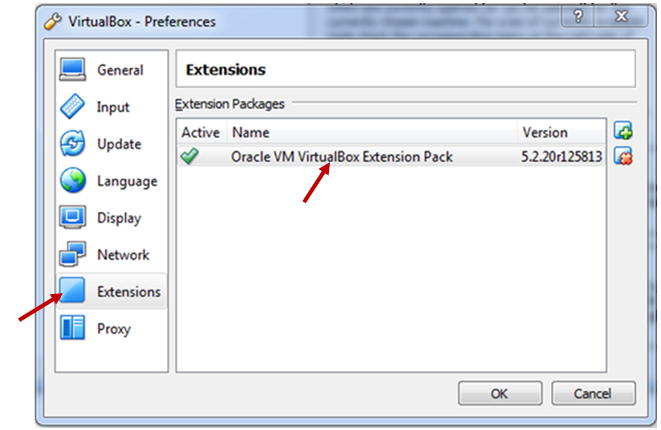
Now download the VirtualBox Extension of 5.2.20 version. Use the following link to download it:

<https://download.virtualbox.org/virtualbox/5.0.20/>

Now click on [Oracle\_VM\_VirtualBox\_Extension\_Pack-5.0.20.vbox-extpack](https://download.virtualbox.org/virtualbox/5.0.20/Oracle_VM_VirtualBox_Extension_Pack-5.0.20.vbox-extpack) and download it.



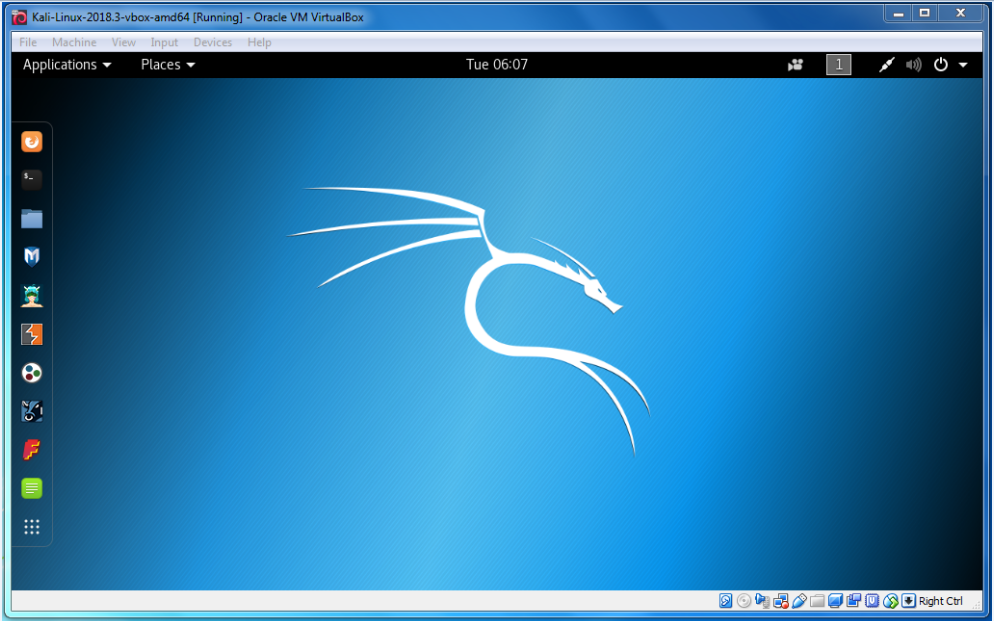
Install the VirtualBox extension pack. After installing, to check it clicks on File → Preferences → Extensions. Here you can see the Oracle VM VirtualBox Extension Pack. Click OK.



Now the problem is fixed, and we can start the virtual machine by clicking the start button.



After starting, it will ask us for the **Username**, and the default Username is root then it will ask us for the **password** and the default password is the reverse of root which is **toor**. Now you will get a screen like this:



# **Network Penetration Testing**

Network penetration testing is the first penetration testing that we are going to cover in this section. Most of the systems and computers are connected to a network. If a device is connected to the internet, that means the device is connected to the network because the internet is a really big network. Therefore, we need to know that how devices interact with each other in a network, as well as how networks works.

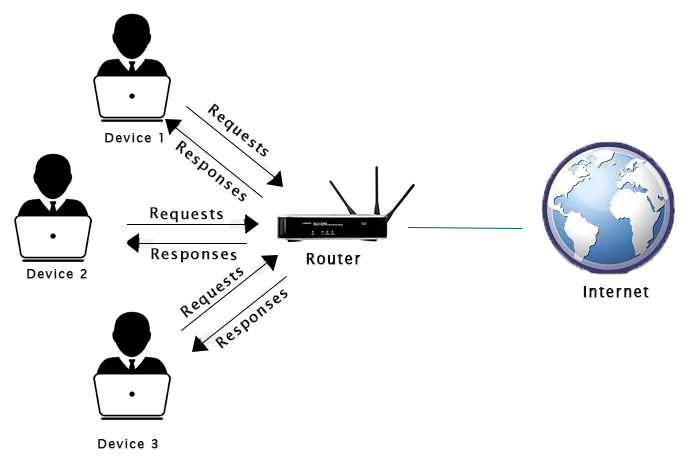
Network penetration testing is divided into 3 subsections:

1. [**Pre-connection attacks:**](https://www.javatpoint.com/pre-connection-attacks) In this section, we will learn about all the attacks that we can do before connecting to a network.
2. **Gaining attacks:** In this section, we will learn that how to crack Wi-Fi keys and gain access to Wi-Fi network whether they use WEP/WPA/WPA2 network.
3. **Post-connection attacks:** These attacks apply whenever you are able to connect to the network. In this section, you will learn the number of powerful attacks that will allow you to intercept the connections and capture everything like the user-name, password, URL, chat messages. You can also modify the data as it has been sent in the air. These attacks can apply on both Wi-Fi or wired networks.

# **Basic of Network**

A network is a group of two or more devices that are connected to each other to share the data or share the resource. A network contains a number of different computer system that is connected by a physical or wireless connection like server or router. This router has direct access to the internet. The device can only connect to the internet through the router or access point.

**For example:** Suppose the client or device connected to the network through Wi-Fi or Ethernet. If the client opens the browser and types google.com, then your computer will send a request to the router for asking google.com. The router will go to the internet and request google.com. The router will receive google.com and forward that response to the computer. Now the client can see google.com on the browser as a result.



In networking, devices on the same network communicate with each other using packets. If you send a video, login a website, sending chat messages, sending email, all the data is send as packets. In networking, devices ensure that these packets go in the right direction using the mac address. Each packet has the source mac and destination mac, and it flows from the source mac to destination mac.

# **Pre-connection Attack**

Pre-connection attack is the first part of the network penetration testing. To perform this attack, we will look at the fundamentals like how to show all the networks around us, how to find the details of all the connected devices to a particular network. Once we know about the network and connected devices to it, we can disconnect any device without knowing the password of that device.

Following are the basic steps we will be going through to perform Pre-connection attack:

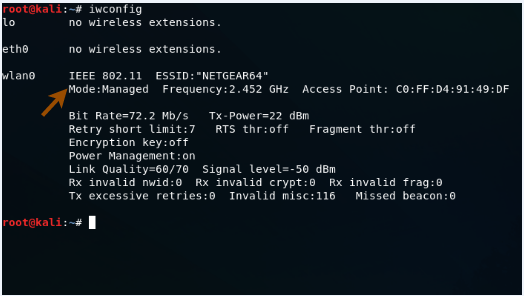
1. [**Wireless Interface in Monitor mode:**](https://www.javatpoint.com/wireless-interface-in-monitor-mode) In this step, we will change the mode of wireless device as Monitor mode.
2. [**About airodump-ng:**](https://www.javatpoint.com/airodump-ng) In this step, we will use airodump-ng to list all the network around us and display useful information about them.
3. [**Run airodump-ng:**](https://www.javatpoint.com/run-airodump-ng) In this step, we will see all the devices that are connected to a particular network and collect more information about it.
4. [**Deauthenticate the Wireless client:**](https://www.javatpoint.com/deauthenticate-wireless-client) In this step, we can disconnect any device which is shown in the previous step using the **aireplay-ng**.

# **Wireless interface in Monitor Mode**

This step is used to put your wireless card into Monitor mode. In Monitor mode, your card can listen to every packets that's around us. By default, the mode of wireless devices is set to "Managed" that means our wireless device will only capture packets that have our device's MAC address as the destination MAC. It will only capture packets that are actually directly to my Kali machine.

But we want to capture all the packets that are within our range even if the destination MAC is not our MAC or even without knowing the password of the target device. To do this, we need to set the mode as **Monitor** mode.

We can use **iwconfig** to see the wireless interfaces.



In the above image, you can see that the wireless interface wlan0 is in Managed mode. Use the following command to set it in Monitor mode.

Backward Skip 10sPlay VideoForward Skip 10s



**Where**

* **ifconfig wlan0 down** command is used for disabling the Managed mode
* **airmon-ng check kill** command is used to kill any process that could interfere with using my interface in monitor mode. After this command, your internet connection will be lost.
* **iwconfig wlan0 mode monitor** command is used to enable monitor mode
* **ifconfig wlan0 up** command is used to enable the interface
* **iwconfig** command shows that the mode is set to Monitor

In the above figure, you can see that the mode is changed as Monitor mode. Now we are able to capture all the Wi-Fi packets that are within our range even if the packets are not directed to our computer or even without knowing the password of the target network.

To do this, we need a program that can capture the packets for us. The program we are going to use is airodump-ng.

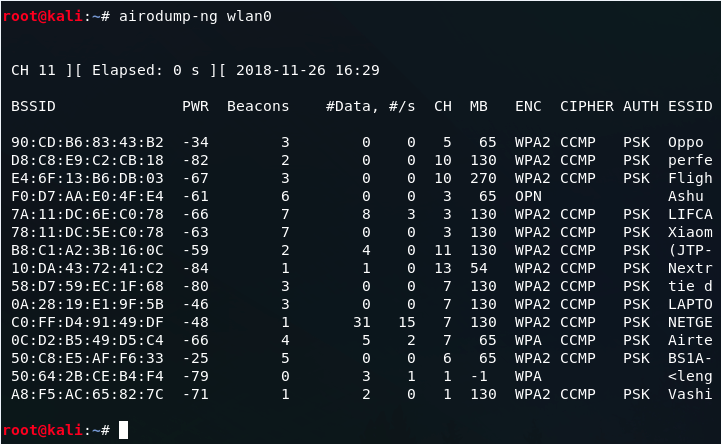
# **About airodump-ng**

airdump-ng is used to list all the network around us and display useful information about them. It is a packet sniffer, so it is basically designed to capture all the packets around us while we are in Monitor mode. We can run it against all of the networks around us and collect useful information like the mac address, channel name, encryption type, number of clients connected to the network and then start targeting to the target network. We can also run it against certain AP(access point) so that we only capture packets from a certain Wi-Fi network.

## Syntax

1. airodump-ng [MonitorModeInterface]

First, let's look at how to run the program. In this case, we need our Wi-Fi card in Monitor mode. The name of the our Wi-Fi card is **wlan0**.



#### **Note: We can press Ctrl + C to stop the following execution.**

**Where**

* **BSSID** shows the MAC address of the target network
* **PWR** shows the signal strength of the network. Higher the number has better signal
* **Beacons** are the frames send by the network in order to broadcast its existence
* **#Data**, shows the number of data packets or the number of data frames
* **#/s** shows the number of data packets that we collect in the past 10 seconds
* **CH** shows the channel on which the network works on
* **ENC** shows the encryption used by the network. It can be WEP, OPN, WPA, WPA2
* **CIPHER** shows the cipher used in the network
* **AUTH** shows the authentication used on the network
* **ESSID** shows the name of the network

In the above image, you can show all the wireless networks like Oppo, perfe, Fligh, Ashu, LIFCA, Xiaom, BS1A-YW5 etc and the detailed information about all the network.

Backward Skip 10sPlay VideoForward Skip 10s

#### **Note: airodump-ng is also used to identify all of the devices connected to the networks around us.**

# **Run airodump-ng**

In this step, we will run airodump-ng to see all the devices that are connected to a particular network and collect more information about it. Once we have a network to the target, it's useful to run airodump-ng on that network only, instead of running it on all the networks around us.

Currently, we are running airodump-ng on all the networks around us. Now we are going to target the network **BS1A-YW5** whose BSSID is **50:C8:E5:AF:F6:33**. We are going to sniff on that network only.

To do this, we will be use the same program. The command will be as follows:

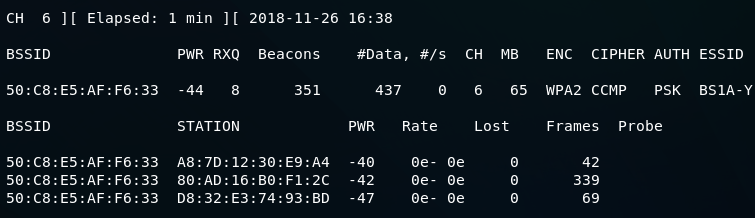
Run airodump-ng

**Where**

Advertisement

* **--bssid 50:C8:E5:AF:F6:33** is the access point MAC address. It is used to eliminate extraneous traffic.
* **--channel 11** is the channel for airodump-ng to snif on.
* **--write test** is used to store all the data in a file named as test. It is not mandatory, you can skip this part.
* **wlan0** is the interface name in Monitor mode.

After execution of this command, the following devices will be shown:



**Where**

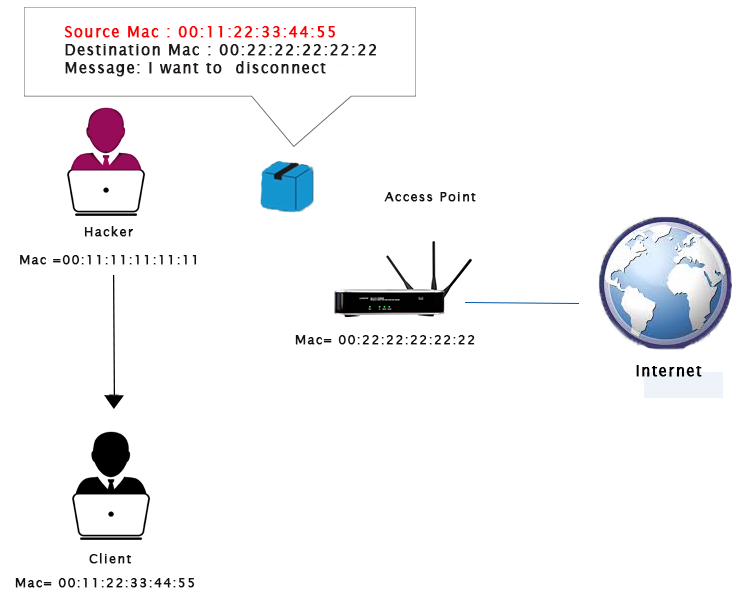
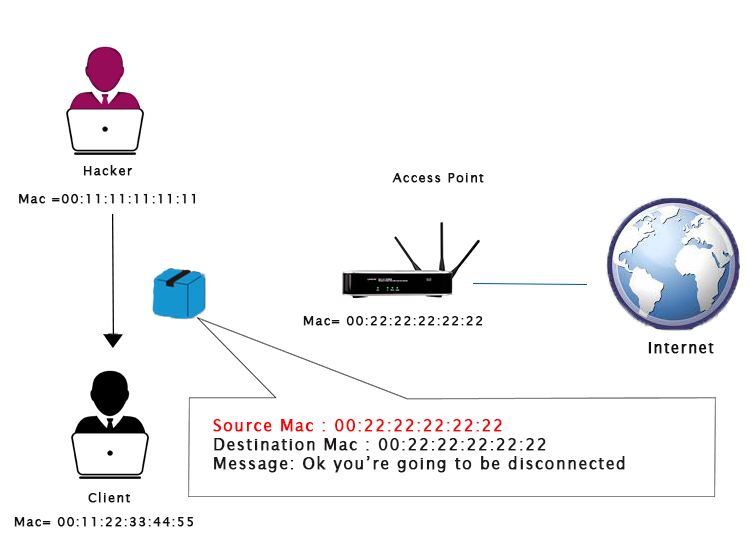
* **BSSID** of all the devices is same because devices are connected to the same network
* **STATION** shows the number of devices that are connected to this network
* **PWR** shows the power strength of each of the devices
* **Rate** shows the speed
* **Lost** shows the amount of data loss
* **Frames** show the number of frames that we have captured

After executing this command, we have 3 devices that are connected to the network BS1A-YW5 and all the devices have the same BSSID as 50:C8:E5:AF:F6:33.

# **Deauthenticate the wireless client**

It is also known as deauthentication attacks. These attacks are very useful. These attacks allow us to disconnect any device from any network that is within our range even if the network has encryption or uses a key.

In deauthentication attack, we are going to pretend to be client and send a deauthentication packet to the router by changing our MAC address to the MAC address of the client and tell the router that we want to disconnect from you. At the same time, we are going to pretend to be router by changing our MAC address to the router's MAC address until the client that we are requesting to be disconnected. After this, the connection will be lost. Through this process, we can disconnect or deauthenticate any client from any network. To do this, we will use a tool called aireplay-ng.

First of all, we will run airodump-ng on the target network, because we want to see which clients or devices are connected to it. This time, we will not need the --write option, so we are just going to remove it. After completion the run process of airodump-ng, we are going to disconnect the device with STATION A8:7D:12:30:E9:A4 using the airoplay-ng.

## Syntax

1. aireplay-ng --deauth [#DeauthPackets] -a [NetworkMac] -c [TargetMac] [Interface]

Deauthenticate the wireless client

After executing this command, the device whose STATION is A8:7D:12:30, lost the internet connection. We can only connect to the network again when we quit this executing command by pressing **Ctrl + C**.

PlayNext

Unmute

Current TimeÂ 0:00

/

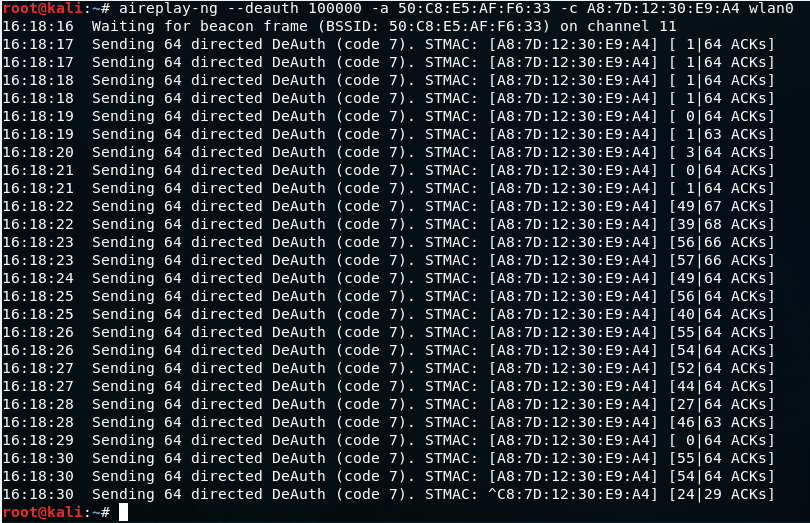
DurationÂ 18:10

Loaded: 0.37%

Â

Fullscreen

Backward Skip 10sPlay VideoForward Skip 10s



**Where**

* **-deauth** is used to tell airplay-ng that we want to run a deauthentication attack and assign 100000 which is the number of packets so that it keeps sending a deauthentication packets to both the router and client and keep the client disconnected.
* **-a** is used to specify the MAC address of the router. 50:C8:E5:AF:F6:33 is the target access point.
* **-c** specifies the MAC address of the client. A8:7D:12:30:E9:A4 is client's MAC address.
* **wlan0** is the wireless adaptor in Monitor mode.

# **Gaining Access**

Gaining access attack is the second part of the network penetration testing. In this section, we will connect to the network. This will allow us to launch more powerful attacks and get more accurate information. If a network doesn't use encryption, we can just connect to it and sniff out unencrypted data. If a network is wired, we can use a cable and connect to it, perhaps through changing our MAC address. The only problem is when the target use encryption like WEP, WPA, WPA2. If we do encounter encrypted data, we need to know the key to decrypt it, that's the main purpose of this chapter.

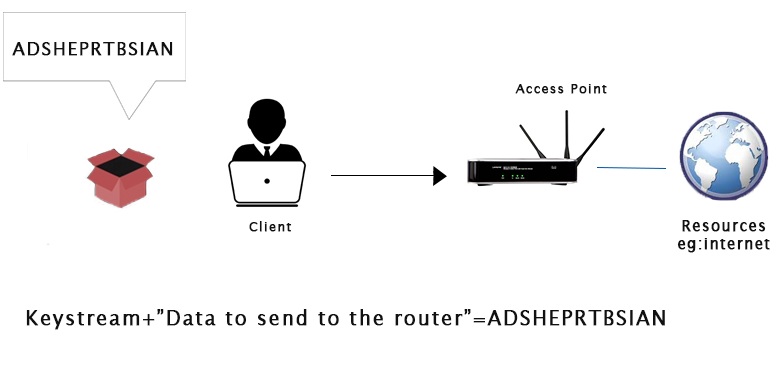
If the network uses encryption, we can't get anywhere unless we decrypt it. In this section, we will discuss that how to break that encryption and how to gain access to the networks whether they use WEP/WPA/WPA2.

This section will cover the following topics:

* [WEP Introduction](https://www.javatpoint.com/wep-introduction)
* [Basic WEP cracking](https://www.javatpoint.com/wep-cracking)
* [Fake authentication attack](https://www.javatpoint.com/fake-authentication-attack)
* [ARP request replay](https://www.javatpoint.com/arp-request-replay-attack)
* [WPA theory](https://www.javatpoint.com/wpa-theory)
* [Handshake theory](https://www.javatpoint.com/handshake-theory)
* [Capturing handshakes](https://www.javatpoint.com/capturing-the-handshake)
* [Creating wordlists](https://www.javatpoint.com/creating-wordlist)
* [Wordlist cracking](https://www.javatpoint.com/cracking-wordlist)
* [Securing network from attacks](https://www.javatpoint.com/securing-network-from-attacks)

# **WEP Introduction**

In this section, we will discuss WEP (Wired Equivalent Privacy). It is the oldest one, and it can be easily broken. WEP uses the algorithm called RC4 encryption. In this algorithm, each packet is encrypted at the router or access point and then send out into the air. Once the client receives this packet, the client will be able to transform it back to its original form because it has the key. In other words, we can say that the router encrypts the packet and send it, and the client receives and decrypts it. The Same happens if the client sends something to the router. It will first encrypt the packet using a key, send it to the router, and the router will be able to decrypt it, because it has the key. In this process, if a hacker captures the packet in the middle, then they will get the packet, but they wouldn't be able to see the contents of the packet because they do not have the key.



Each packet that is sent into the air has a unique keystream. The unique keystream is generated using a 24- bit IV (Initialization Vector). An initialization vector is a random number that is sent into each packet in plain text form, which is not encrypted. If someone captures the packet, they will not be able to read the packet content because it is encrypted, but they can read the IV in plain text form.

The weakness with the IV is that it is sent in the pain text and it is very short(only 24- bit). In a busy network, there will be a large number of packets sent in the air. At this time 24-bit number is not big enough. The IV will start repeating on a busy network. The repeated IVs can be used to determine the key stream. This makes WEP vulnerable to statistical attacks.

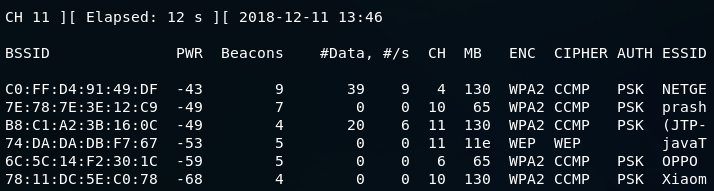
To determine the key stream we can use a tool called as aircrack-ng. This tool is used to determine the key stream. Once we have enough repeated IV, then it will also be able to crack WEP and give us the key to the network.

Backward Skip 10sPlay Video

# **WEP Cracking**

In order to crack WEP, we need first to capture the large number of packets that means we can capture a large number of IVs. Once we have done that, we will use a tool called aircrack-ng. This tool will be able to use statistical attacks to determine the key stream and the WEP key for the target network. This method is going to be better when we have more than two packets, and our chances of breaking the key will be higher.

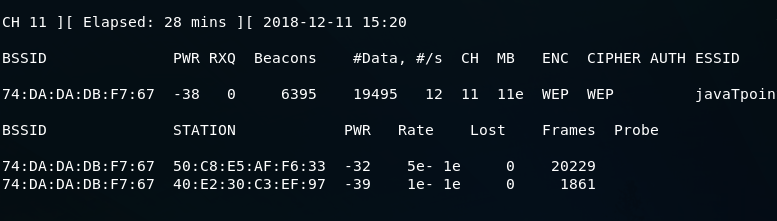
Let's look at the most basic case of cracking a WEP key. To do this, we will set WiFi card in monitor mode. After this, we will run a command **airodump-ng wlan0** to see all of the networks that are within our Wi-Fi range and then we will target one of those networks. Where wlan0 stands for the interface. The following output will be displayed after executing this command:



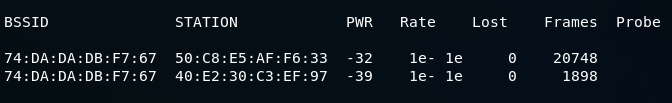
In this figure, the fourth network that has come up is **javaTpoint**. On this network, we are going to perform our attacks. We are going to run **airodump** against **javaTpoint** network by using the following command:

WEP Cracking

Here, we run **airodump** against the javaTpoint network with a --**bssid** as **74:DA:DA:DB:F7:67**. We include the --**channel**, number **11**, and we add --**write** to store all of the packets that we capture into a file, which is wep. After running the above command, the following output will be displayed:



This is a busy network. **#Data**, shows the number of useful packets that contain a different IV and we can use it to crack the key. If the number is higher, then it is more lightly to crack the key for us. In the following section, we can see the clients:



Now we use **ls** command to list all the file.

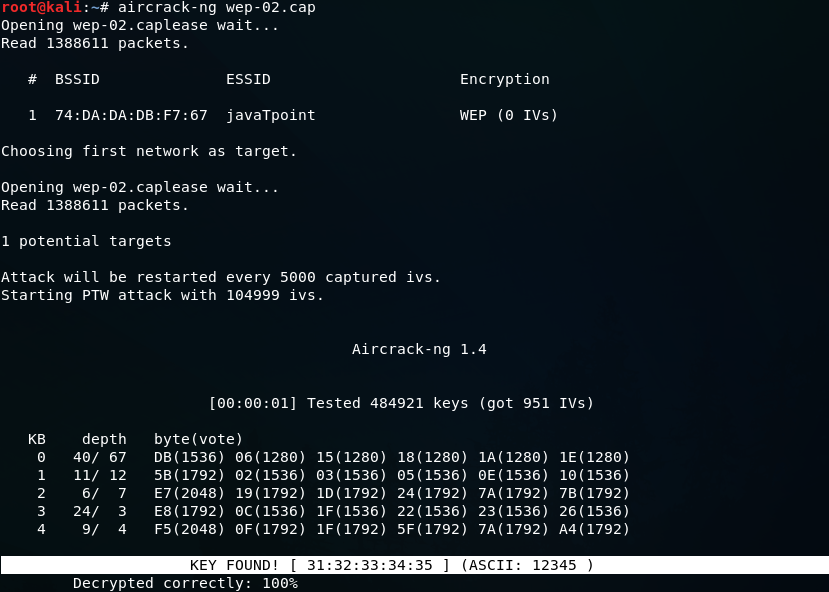
WEP Cracking

We can see that we have the captured file that was specified in the write argument. Now we will launch **aircrack-ng** against the file that **airodump** has created for us. We can launch **aircrack** against it even if we didn't stop airodump. It will keep reading the new packet that **airodump** is capturing. Use the following command in new terminal to run aircrack:

WEP Cracking

When we use **aircrack-ng**, we will put in the filename **wep.cap**. If aircrack fails to determine the key, **aircrack** waits until it reaches 5,000 IVs, and then tries again.

Now, we have to wait until the **aircrack** can successfully crack the WEP key. Once it decrypts the key, we can press Ctrl + C. In the following screenshot, **aircrack** has successfully managed to get the key within data packets:



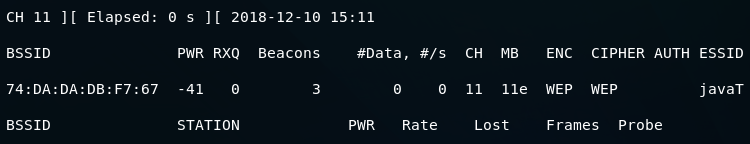
We can see that the key is found. So, we can connect to the target network, **javaTpoint** using **ASCII** password which is **12345**. We need just to copy the **12345** and paste it while connecting the javaTpoint. You can also connect using the **KEY** which is **31:32:33:34:35.** In some cases, we are not able to see the ASCII password, at that time we can use KEY to connect to the network. To do this, just copy 31:32:33:34:35 and remove the colons between the numbers. Now using the **3132333435** key, we can connect to the javaTpoint network.

# **Fake authentication attack**

In the previous section, we saw how easy it is to crack a WEP key on a busy network. In a busy network, the number of data increases very fast. One problem that we could face is if the network is not busy. If the network is not busy, the number of data will be increasing very very slowly. At that time we're going to fake as an AP that doesn't have any clients connected to it or an AP that has a client connected to it, but the client is not using the network as heavily as the client in the previous section.

Let's look at an example. We will run **airodump** against the target AP which is **javaTpoint**. We now have javaTpoint, the same AP that we used before, but the difference is that we've disconnected the clients that were connected to do this attack. As we can see, in the client area, there are no clients connected and the #Data is 0, it didn't even go to 1.

In this section, we want to be able to crack a key like this, with 0 data:



To solve this problem, what we can do is inject packets into the traffic. When we do this, we can force the AP to create a new packet with the new IVs in them, and then capture these IVs. But we have to authenticate our device with the target AP before we can inject packets. APs have lists of all of the devices that are connected to them. They can ignore any packets that come from a device that is not connected. If a device that doesn't have the key tries to send a packet to the router, the router will just ignore the packet, and it wouldn't even try to see what's inside it. Before we can inject packets into a router, we have to authenticate ourselves with the router. To do this, we're going to use a method called fake authentication.

PlayNext

Unmute

Current TimeÂ 0:00

/

DurationÂ 18:10

Loaded: 0.37%

Â

Fullscreen

Backward Skip 10sPlay VideoForward Skip 10s

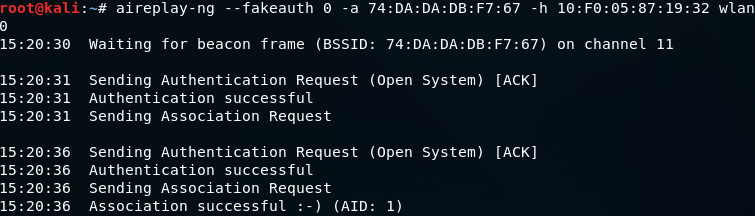
In the previous section, we already executed **airodump**. Let's see how we can use fake authentication. In the previous screenshot, we can see that **AUTH** have no value. Once we have done fake authentication, we will see an **OPN** show up there, which will mean that we have successfully falsely authenticated our device with the target AP. We will use the following command to do that:

Fake authentication attack

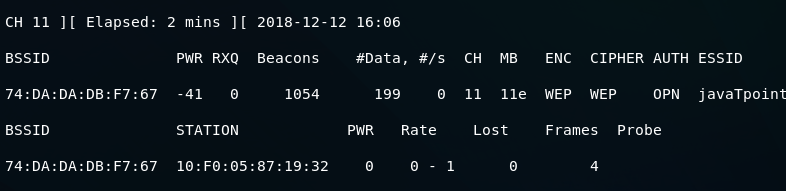
With **aireplay-ng**, we're going to use a --**fakeauth** attack. In this attack, we include the type of attack and the number of packets that we want to send, which is --**fakeauth** 0. We are going to use -a, to include the target network which is **74:DA:DA:DB:F7:67**. Then we're going to use -h, to include our **MAC** address. To get our MAC address, we are going to run the **ifconfig wlan0** command:

Fake authentication attack

Here, **wlan0** is the name of our Wi-Fi card. With aireplay-ng, the type of attack that we're trying to do, we're trying to perform a fake authentication attack, to authenticate our MAC address so that we can inject packets into the target network. We will send 0 which means do it once, then -a with the MAC address of the access point(AP), then -h with the MAC address of the device that we want to perform a fake authentication to, and then **wlan0**, the name of the WiFi card in monitor mode. Now we hit Enter:



In the above image, we can see that -a sent an authentication request, and it was successful. The network becomes an open network, and our client showed up as if it was a client connected to the network. We're not actually connected, but we are authenticated with the network and have an association with it so that we can inject packets into the AP. It will now receive any request that we send to it. Following is the output:



# **ARP request replay attack**

The AP now accepts packets that we send to it because we've successfully associated ourselves with it by using a fake authentication attack. We are now ready to inject packets into the AP and make the data increase very quickly, in order to decrypt the WEP key.

ARP request replay is the first method of packet injection. In this method, we're going to wait for an AP packet, capture the packet, and inject it into the traffic. Once we do this, the AP will be forced to create a new packet with a new IVs. We will capture the new packets, inject it back into the traffic again, and force the AP to create another packet with another IV. We will be repeating this process until the amount of data is high enough to crack the WEP key.

Using the following command we can launch **airodump-ng**:

ARP request replay attack

We're going to add a --**write** command to store all of the packets that we capture into a file which is **arp-request-reply-test**. When it runs, we will see that the target network has 0 data, it has no clients associated with it, and there is no traffic going through, which means that it's not useful, we can't crack its key.

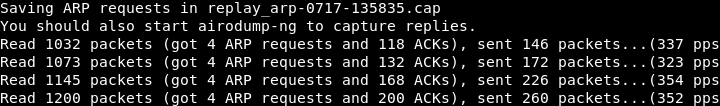
Backward Skip 10sPlay VideoForward Skip 10s

To solve this problem, we are going to perform a fake authentication attack as shown in the Fake **authentication** section, so that we can start injecting packets into the network, and it will accept them.

That leads us to our next step, which is the ARP request reply step. In this step, we will inject packets into the target network, forcing it to create new packets with new IVs. Following command is used to do this:

ARP request replay attack

This command is very similar to the previous command, but in this command, we're going to use --**arpreplay** instead of ?**fakeauth**. We will also include -b, for BSSID. With this command, we are going to wait for an ARP packet, capture it, and then reinject it out into the air. We can then see that we have captured an ARP packet, inject it, captured another, inject it into the traffic, and so on. The AP then creates new packets with new IVs, we receive them, we inject them again, and this happens over and over. After executing the above command, the following output will be shown:

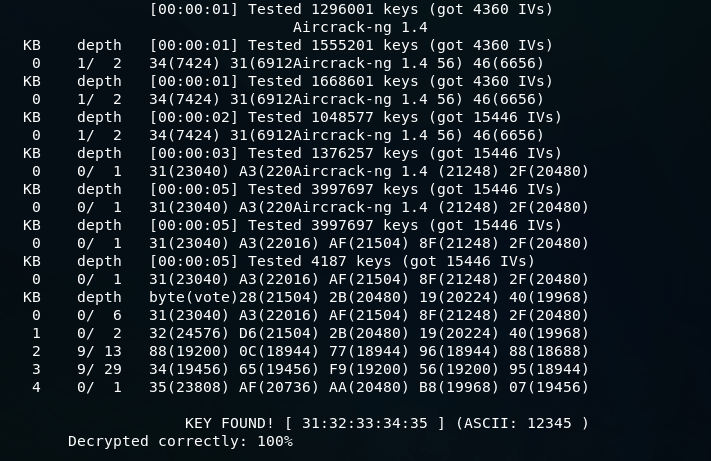


At this time, the wireless adapter wlan0 is waiting for an ARP packet. Once there is an ARP packet transmitted in the network, it's going to capture that packets and then retransmitted it. Once it has done, the access point will be forced to generate a new packet with a new IV, and we will keep doing this since the access point will continuously generate the new packet with new IV.

When the amount of Data reaches 9000 or above, we can launch **aircrack-ng** to crack it. Use the following command to do this:

ARP request replay attack

After running the above command, the following output will be shown. We can see the WEP Key, and we are able to crack it.



# **WPA Theory**

In this section, we are going to discuss **Wi-Fi Protected Access(WPA)** encryption. After WEP, this encryption was designed to address all of the issues that made WEP very easy to crack. In WEP, the main issue is the short IV, which is sent as plain text in each packet. The short IV means that the possibility of having a unique IV in each packet can be exhausted in active network so that when we are injecting packets, we will end up with more than one packet that has the same IV. At that time, **aircrack-ng** can use statistical attacks to determine the key stream and WEP key for the network.

In WPA, each packet is encrypted using a temporary key or unique key. It means that the number of data packets that we collect is irrelevant. If we collect one million packets, these packets are also not useful because they do not contain any information that we can use to crack the WPA key. WPA2 is the same as WPA. It works with the same methods and using the same method it can be cracked. The only difference between WPA, WPA2 is that WPA2 uses an algorithm called **Counter-Mode Cipher Block Chaining Message Authentication Code Protocol (CCMP) for encryption**.

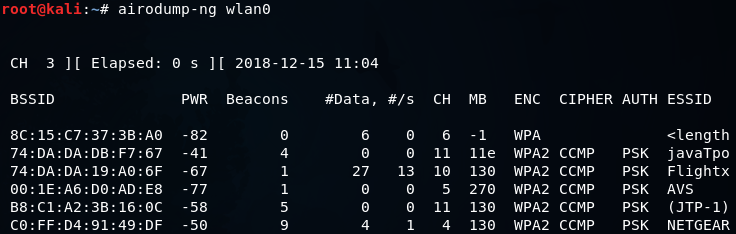
# **Handshake theory**

In WPA, each packet is encrypted using a unique temporary key. It is not like WEP, where IVs are repeated, and we collect a large number of data packets with the same IVs. In each WPA packet, there is a unique temporary IV, even if we collect 1 million packets, these packets will not be useful for us. These packets don?t contain any information that can help us to determine the actual WPA key.

The only packets that contain useful information and help us to determine the key are the **handshake packets**. These are the four packets, and these packets will be sent when a new device connects to the target network. For example, suppose we are at home, our device connect to the network using the password, and a process called four-way handshake happens between the AP and the devices. In this process, four packets called the handshake packets, get transferred between the two devices, to authenticate the device connection. We can use a wordlist using the aircrack-ng and test each password in the wordlist by using the handshake. To crack WPA encrypted network, we need two things: we need to capture the handshake, and we need a wordlist that contains passwords.

# **Capturing the handshake**

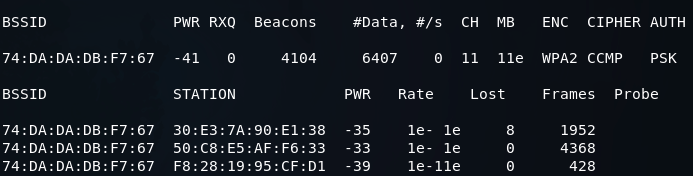
To crack WPA key, firstly we will capture the handshake. Using the airodump-ng, we will capture the handshake, in the same way, that we used it with WEP-encryption networks. Use the following command to capture all the network around us:



Now we will run airodump-ng against the javaTpoint network with a --bssid as 74:DA:DA:DB:F7:67. We will include the --channel, number 11, then we add --write to store all of the packets that we capture into a file which is wpa\_handshake, and then we include the wireless card in monitor mode which is wlan0. The command is as follows:

Capturing the handshake

Once we launch this command, we will have our WPA encrypted network, and we will have the clients connected to the network.



We can capture the handshake in two ways. First, we can just sit down and wait for a device to connect to the network. Once a device is connected then we can capture the handshake. Second, we can use deauthentication attack which we learned in the previous section, in Pre-connection attacks section.

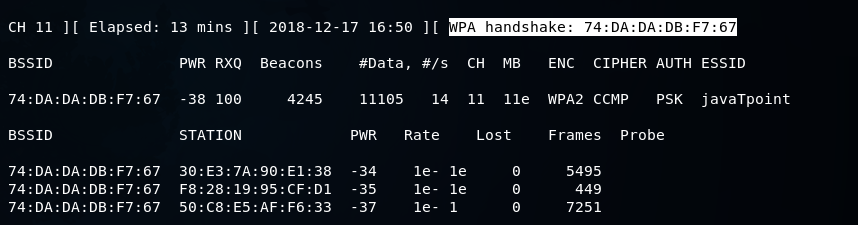
Backward Skip 10sPlay VideoForward Skip 10s

In a deauthentication attack, we can disconnect any device form a network that is within our Wi-Fi range. If we apply this attack for a very short period of time, we can disconnect a device form the network for a second, the device will try to connect to the network automatically, and even the person using the device will not notice that the device is disconnected or reconnected. Then we will be able to capture the handshake packets. The handshake gets sent every time a device connects to a target network.

Now using the **aireplay-ng**, we're just going to run a basic authentication attack. We use **aireplay-ng --deauth**, the name of the attack, and **4** authentication packets to the AP, and disconnect the device from it. Then we're going to put **-a**, to specify the MAC address of the target AP, and **-c**, to specify the client MAC address that we want to disconnect. Then we're going to put the name of the WIFI card, which is **wlan0**. The command is as follows:

Capturing the handshake

In the following screenshot, we can see that we captured the WPA handshake, and our target device didn't even change, nor was it disconnected:



We were disconnected for a very short period of time that's why we didn't get any message about being disconnected that's why even the person using the device didn't notice, and we were able to capture the handshake. To determine the WPA Key, we can use a wordlist and run it against the handshake.

# **Creating a Wordlist**

Now we've captured the handshake, all we need to do is create a wordlist to crack the WPA key. A wordlist is just a list of words that aircrack-ng is going to go through, and trying each one against the handshake until it successfully determines the WPA key. If the wordlist is better, the chances of cracking the WPA key will be higher. If the password is not in our wordlist file, we will not be able to determine the WPA key.

To create the wordlist, we're going to use a tool called crunch. The syntax is as follows:

1. crunch [min] [max] [characters] -o [FileName]
2. or
3. crunch [min] [max] [characters] -t [pattern] -o [FileName]

**where**

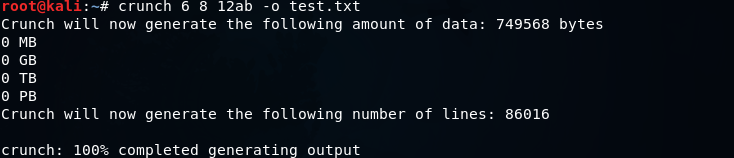
* **crunch** is the name of the tool.
* **[min]** specifies the minimum number of characters for the password to be generated.
* **[max]** specifies the maximum number of characters for the password.
* **characters** specify the characters that we want to use in the password. For example, you can put all lowercase characters, all uppercase characters, numbers, and symbols.
* **-t** is optional. It specifies the pattern.
* **-o** option specifies the filename where the passwords are going to be stored.

If we know the part of the password, **-t** option is very useful. For example: if we're trying to guess the password of someone and we have seen him typing the password, we know that the password starts with a and end with b. Now we can use the pattern option and tell crunch to create passwords that always start with a and end with b and put all possible combinations of the characters that we put in the command.

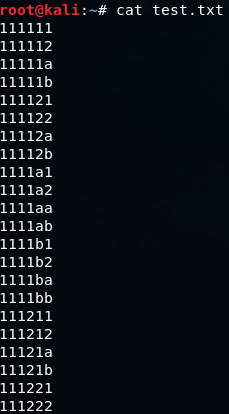
We're going to use **crunch**, and then we're going to make a minimum of **6** and maximum of **8**. We're going to put **12ab**, and store it in **test.txt**. The **crunch** is going to create a combination of passwords (minimum of 6 characters and maximum of 8 characters), and it's going to create all possible combination of **12ab**. It's going to store all the combination in a file called **test.txt**. The command will be as follows:

Creating a Wordlist

The following output will be shown after executing the above command:



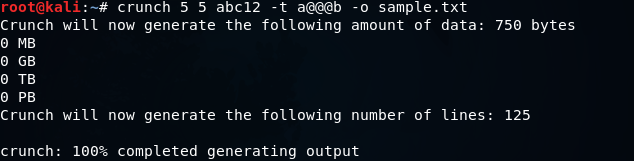
Using **cat test.txt** command, we can see all of the passwords that are stored in the file test.txt. The following screenshot shows all the passwords:



Now let's take a look at the pattern option. We will go to **crunch**, using a minimum of 5 and maximum of 5, so all password will be five characters long. Then we will put the characters, which are **abc12** and we will add the **-t** option, which is the pattern option, then we will put **a@@@b** that means the password starts with an a and end with b. Through this, we will get all possible combination of characters between a and b. Then, we are going to specify the output file **-o**, let's call it **sample.txt**. The command will be as follows:

Creating a Wordlist

The output will be as follows:



It creates **125** passwords. Now let's take a look at them. In the following screenshot, we can see that they always start with an **a** and always end with **b**.



We can use **crunch** to create the wordlist. In the next section, we're going to use the handshake file and the wordlist to determine the actual WPA key.

# **Wordlist cracking**

To crack WPA or WPA2, we need to first capture the handshake from the target AP and second have a wordlist which contains a number of passwords that we are going to try. Now we've captured the handshake, and we have a wordlist ready to use. Now we can use **aircrack-ng** to crack the key for the target AP. The **aircrack-ng** will be going through the wordlist file, combine each password with the name of the target AP, and create a **Pairwise Master Key(PMK)**. This PMK is created by using an algorithm called PBKDF2. It is not like just combining the password and the BSSID. It is encrypted in certain way, and compare the PMK to the handshake. The password that was used is the password for the target AP if the PMK is valid. If the PMK wasn't valid, then **aircrack-ng** tries the next password.

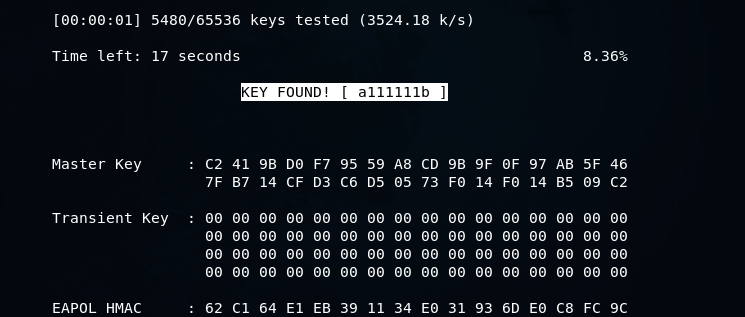
We will use **aircrack-ng**, the file name that contains the handshake, **wep\_handshake-01.cap**, -w and the name of the wordlist, **text.txt**. The command is as follows:

Wordlist cracking

Now click **Enter**, and **aircrack-ng** is going to go through the list of the password. It will try all of the passwords, and will combine each password with the name of the target AP to create a PMK, then compare the PMK to the handshake. If the PMK is valid, then the password that was used to create the PMK is the password for the target AP. If the PMK is not valid, then it's just going to try the next password.

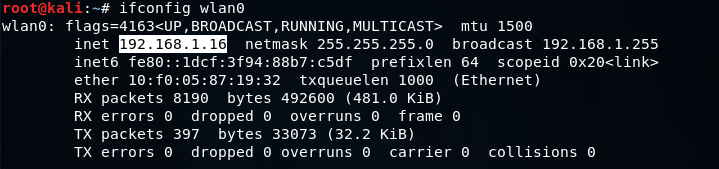
In the following screenshot, we can see that the key was found:

Backward Skip 10sPlay VideoForward Skip 10s



# **Securing network from attacks**

In order to prevent our network from preceding cracking methods explained in the pre-connection attacks and gaining access section, we'll need to access the settings page for our router. Each router has a wep page where we can modify the settings of our router, and it's usually at the IP of the router. First, we're going to get the IP of my computer and to do this we are going to run **ifconfig wlan0** command. As seen in the following screenshot, the highlighted part is the IP of the computer:

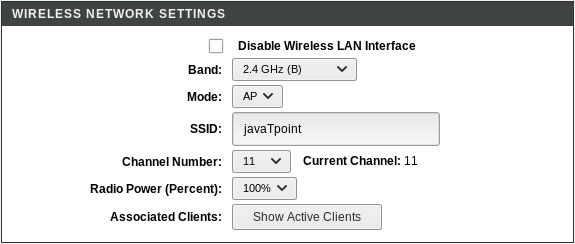


Now open the browser and navigate to 192.168.1.1. For this example, the IP of the computer is 16. Usually, the IP of the router is the first IP of the subnet. At the moment, it's 192.168.1.0, and we are just going to add the number 1 because that's the first IP in the subnet, and that will take us to the router settings page. At the setting page, it will ask to enter the username and password. To enter username and password, we can login to the router settings.

Sometimes the attacker might be doing deauthentication attack against us. To prevent it, what we can do is connect to the router using an Ethernet cable and modify our security settings and change the encryption, change the password, do all the things that are recommended in order to increase the security. So, the attacker will not be able to attack the network and get the key.

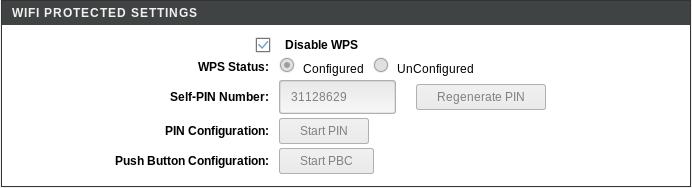
Now, the setting of each router is different. They depend on the model of the router. But usually, the way we change the setting is the same. Most of the cases, the router is always at the first IP of the subnet, we just need to get our IP using the ifconfig command, like we did at the start of this topic. We got the 192.168.1.16 IP, and then we changed the last 16 to 1 to the first IP, and that is IP of our router.

Now, we're going to the **WIRELESS NETWORK SETTINGS**. As we can see, there are lot of settings that we can change for our network:



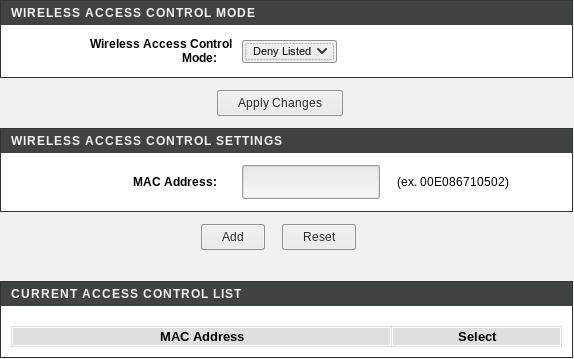
In the above screenshot, we can see that the wireless setting is **Enabled**, we can change the name of the network under **SSID**, we can also change the **Channel Number** and **Band**.

After going to the **WPS** option, we can see that WPS is **Disabled**. We are not using WEP that's why the attacker can't use any of the attacks to crack WEP encryption:



We have disabled WPS, and use WPA, which is much more secure, so the attacker can't use **reaver** to determine the WPS PIN and then reverse-engineer the password. The hacker can only get the password by obtaining the handshake first and then using a wordlist to find the password. The password of the network is very random, even though it doesn't actually use numbers or digits, just letters, so there are very small chances of someone being able to guess it.

After going to the **Access Control**, we can see that we can add Mode, such as an Allow List or a Deny list.



Here, we can specify the MAC address of the network that we want to allow to connect to our network. We can also specify the MAC address of the network that we want to deny form our network. For example, if we are in a company, and we have specified number of computers and we only want to allow a number of computers to connect to the network, then you can obtain the MAC address of the system that you want to allow and add them onto an **Allow list** or **Whitelist**. Even if a person has the actual key, and they don't exist in the **Allow List**, they will not be able to access the network. We can also add a certain computer or certain person onto a **Deny List** if we think that it is suspicious, we need to just add their MAC address onto the **Deny List**, and they will not be able to connect to our network.

# **Post-Connection Attacks**

All the attacks that we performed in the pre-connection and gaining access section, we weren't connected to a network. In this section, we are going to be talking about post-connection attack that means the attacks that we can do after connecting to the network. Now, it doesn't matter that the network is a wireless or a wired network and it doesn't matter that the target was using the WEP or WPA key, we can launch all of the attacks that we're going to talk about in this section.

In all the previous attacks, we kept our wireless card in monitor mode, so that we could capture any packet that goes in the air. In this section, we're going to use our wireless card in managed mode because we have access to the network, so we really don't need to capture everything, we only want to capture packets that are directed to us.

In this section, we're going to look at the attacks that can perform when we break through the network. Firstly, we're going to use a tool **netdiscover** to gather all the important information about the network, and that information will help us to launch attacks. It is used to explore all the clients that are connected to a system. After this, we will learn a tool called **Zenmap**. This tool has a better interface and is more powerful than netdiscover. This tool is used to gather detailed information about all of the clients connected to the same network.

# **netdiscover**

The netdiscover is a tool which is used to gather all the important information about the network. It gathers information about the connected clients and the router. As for the connected clients, we'll be able to know their IP, MAC address and the operating system, as well as the ports that they have open in their devices. As for the router, it will help us to know the manufacturer of the router. Then we'll be able to look for vulnerabilities that we can use against the clients or against the router if we are trying to hack them.

In the **Network penetration testing**, we used **airodump-ng** to discover all the connected clients to the network. In the second part of the **airodump-ng** output, we learned how we could see the associated clients and their MAC addresses. All these details we can get before we connect to the target access point. Now, after connecting to the network, we can gather much more detailed information about these devices. To do this task, there are a lot of programs, but we're going to talk about two programs. Now start with the simplest and quickest one, **netdiscover**.

The **netdiscover** is a quicker and simplest program to use, but it doesn't show very detailed information about the target clients. It'll only show us their IP address, their MAC address, and sometimes the hardware manufacturer. We're going to use it by typing netdiscover, then we are going to use -r, and then we are going to specify the range, which can be any range we want. Looking at the IP (which is 10.0.2.1) tells us which network we are in. We want to discover all the clients that are in this network, so we're going to try and see if there is a device in 10.0.2.1. Then we're going to try **12, 13, 14, 15, 16**, up to **254**, that's the end of the range. So, to specify a whole range, we can write /24. That means we want **10.0.2.1**, and then this IP is just going to increase up to **10.0.2.254**, which is the end of the IP range in the network. The command for this is as follows:

Netdiscover

Now hit Enter. It will return the output very fast, producing the result shown in the following screenshot:

PlayNext

Unmute

Current TimeÂ 0:00

/

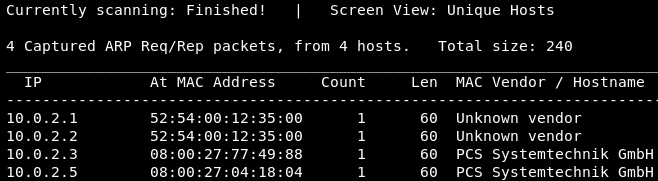
DurationÂ 18:10

Loaded: 0.37%

Â

Fullscreen

Backward Skip 10sPlay VideoForward Skip 10s

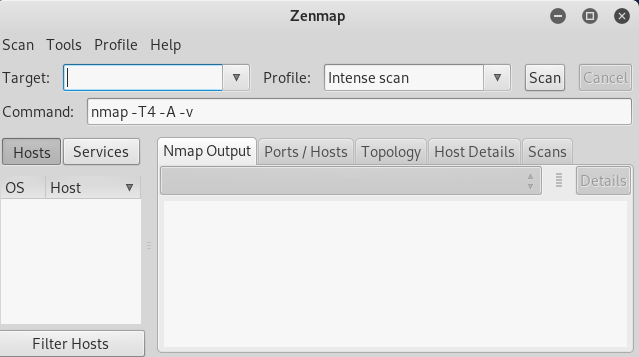


In the above screenshot, we can see that we have four devices connected to the network. We have their IP address, MAC address, and the MAC Vendor. This method was very quick, and it just shows simple information.

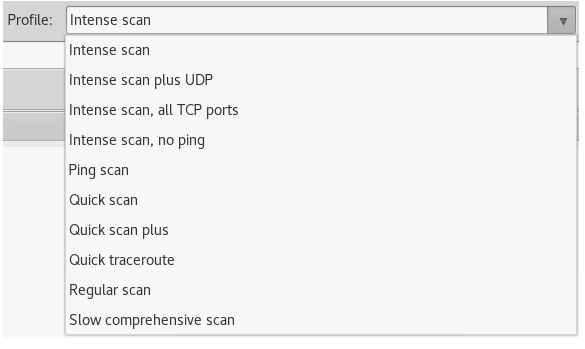
# **Zenmap**

**Nmap(Network Mapper)** is the second program that we're going to look. It is a huge tool and has many uses. Nmap is used to gather information about any device. Using the Nmap, we can gather information about any client that is within our network or outside our network, and we can gather information about clients just by knowing their IP. Nmap can be used to bypass firewalls, as well as all kinds of protection and security measures. In this section, we're going to learn some of the basic Nmap commands that can be used to discover clients that are connected to our network, and also discover the open ports on these clients.

We're going to use **Zenmap**, which is the graphical user interface for Nmap. If we type zenmap on the Terminal, we'll bring up the application like this:



In the **Target** field, we're going to put our IP address. In the **Profile** drop-down menu, we can have various profiles:



In the **Target** filed, if you want to gather information of only one IP address, we can just enter that address. We can also enter a range like we did with netdiscover. We're going to enter 198.168.1.1/24. Then we are going to select the **Ping scan** from the **Profile** drop-down menu and hit the **Scan** button:

PlayNext

Unmute

Current TimeÂ 0:00

/

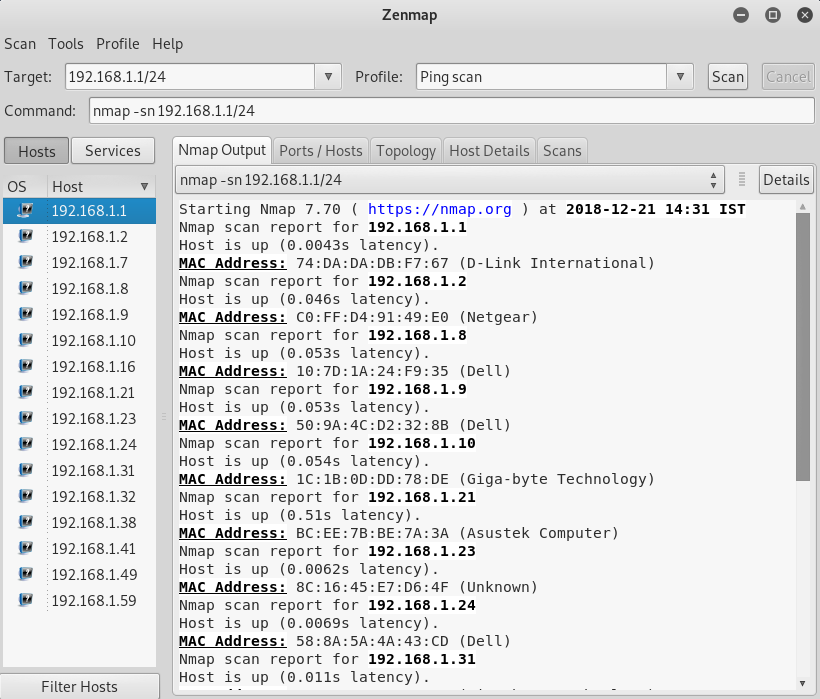
DurationÂ 18:10

Loaded: 0.37%

Â

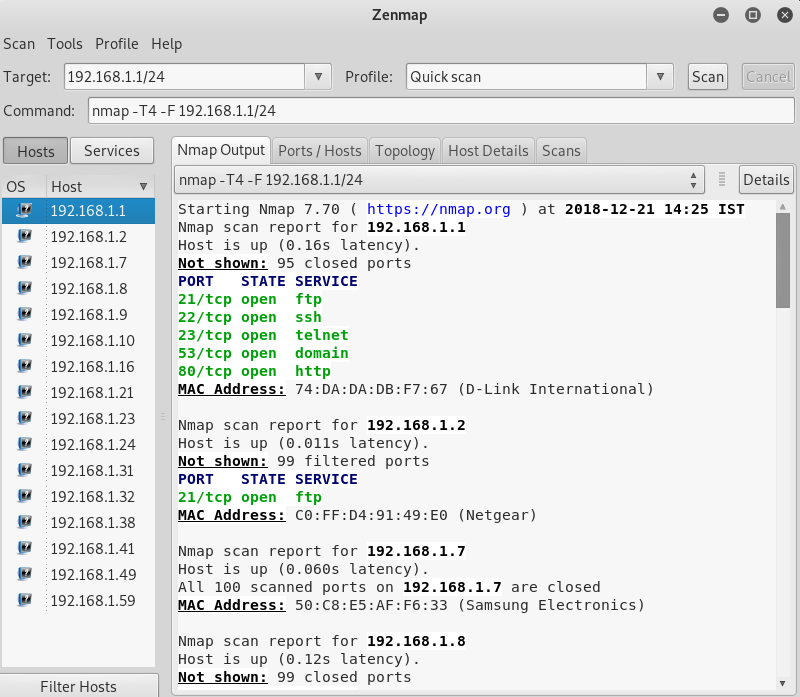
Fullscreen

Backward Skip 10sPlay VideoForward Skip 10s



The preceding scan is kind of a quick scan, but it doesn't show too much information, as we can see in the preceding screenshot. It only shows the connected devices. This scan is very quick. We are able to see the connected devices on the left-hand panel, and we can see their IP addresses, their MAC addresses, and their vendors.

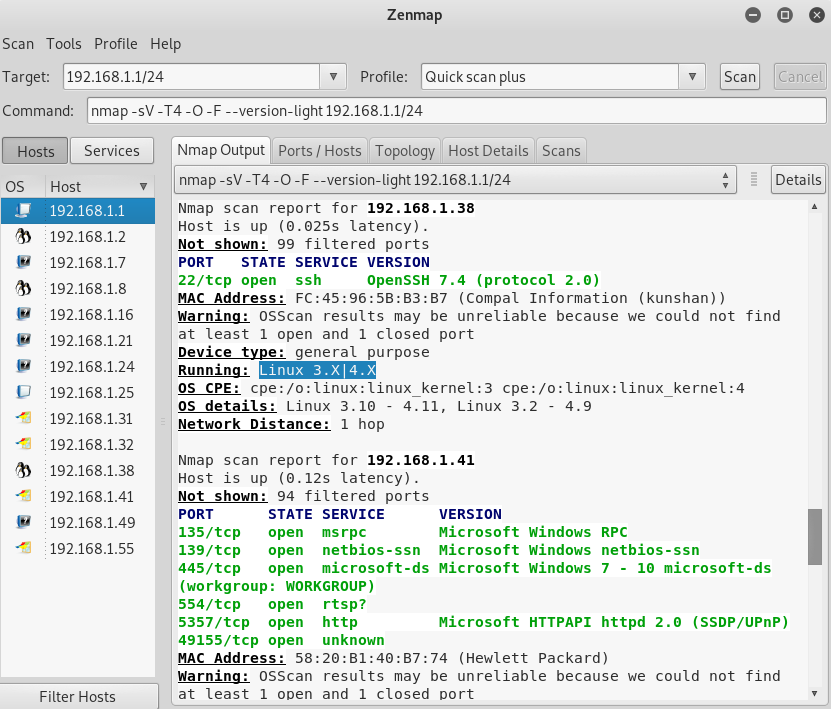
The next scan we're going to learn is the **Quick Scan**. Now, the **Quick scan** is going to be slightly slower than the **Ping scan**. But in Quick scan, we will get more information than the **Ping scan**. We're going to be able to identify the open ports on each device:



In the above screenshot, we can see that it shows the open ports on each one of the discovering devices. The main router has an open port called **53/tcp. 80/tcp** is the port used at the router setting page because it runs on a web server.

Now, let's take a look at the **Quick scan plus**, which take the **Quick scan** one step further. It's going to be slower than the Quick scan, but it will show us the programs that are running on the opened ports. So, in Quick scan, we saw that port **80** is open, but we didn't know what was running on port **80**, and we saw that port **22** was running, but we didn't know what was running. We knew it was SSH, but we don't know what SSH server was running on that port.

So again, Quick scan plus will take longer than Quick scan, but it will gather more information, as shown in the following screenshot:



In the preceding screenshot, we can see that we have a Linux device connected. We can see that the operating system of the device is connected and that it also got us the version for the programs. In Quick scan, we only knew that port **22** was open but now we know that it's running, and the server is OpenSSH 4.7. Now we know that it was Apache HTTP server 2.2.8 and it was a Linux device. We can go ahead and look for exploits and vulnerabilities.