BACHELOR IN TELECOMMUNICATION TECHNOLOGIES AND SERVICES ENGINEERING

DESIGN, DEVELOPMENT OF AN INTERNET OF THINGS ADVANCED BOTNET AND PROPOSAL OF COUNTERMEASURES

Author: D. Alejandro Antonio Ly Liu
Tutor: D. Victor Abraham Villagrá González
Department: Telematics Systems Engineering Department

Tribunal:

President:
Vocal:
Secretary:
Substitute:

Reading date:
Grade:

UNIVERSIDAD POLITÉCNICA DE MADRID

ESCUELA TÉCNICA SUPERIOR DE INGENIEROS DE TELECOMUNICACIÓN

BACHELOR IN TELECOMMUNICATION TECHNOLOGIES AND SERVICES ENGINEERING

FINAL PROJECT

DESIGN, DEVELOPMENT OF AN INTERNET OF THINGS ADVANCED BOTNET AND PROPOSAL OF COUNTERMEASURES

Author: Alejandro Antonio Ly Liu

Tutor: Víctor A. Villagrá González

Course 2018/2019

Abstract

In this Bachelor thesis, we will go through a piece of the cybersecurity world. Firstly, we will analyze the current situation of IoT; then, we will proceed through the historical evolution of botnets; and finally, the current situation of companies and their point of view in IoT security.

We will deepen in the world of botnets by analyzing their architecture, protocols and operation. After that, we will go through the development of an advanced botnet by choosing the best architecture and communication system; its implementation and deployment. Afterwards, we will make an analysis of the measures and countermeasures that could be used for preventing botnets in our systems. At the same time, we will analyze the development of infection, evasion and resilience techniques so we can prepare ourselves against them.

Finally, we will proceed to analyze the services that botnets provide and the impact on the different entities.

Keywords: Security, Internet of Things, Botnet, Distributed system, Malware services, Countermeasures, Detection.

Resumen

Comenzaremos este Trabajo de Fin de Grado adentrándonos en una parte del mundo de la ciberseguridad. Analizaremos la situación actual del IoT y procederemos a través de la evolución histórica de las botnets. Más tarde, analizaremos la situación actual de las empresas y su punto de vista en materia de seguridad de IoT.

Profundizaremos el mundo de las botnets analizando su arquitectura, protocolos y funcionamiento. Después, hablaremos sobre el desarrollo de una botnet avanzada al elegir la mejor arquitectura y el sistema de comunicación, además de proceder a su implementación y despliegue. Posteriormente, haremos un análisis de las medidas y contramedidas que podrían usarse para prevenir las botnets en nuestros sistemas. Al mismo tiempo, analizaremos el desarrollo de las técnicas de infección, evasión y resiliencia para que podamos preparar medidas de contingencia u contramedidas.

Finalmente, procederemos a analizar los servicios que prestan las botnets y cuál es el impacto en las distintas entidades.

Palabras clave: Seguridad, Internet de las cosas, Botnet, Sistema distribuido, Servicios del malware, Contramedidas, Detección.

Glossary

Smart City: a place where traditional networks and services are made more efficient with the use of digital and telecommunication technologies for the benefit of its inhabitants and business.

Industry 4.0: the age in which scientific and technological breakthroughs are disrupting industries, blurring geographical boundaries, challenging existing regulatory frameworks, and even redefining what it means to be human.

Internet Of Things: the connection of devices to the Internet. Other than typical fares such as computers and smartphones.

Smart Environment: the idea to build an environment with embedded sensors, displays, and computing devices so users can better understand and control the environment.

Botnet: a network of infected devices coordinated together to perform a task.

Bot: an infected device which is a member of an infected network.

Common Vulnerabilities and Exposures(CVE): list of information security vulnerabilities and exposures that aims to provide unique common names for publicly known problems.

Social engineering: a term used for a broad range of malicious activities accomplished through human interactions to obtain confidential information.

Hashing: to generate a value or values from a string of text using a mathematical function. Usually, it is a one-way function used to check data integrity.

Index

1. Introduction

On the last decades, we have experienced great technological advances thanks to the miniaturization of systems, communications, and power consumption improvements allowing an exponential increase in the number of devices connected to the Internet.

The processing capacity of modern microprocessors together with the pursuit and improvement of automatization have created new business lines, services, technologies and terms, such as Big Data, Smart Cities, Industry 4.0, Artificial Intelligence, etc.

Around all this terminology, we find the term **Internet of Things**, which help us to make our daily tasks easier by the use of data gathering, making an exhaustive analysis through data modeling and improving our environment by the use of sensors, creating what we define as **Smart Environments**.

The pursuit of these smart environments and their high demand in the market has produced the situation where a large number of technology producers, seeking a high economic income, have created devices with critical vulnerabilities generated mainly by design, implementation or configuration errors.

Those vulnerabilities have allowed cybercriminals to get new attack vectors and to generate new threats by making easier to attain illegitimate control of those devices and generating huge infected networks defined as **Botnets**.

Botnets allow performing attacks to different entities in a short period of time causing immense economic losses due to denial of service, device infections, ransomware or even data exfiltration.

In this project, we will make an approach on how Internet of Things botnets are used in real life and we will go through a close evolution of them. At the same time, we will make the design and development of a botnet, analyzing the impact on the different kind of entities and the services that they can provide.

We will also go through the detection measures and countermeasures employed by entities to avoid getting botnets or any kind of malware in their networks and how cybercriminals have improved their infection, evasion and resilience techniques to avoid these measures.

Finally, we will provide a countermeasure proposal for the detection and mitigation of botnets.

1.1. Final project objectives

Nowadays numerous botnets have been made and their use has been increasing during the last years, having its greatest growth after the publication of the source code of the botnet Mirai. However, the vast majority of the botnets detected so far, do not employ either complex or advanced techniques.

The objectives of this project are the development in detail of an advanced botnet through the analysis of the architecture and operation of botnets discovered so far. For a better approach to the real world, we will provide our botnet with the usual services such as Spam, Denial of Service, Ransomware, etc. We will also determine their possible evolution in terms of ways of evasion and persistence on the systems and provide detection measures, countermeasures for these techniques.

Moreover, raising awareness of the importance of security in all phases of a product life cycle.

1.2. Document structure

- **"State of Art" chapter.** It begins with an introduction to the Internet of Things and afterward, we take a look at the historical evolution of Botnets, as well as highlighting the importance of awareness. An analysis of botnets design, architecture, operational and services is made taking special attention in the different ways they get economic income. At the same time, an analysis of the impact on the different kind of entities with the actual legal framework is made, putting special attention to the General Data Protection Regulation.

- **Architecture proposal chapter.** After an analysis of botnets design, architecture, operational and services, it explains in detail the development and implementation to be carried out for a distributed botnet.

- **Deployment proposal chapter.** Having already defined our scenario, we will proceed to deploy the system by making a proof of concept and analyzing the results. Likewise, we will explain new ways of detection and prevention that are being used and new ways of infection, evasion, and resilience that cybercriminals could use.

.

2. State of Art

In this chapter, an introduction to the Internet of Things market is made. Likewise, we will go through the evolution of botnets until the moment of writing this project. Afterward, we will explain how botnets works and the services they provide.

2.1. Internet of Things market

The growing interest of consumers in Smart environments has generated a wide range of new products and services in different industries, such as home, automobile, retail, health, industry, infrastructure, etc.

These industries have generated high expectations in relation with the evolution of this market due to the new possibilities that Internet of Things provides, such as better use of the resources or the offer of new services such as Smart grids, Autonomous driving, wearables, Augmented Reality, Patients monitorization, etc.

According to [1] and [2], around 500 billion devices will be connected to the Internet by 2030, generating incomes up to 1.1 trillion dollars for 2025. Being these incomes, the main reason for the lack of security on the design of the devices.

However, this lack of security is not only presented on the design. The pursuit of having IoT devices with a long battery life that can last years have produced long periods to perform maintenance which is done remotely if possible. Moreover, IoT devices that can receive patches takes too long for being updated. In consequence, firmware and software decay is produced, generating bugs and issues that can not be solved unless having physical access or by a complete replacement of the device. Other problems such as hardware issues are being addressed using artificial intelligence known as predictive maintenance providing cost reduction.

Although the situation is not favorable, some tools to address these problems are being used such as Architecture Analysis and Design Language (AADL) which tries to address the design problems. In relation to the firmware and software updates, there are not many ways to deal with this problem other than increasing the budget for maintenance or providing the software update to end users, allowing them to update their own firmware.

Now, we will proceed to introduce some of the events with the highest impact and to describe the evolution of botnets.

2.2. Internet of Things Botnets history

Following the timeline on [3], and analyzing [4], [5], [6], [7], [8], [9] we can observe the evolution of some of the botnets that have generated a higher impact and their consequences:

- **August 2001:** One of the first botnets was “Kaiten / Knight"[4], based on the distributed attack tool called “Knight” which makes use of IRC channels for communication. It took advantage of the default setup of the password from the user “sa” that was initialized with a null value. It affected Microsoft SQL Server and Microsoft Data engine.

- **Early 2005:** The first samples of RBOT Spybot are detected [5]. RBOT is a worm that stores itself on memory and propagates through shared networks, trying to access to remote devices by the use of a list of users and passwords. In addition, it takes advantage of the vulnerabilities CVE-2003-0352, CVE-2003-0003 and CVE-2003-0109 for propagating to other devices. It has backdoor capabilities and uses IRC channels for communication, being also able to delete processes related to antiviruses or firewalls.

- **2007:** ZLOB [6] is a malware that tries to hide itself by pretending to be a video codec. Its main function is changing the DNS configuration, allowing the redirection of the victims to malicious sites.

- **2009:** the first appearance of botnets that direct their attacks to network devices such as routers, modems, etc. They try to access the device by the use of brute force and known exploits. These botnets are the first ones to make DDoS attacks making use of devices on the network and the first ones on attacking services from other devices using brute force. Botnets standing out are psyb0t, ChuckNorris y Aidra. [7]

- **2010:** Stuxnet [8], it represents a great change in the world of cybersecurity since it was an attack directed to industrial control systems. It stands out due to its process of infection of programmable logic controllers made through the interception and modification of the code. It is considered the first cyber weapon and was developed by Israel and USA with the purpose of stopping the Iranian nuclear program.

- **2012:** Carna Botnet [9] was created by a hacker in order to map the entire Internet for research purposes. For doing this, a Nmap script and a list of default credentials of the telnet service was used. He obtained a map composed of 460 million IPs.

- **2014:** An increase in the number of botnets is detected, but the first botnets with added functionalities such as crypto mining or ransomware appear. Although they are active for a short period (2 months), the botnets standing out during this period are SynoLocker [10] and Synology Dogecoin [11]. Both target NAS servers from the brand Synology, they were taking advantage of a vulnerability in the company's own Operative System.

  In one hand, SynoLocker is ransomware similar to CryptoLocker. In the other hand, Synology Dogecoin is a Dogecoin mining program which employed the CPU. These vulnerabilities were resolved by publishing a security patch. However, the attacker was able to make a profit of around $500.000.

- **2015:** In this period it is worth standing out two botnets with new features. The first one is Moose [12], it stands out for its mode of operation since is not persistent and focuses on generating followers to the attacker's account for a future sale. Secondly, Umbreon [13] is a rootkit characterized by the fact that it can be executed at the user level and for its evasion techniques, easily infecting other devices on the network.

- **2016:** On this year, we must highlight two botnets Mirai [14] and Hajime [15]. In the case of Mirai, it does not stand out greatly due to its infection process, but for the great impact on large technology businesses. It was the cause of the largest denial of service in history. This DoS was made against Dyn servers by using IoT devices, resulting in the inaccessibility to the websites of their customers such as Paypal, Twitter, Amazon, Github, Airbnb, etc. In this same year, the author from Mirai Botnet made its source code public.

  In the case of Hajime, it stands out due to the fact that the real purpose of the botnet is unknown. It make use of an own P2P network for communications and unlike other botnets, it has not been seen any malicious behavior coming from it yet. However, what causes more fear from this botnet is that it continues in development and getting improved.

- **2017:** It starts appearing more evolved botnets with other goals such as Brickerbot [16] and IoT Reaper [17]. The former, unlike other botnets that seek to control the largest number of devices, has the purpose of leaving the infected device unusable, forcing the user to make a complete replacement. The latter makes use of Mirai source code. However, it is mainly highlighted by the employment of nine vulnerabilities from different manufacturers to gain access to the device and to update the code in post-infection phases if it is required.

- **2018:** The appearance of more advanced and complex botnets continues, being able to perform all the functionalities presented until now. Standing out VPNFilter [18] and HNS [19].

VPNFilter aims to infect SCADA systems and its detection has been mostly seen in Ukraine. Like its predecessors, it infects the devices through the use of known vulnerabilities and at the same time has the ability to perform MitM attacks.
However, in addition to these functionalities, it can make infected NAS and routers useless something similar to Brickerbot. The most remarkable thing is the elimination of its trace from the infected equipment before leaving it unusable.

HNS is a botnet that has not had a great impact in terms of criminal activities and is believed to be in development since it lacks a denial of service module which is very characteristic of botnets. However, it has great relevance since it supposes a radical change in the operation of the current botnets, adding persistence to the infected device.

There has been a huge development in the services provided by botnets. However, we can observe the tendency is slowly evolving towards a Botnet-as-a-service providing the whole infrastructure, updates, and maintenance of the network and allowing cybercriminals to rent or sell their botnet to other malicious actors.

2.3. Current situation

In [20], we can observe a survey made to 950 IT businesses from the most computerized countries in the world, obtaining the following results:

- 48% of the businesses can detect if their IoT devices have suffered a security breach.
- 71% of the businesses makes use of two basic security aspects, which are encryption and system authentication. However, just 59% encrypts all the traffic from their IoT devices.
- There is an increase of 2.08% of the companies that consider IoT security in their budgets, staying at 13.15%.
- 57% of IoT device providers adopt a "security from design" point of view.
- 41% of IoT device or services providers try to guarantee greater data privacy.
- 91% of companies consider employing Blockchain technology as a security measure in their IoT devices. However, just 52% of them seriously consider its implementation.
- 46% of businesses believe their customers consider security as an important factor.
- 95% of companies are associated with cloud service providers, IoT specialists, IoT service providers to keep information safe at all times and situation.

- 36% of the companies claims having full control of the information at all times and all places.

Although is a new technology, if we take a look at these statistics, we can observe that the security measures employed from an information security point of view are insufficient and requires greater adoption, being evident the lack of commitment of the companies. Therefore, the preference of companies is the transfer of risk to third parties being those, cloud service providers.

Other relevance information coming from this survey are the following:

- The main difficulties that companies face when securing IoT devices are to guarantee data privacy, the large amount of information that is being collected, balance security with the user experience, access authentication, etc.

- 95% of the businesses consider the need for regulation regarding the way to secure IoT environments. At the same time, 57% of them, consider it of extreme importance.

- Around 50% considers that such regulation should contemplate matters such as who is responsible for securing the data in each state, the security methods that should be used to store the information, the implications of non-compliance, how to deal with the great amount of data, etc.

As we can see, although in the European Union there is the General Data Protection Regulation (GDPR), there is great ignorance about how to proceed and how the information should be treated, since the number of devices to handle is overwhelming. These ignorance causes fear in the management of these devices to the different professionals of the IT sector.

2.4. Botnets Architecture

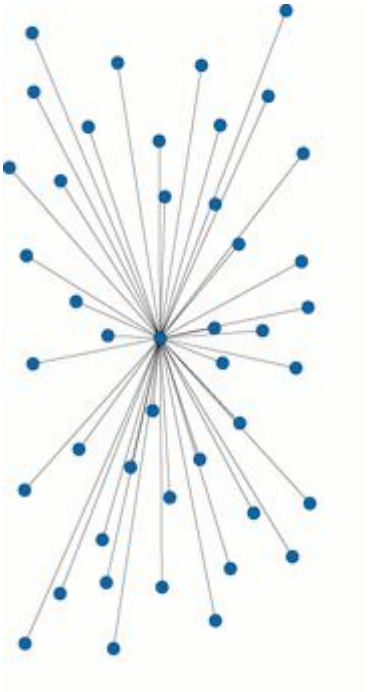
Next, an explanation of the possible Botnet architectures and a classification according to them will be made, providing the advantages and disadvantages of each architecture. In the same way, we will explain its protocols, usual operation, and services they provide while making emphasis on their impact on businesses.

2.4.1. Types of networks

Botnets architecture as any computer network can follow any of the following topologies:

a) Centralized
b) Decentralized
c) Distributed

a) Centralized

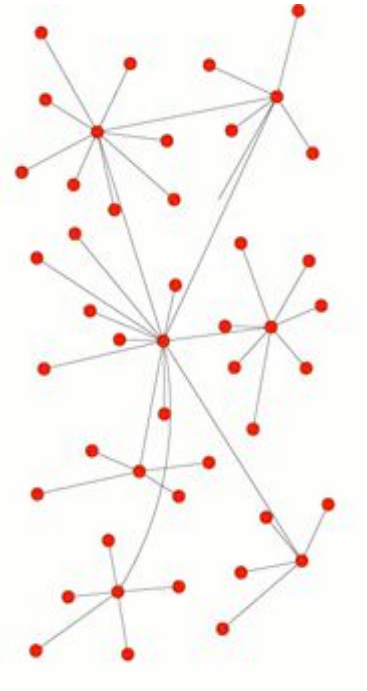


These kind of networks are the simplest to perform and the most used. Its structure is composed by a fixed main node (Example: servers, Command /Control (C&C)), which handles and validates other nodes communications (Example: clients, bots, etc), which means it allows the communications between nodes. An example would be a router. The main benefits of these networks are easy deployment and low latency.

The use of this type of networks has disadvantages such as low scalability and robustness. In consequence, it has low resilience because to disable the complete system or service it would be just necessary the disruption of the main node.
Its low scalability is due to the reason that a huge increase in the number of nodes could cause the congestion of the main node.

From an information security point of view, protecting such a network may seem easier since it reduces to protect a single device (main node). However, the fortification of this device is never perfect, so there is always a high-risk factor of service interruption.

b) Decentralized

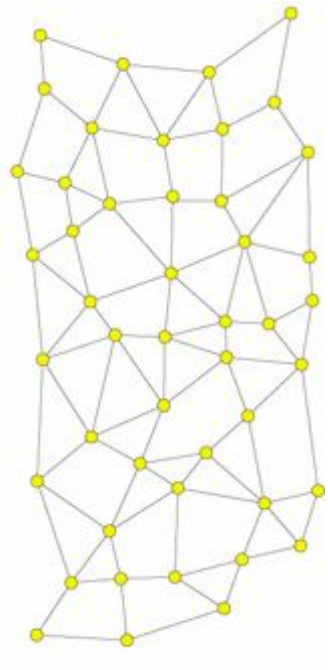


This kind of networks could be seen as a combination of centralized networks where each fixed main node operates independently, which means other main nodes do not need to know the information of other nodes.

Compared to centralized networks, these networks are also easy to be done. In the same way, they keep a low latency but it has greater robustness and resilience cause a disruption of a main node does not disable the complete network. At the same time, this allows managing a greater number of devices.

From a security point of view, these types of networks are more complex since all the information and connections are no longer controlled by a single element. Although the attack surface increases, the violation of a main node does not incur the violation of the complete network as it happens in centralized systems.

c) Distributed



These networks compare to the previous networks, they do not require a fixed main node to communicate with other nodes.

On these type of networks, any node has the possibility to be a main node. An example of this kind of network is Peer-to-Peer networks or Cloud Computing networks.

Other advantages are high robustness and resilience because to take down the whole network, all nodes must be taken down first. Another advantage is its high scalability due to the fact that each node just communicates with its neighbors.

However, these networks have also great disadvantages. Firstly, the latency of the whole network can be high because no node knows the complete size of the network and the communications are made node to node.

Secondly, these networks have a problem by default, the CAP theorem [21]. The theorem establishes that a distributed system can just achieve simultaneously two of the next three conditions:

- Consistency: all nodes see the same information at all time.
- Availability: to guarantee that all request made by a node will receive a response, being satisfactory or not.
- Partition tolerance: the system continues working although some nodes have been shut down.

From a security point of view, it presents a high risk because a high level of trust is granted to each node.

Network	Latency	Scalability	Robustness / Resilience	Deployment difficulty
Centralized	Low	Low	Low	Low
Decentralized	Low	Medium	Medium	Medium
Distributed	Medium	High	High	High

Table 1.Types of networks. Summary.

2.4.2. Protocols

The most common communication protocols employed by botnets [22] are the following:

1. **IRC:** real-time communication protocol based on text. It makes use of a centralized system for message delivery through the use of channels. These channels can be protected by the use of a password, preventing other users from taking control of the channel. Likewise, file transfer is supported, allowing the distribution of binaries, configuration files, and updates.

2. **HTTP:** protocol designed to transfer web pages content from servers to browsers. The communication is made by request of the client and the server response, so it does not allow group communication.

   By the use of this protocol, a P2P system can be made if each device implements an HTTP server and HTTP client.

   Being HTTP the more usual traffic on the Internet, the main benefit of using this protocol is reducing the probability of being detected.

3. **SMB:** a protocol developed by IBM and later by Microsoft. It is used for file and network resources sharing. The protocol is based on request-response and allows the use of a previous authentication system.

   It is used by botnets to perform passive infection, hosting malicious files for subsequent execution by the user.

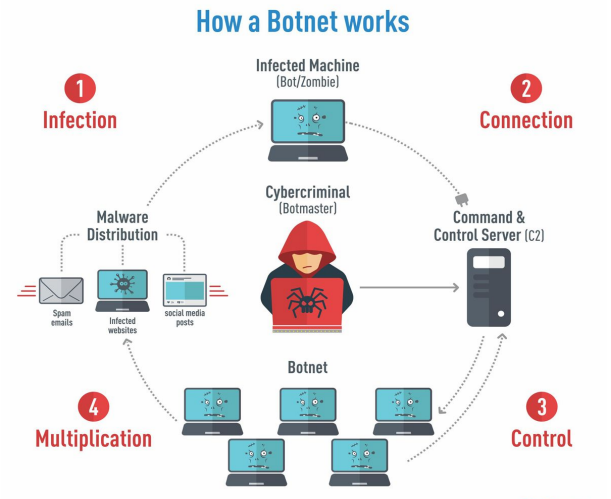
4. **P2P:** protocols developed for file sharing and collaboration reasons. Botnets usually use this type of protocols to create new P2P networks or to hide part of their network within another P2P network. Making more difficult the traceability of the communications.

5. **UDP y TCP:** they are protocols used by the transport layer of the OSI model. The former is characterized as a not oriented to connection protocol, without flow control and without acknowledgment. The latter is characterized by the opposite being an oriented to connection protocol, with flow control, and with acknowledgment.

   The headers of both protocols are used by some botnets to communicate with different devices by using the optional fields.

2.4.3. Operational

Now, we will proceed to explain the usual operation of botnets which is composed of four phases [22]:



1) **Infection or propagation:** Usually cybercriminals send out malware to infect devices by the use of spam, social engineering or shortened malicious links.

   With user interaction, this malware will be downloaded and executed. Afterward, it will try to exploit the device and try to attain root privileges, gaining the whole control of the device. The process defined above is called **passive infection.**

   Another way of infection is an **active infection,** where the bots try to infect other devices by the use of network scanners, brute force techniques, existing vulnerabilities, etc.

2) **Connection:** After completion of the previous phase, the device will try to connect to the Command and Control Server (C&C). If the connection is made, the device will stay idle and waiting for the cybercriminal commands being this difficult to detect.

   Usually, the communication channels used at this stage are encrypted in order to achieve higher privacy, so it increases the difficulty for researchers to obtain a bot executable for reverse engineering.

3) **Control:** the owner of the botnet will wait until a number of infected devices is reached and also to have a purpose for the botnet. After achieving these requirements, the cybercriminal will send the commands to each device (bot) through the C&C server.

4) **Multiplication:** Meanwhile, the cybercriminal will be focussed on recruiting more devices to expand out the botnet by the use of previously infected devices or new ones.

2.4.4. Botnet services and impact on entities

In this section, we will explain in detail the most common services a botnet offers and their impact on the different entities.

2.4.4.1. Spam and phishing

The term spam is considered as anonymous and unsolicited bulk email, while phishing means stealing confidential data from a person's computer to steal their money by the use of the previous data.

Spam and phishing scams are the main entry point of infection of businesses having a higher impact in companies or entities where their employees have not been taught about security or how to recognize malicious emails.

Attackers usually by the use of social engineering sends spam and phishing scams to different members of a company with an attachment containing malware and being the topic of the email something personal or financial. So the victim worried about it, clicks on the attachment executing the malware.

Another possible scenario, instead of installing malware on the victim's device is redirecting the victim to a phishing website, which is a cloned website from a legitimate one, where a username and a password are asked to steal their credentials. Afterward, these credentials are employed in different services trying to obtain more benefit from the attack.

After a successful attack, the attacker will send spam and phishing scams to any new email found by using the victim's email, trying to spread to other networks.

2.4.4.2. Denial of service

A DoS attack is designed to hinder or to stop the normal operation of a website, server or other network resources. It involves manipulating the way incoming data is handled by the resource to overload it, preventing it from carrying out its usual functions and in some circumstances producing server crashes.

The consequences of a denial of service [38] may incur in technical support costs, expenses to normal operation, loss of users productivity, damage of IT assets, etc. These events can produce great economic losses of around $1.5 million. Being the most important, the loss of service and the reputation damage which affects future income from investors.

In reference to the Healthcare sector, healthcare institutions are starting to get more technological using computers to schedule patients meetings, store patients electronic medical records, schedule operations, store patients recipes and many other services. At the same time, most of the healthcare institutions make use of not updated software and systems making them the perfect target of this kind of attack.

Although GDPR [40] offers a guideline and what are the means of protection required to correctly treat the information of the different kinds of personal data, many healthcare businesses are unable to implement or afford such a system. For these reasons, DoS attacks have an enormous impact in this sector due to the critical service they offer, which produces not only economic losses but also puts in risk people's life.

2.4.4.3. Data exfiltration

Data exfiltration occurs when an unauthorized data transfer is carried out from a computer by malware or a malicious actor. The data is usually published publicly or sold in the dark web containing some important information such as network architecture, projects, users credentials, customers and business emails, stakeholders and other personal or professional information. Not less important, this has a higher repercussion which is losing trust from customers and stakeholders producing great losses.

After the occurrence of the data breach, a forensic investigation is made taking around three months to determine the scope, which means a full recovery takes a long period of time.

In the case personal data is involved on the data breach, we need to consider GDPR [40] which is especially focused on how personal data from an EU citizen must be collected, employed and protected regardless of the geographical area. Depending on the measures employed to safeguard the information, the EU can sanction businesses with high fines if the necessary protection measures were not placed correctly or were insufficient. The less severe infringements could result in a fine of up to €10 million, or 2% of the firm's worldwide annual revenue from the preceding financial year, whichever amount is higher. In case of getting fine twice, the infringement could result in the double of the first fine being the result €20 million, or 4% of the firm's worldwide annual revenue from the preceding financial year.

Another important thing implemented by GDPR [40] is data breach notification which establishes the mandatory requirement to notify the relevant supervisory authority within 72 hours of becoming aware of the breach and also notifying the individuals affected by such breach.

Data exfiltration can be made using a communication channel or a storage device.

2.4.4.4. Ransomware

Ransomware is a malicious program designed to extort money by preventing access to the computer or encrypting the data stored on it. It usually displays a message offering to restore the system in return for a payment providing hope to the victim so they can get back the data. The message states that the system has been blocked or encrypted because the victim is running unlicensed software or has accessed illegal content so they must pay a fine.

As spam, ransomware main entry point is an email or by the use of an infected Universal Serial Bus (USB). Ransomware, as described, encrypts any data stored being in most cases impossible to recover the decryption key.

The worst case for a business is if a server gets hit leaving it unusable provoking, in consequence, a denial of service. Being hit by ransomware is probably worse than a DoS, due to the fact, the information that has not been backed up gets completely lost in addition to the reputation damage.

2.4.4.5. Cryptojacking

Cryptojacking is mining cryptocurrency without the owner's knowledge by the use of compromised devices. Mining can be performed either by installing a malicious program or by fileless malware such as web pages ads or scripts, captchas, etc.

Depending on the CPU/GPU power employed by the mining program, it may affect the system producing in most of the cases unusual waiting times. The worst case would be the use of the whole CPU/GPU, producing slowdowns and unexpected shutdowns.

3. Architecture proposal

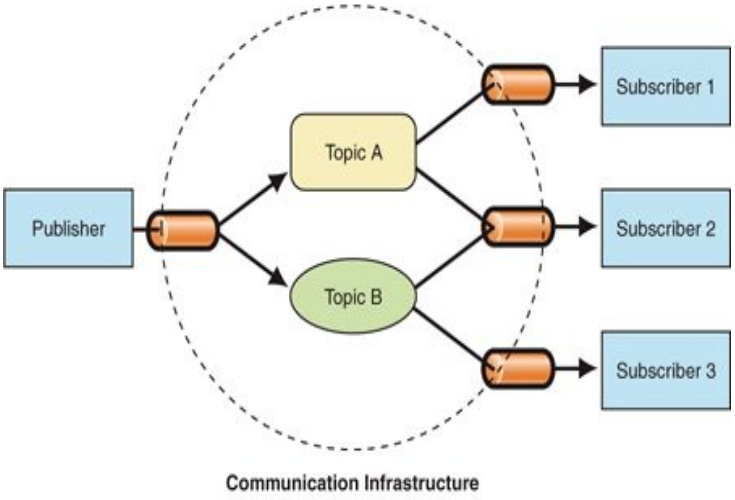
With the different aspects discussed in the previous sections, we will proceed to specify the requirements of our advanced botnet. Also, we will explain in detail the different tools to be used to achieve these requirements and how they operate.

3.1. Aspects and selection

We are looking for a high availability system which offers greater independence to the devices and high resilience, allowing us to maintain the system with the minimum number of devices causing greater difficulty to disrupt the entire network. On the other hand, we look for an easy and simple communication system.

For these reasons, our advanced botnet will be based on a distributed system and will employ a communication system based on Pubsub.

PubSub [23] is a communication pattern where an information provider (publisher) supplies information about a topic to different consumers (subscriber). Subscribers decide to what topics they want to subscribe and then wait to receive that information.



Some advantages of this mean of communication are:

- It facilitates asynchronous workflows.
- It decouples subsystems that still need to communicate. Subsystems can be managed independently, and messages can be properly managed even if one or more receivers are offline.
- It increases scalability and improves the responsiveness of the sender.

To achieve the conditions mentioned above, we will make use of IPFS. **IPFS** [24], called as Interplanetary File System, is a protocol and a network designed to create a distributed system to store and allow access to files, web pages, applications and data through the use of storage addressed by content. It was designed by Juan Benet and is an open source project that still in development being in alpha version.

We will now proceed to explain how IPFS works.

IPFS makes use of different implementations [25] to achieve a complete distributed communication and storage between peers. IPFS is divided into five layers:

- Naming
- Merkle Dag
- Exchange
- Routing
- Network

- **Network layer** provides point-to-point transports (reliable and unreliable) between any two IPFS nodes in the network. It handles:

  - NAT traversal such as Hole punching, port mapping, and relay.
  - supports multiple transport protocols such as TCP, SCTP, UTP, etc.
  - supports encryption, signing or clear communications
  - Multi-multiplexes such as multiplexes connections, streams, protocols, peers, etc.

- **Routing layer** is used for finding peers and data in the IPFS network by addressing, instead of searching. There are different kinds of implementations:

  - **Distributed Hash Tables (DHT)** are used as a semi-persistent routing record distributed cache. This DHT contains a (key, value) pairs, where the key is the content hash and the value are the Peer IDs which can provide the content. The content and Peer IDs hashes are uniquely generated which provide us with the ability to identify each node and guarantee data has not been modified. The use of DHT provides autonomy, decentralization, fault tolerant and scalability.
  - **mdns** which is a local discovery service.
  - **DNS** which is used nowadays on the Internet.

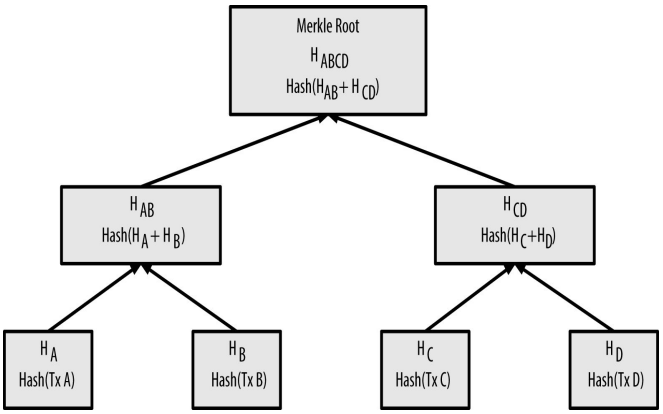
- **Exchange layer** takes care of negotiating bulk data transfers. It governs how the transfer of content-addressed blocks occurs and it has various kind of implementations:
  - **Bitswap** is a message based protocol and is the main protocol for exchanging data. It is a generalization of BitTorrent to work with arbitrary DAGs. It manages to request and to send blocks to and from other peers in the network.
  - **HTTP** can be implemented with HTTP clients and servers.

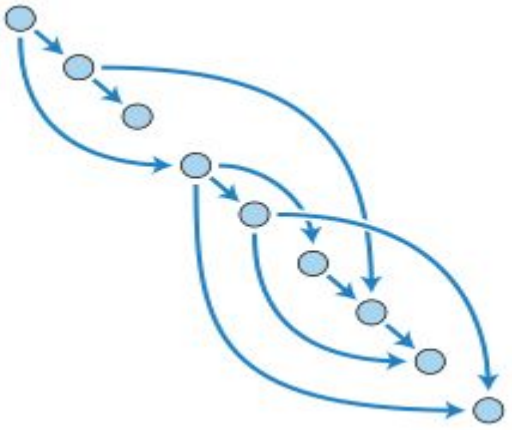
The three layers explained above are implemented by the use of libp2p library [26] and make possible data transfer.

- **MerkleDAG layer** is the data structure at the heart of IPFS. It is a Merkle directed acyclic graph whose edges are hashes.

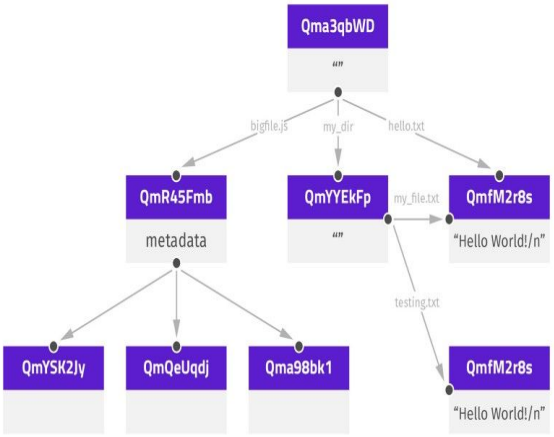
  A Merkle tree is a binary tree in which every leaf node is labeled with the hash of a data block, and every non-leaf node is labeled with the cryptographic hash of the labels of its child nodes.

  A Directed Acyclic Graph (DAG) is a graph without cycles, which means a node is never encountered twice, and where the connections between the nodes have a direction.





A MerkleDAG would have the next structure:



- Every IPFS Object is a Merkle DAG data structure which contains data and an array of content addressed links.

- Merkle DAG gives us the ability to have a **versioned file system,** like Git. Being able to see a history or how data has been modified over time and to link to any one of those versions.

This kind of data structure is implemented in IPFS by the use of IPLD (Interplanetary Linked Data) [27]. IPLD is a data model of the content-addressable web. It allows treating all hash-linked data structures as subsets of a unified information space, unifying all data models that link data with hashes as instances of IPLD, such as Ethereum, Bitcoin or Git, allowing communication between them.

- **Naming layer (IPNS)** handles the creation of mutable pointers to objects and human-readable names. This is important due to the nature of content-addressed data which is immutable meaning that changing an object would change its hash and in consequence, it would produce a different object.

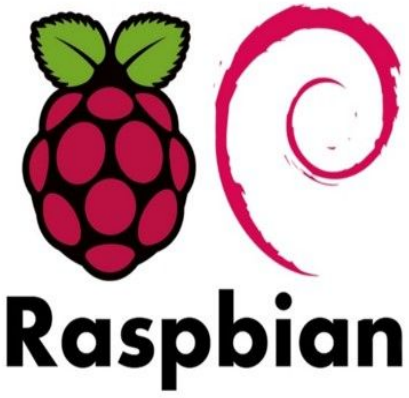
  IPNS is based on Self-certifying File System (SFS) [28], which is a PKI namespace where a name is the hash of a public key providing us with mutable namespaces. Whoever controls the private key controls the name. At the same time, Records are signed by the private key and distributed anywhere.

In reference to the hardware, we will make use of two Raspberry Pi with Raspbian and they will act as our IoT devices. The reason for choosing Raspberry Pi is due to its huge community, documentation and easy use.

Raspberry Pi [29] is a low-cost single board computer of the size of a credit card. It makes use of computer monitors or TVs, standard keyboard and mouse being capable of doing daily tasks as a desktop computer would do, such as browsing the internet, play high-definition video, word-processing, etc. At the same time, these kinds of devices are often used for IoT or home automation projects due to its cost, usability, the allowance of adding hardware peripherals such as sensors or board extensions and the existence of an operating system developed specifically for this board, called Raspbian.

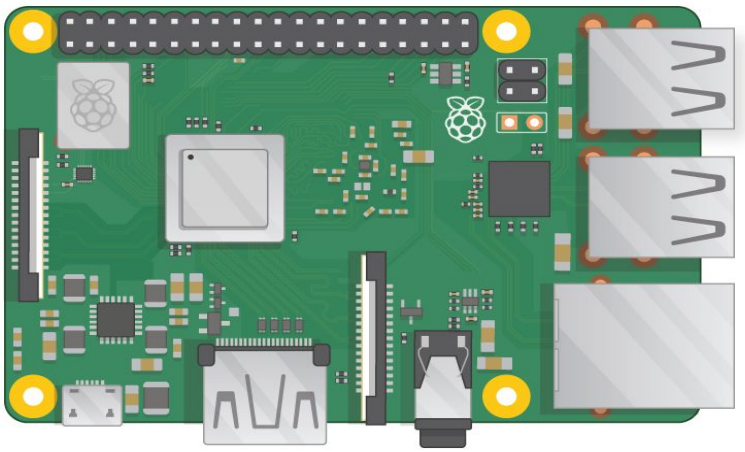
Raspberry Pi was developed by the Raspberry Pi Foundation which is a UK-based charity that works to put the power of computing and digital making into the hands of people all over the world. So, more people are able to learn about technology.

Raspbian [30] is a free operating system based on Debian optimized for the Raspberry Pi hardware. An operating system is the set of basic programs and utilities that makes Raspberry Pi run.



Raspbian was created by Mike Thompson and Peter Green as an independent project of the Raspberry Pi Foundation. Nowadays, Raspbian also receives the support of Raspberry Pi community members who wish to get the maximum performance from their device.

Raspberry Pi has different models with different hardware specifications, in our case, we will make use of two different models, model 2 B+ and model 3, being the specifications of both models put in detail below. The reason for choosing these two models is due to the software requirements and availability.

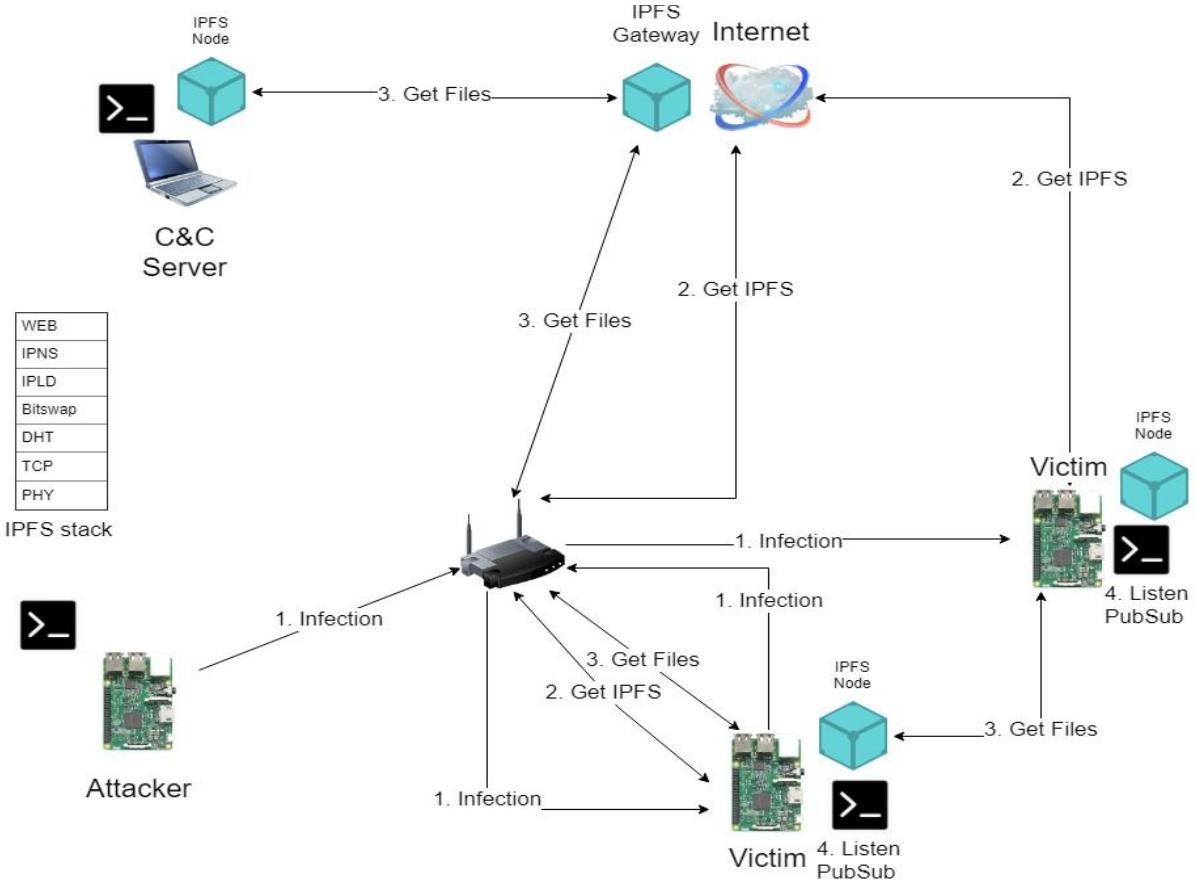


	Raspberry Pi	
	Model 1 B	**Model 2 B**
CPU	ARM1176JZF-S (700 MHz)	Quad Core Cortex A7 (900 MHz)
RAM	512 MB	1 GB
WIFI / Bluetooth on board	No	No
Peripherals	100 Base Ethernet, USB ports, GPIO pins, HDMI port, CSI interface, DSI interface, Micro SD card slot	

Table 2. Raspberry Pi specifications. [31]

3.2. Scenario description

We will now proceed to explain in detail the scenario for this project. Our scenario will be formed by two Raspberry Pi, a router and a laptop that will act as our C&C server and as an IPFS node, being this our initial IoT Botnet setup. Our scenario and process will end being similar to the image below:



Firstly, we will proceed to explain the initial setup included in each device:

- C&C server will be set up with an IPFS node storing all the needed files which are a text file, a pdf file and a shell script for its posterior download. Afterward, it will be put into the PubSub channel [37] which is used as our communication channel with the other devices.
- Attacker's device will be set up with the shell script, system packages sshpass [32] , mutt [33], and it will be the one performing the attack.
- Victim's device will be a clean setup, being just connected to the same network.

Our shell script has been implemented with six functions (infection, encryption, spam, data exfiltration, denial of service and to brick a device) which are explained in the next chapter.

The process to obtain the final scenario represented above is by doing the next steps:

1) Attacker's Raspberry Pi by running the shell script infection module, it will proceed to do a network scan and afterward proceed to try to access SSH service. Once successful access has been made, the script will download IPFS, mutt, sshpass, and awk [41] install them on the device.
2) After IPFS has been installed, it will proceed to download the files needed through the IPFS network from our C&C IPFS node.
3) We set up the infected device as a malware delivery IPFS node.
4) We connect the infected to the PubSub channel, waiting for new commands.
5) We just need to send any available command in our shell script to the PubSub channel for being executed by all devices.

4. Deployment proposal

We will proceed to deploy the previous scenario by making a proof of concept (PoC), analyze the results and provide possible detection measures and countermeasures. At the same time, we will analyze the development of infection, evasion and resilience techniques employed by these systems.

4.1. Proof of concept

In this section, we will explain the operation from the different modules implemented in our advanced botnet for the realization of the different services. We have to notice the IPFS version employed in this project is go-ipfs v.0.4.20. The whole code and required files can be found here [**Github**].

As explained above our shell script is composed of six modules:

a) Infection module will be in charge of looking for vulnerable devices and to complete the setup by downloading and installing the required software and files. This phase is divided into the next steps and important commands:
   1) Attacker's device will perform a network scan:
      a) Check the network interfaces available and get **subnet /24** by the use of **"ifconfig"**, **"grep"**, and **"sed"** command.
      b) Check available devices by the use of **"ping"** command.
      c) Check port availability through **/dev/tcp/<ip>/<port>** which is incorporated in Linux File System.
   2) If port number 22 is available, which is associated with SSH service. It will try to connect to it by the use of **"sshpass"** command and try to execute the command passed as an argument.

a) **sshpass -p <password> ssh <username>@<ip> -o StrictHostKeyChecking=no '<commands>'**, StrictHostKeyChecking is disable to avoid host verification.

3) The following steps are to set up the IPFS node and retrieve the required files. This is made by the use of **'wget'**, and **'ipfs'** commands. The ipfs commands used are:
   a) **ipfs init -** generates the configuration files and the required RSA key pair used as identification.
   b) **ipfs config -** we configure the number of connections, the period of time to keep them alive, routing protocol to be used, etc.
   c) **ipfs daemon --enable-pubsub-experiment & -** enables the use of the experimental pubsub functionality and runs it in background.
   d) **ipfs get -** downloads files from IPFS.
   e) **ipfs pin** - Pin objects to local storage. This command enables the possibility to be a provider of such files.
   f) **ipfs pubsub -** experimental publish-subscribe system on ipfs.

A more complete explanation is made in IPFS [24] under the subject IPFS CLI commands.

4) We configure a task to be run every minute which will be in charge of reading the commands sent from our C&C server by the use of **"crontab"**.

After completing this phase, we will be able to launch any command to the bots being these commands run in the background.

b) Encryption module will be used as our ransomware service. The command to run it is **"./botnet.sh encrypt <path>**, through the use of the directory passed as an argument, it generates a compressed file of all the subdirectories and files it contains by the use of **"tar"**, and it will try to make use of **OpenSSL** to encrypt the compressed file with AES-256-CBC with a password provided in a text file. Afterward, it will remove the subdirectories and files compressed, and add a text file with the ransomware rescue message.

c) Spam module will compose a mail with a spam message and a malicious attachment. By the use of the command "mutt" [33], we will send it to the recipient email by the use of Gmail SMTP server. An example of its use is:
./botnet.sh spam <recipient mail>

d) Data exfiltration module, in a similar way to encryption module, we generate a compressed file of all subdirectories and files of the path provided protected with a password. Afterward, we upload it to IPFS and communicate it to the C&C server. Use example: **./botnet exfiltrate <path>**

e) Denial of service module will attack the webpage provided as a parameter by the use of HTTP flood [39] by using GET requests.

HTTP flood is a type of Distributed Denial of Service attack which is based on sending legitimate HTTP or POST requests to attack a web server or application, trying to consume the maximum resources possible in response to each request. This attack is a volumetric attack which means it tries to exceed the number of requests the server or application can handle. Due to its nature, this kind of attack is significantly harder to detect and block.

An example of its use is: **./botnet ddos <webpage>**

f) The brick module will leave the device unusable by wiping out the root directory. It can be used by running **./botnet brick**

4.2. Results

Through the employment of IPFS and shell scripts, we have been able to establish a complete distributed botnet with the advantages of reducing our chances of being detected by antivirus software or web page filters being this possible by the use of trusted SSL certificates and by the nature of shell scripts which are used on a daily basis in Linux based systems.

However, the possibility of this scenario in the real world still far due to certain challenges that have been found during the development of this project, which is explained below:

a) Alpha version. IPFS is still in development which means many of the functions used are in the experimental phase and are unstable, might be subject to changes, deprecation or improvements.

b) Communication channel security. The PubSub channel, it neither has encryption or authentication so anyone would be able to enter and control the botnet, being also able to view all the messages in plain text.

c) Communication channel stability. Due to the P2P network nature of IPFS, peer finding is continuously running by default, dropping and making new connections every 30 seconds.

d) Protocols. We have made use of IPFS default routing protocol which is DHT and HTTPS. In the case of DHT, it takes a long time to complete files requests. However, this is something expected in P2P networks due to the peer discovery process.

e) Anonymity. In the current version, any node can be easily identified by their public and private IP, transport protocol and port used for IPFS.
Example: /ip4/127.0.0.1/udp/9090.

f) Easy Malware binary download. At the same time, IPFS provides us with a distributed system, it also provides an inconvenient which is every peer can download the binary, making possible to researchers to easily obtain a sample.

Following IPFS Requirement Roadmap [34], most of these problems will be solved in future implementations by adding Communication Channel authentication and encryption, support for I2P and TOR communications, peer and content routing anonymity and other security and network improvements.

4.3. Detection measures and countermeasures

After an analysis of the development, communication channels and patterns used, we can observe there are many different ways to effectively protect our networks from this kind of attacks.

The most common network security measures and countermeasures used by businesses worldwide are Artificial intelligent based Malware Detection, Firewalls, Network Monitoring, Cloud computing, and Sandboxing.

Most of the measures mentioned above are introduced by default in software such as Antivirus Software or Browsers, being this implement as a basic security measure.

We will now proceed to explain how these measures work and how they help us to protect our systems.

- Nowadays, traditional Antivirus Software based on signature, behavioral, heuristic and hash matching analysis [35] are not able to protect end users devices. For this reason, Artificial intelligent based Malware detection has been implemented, not just with the parameters mentioned above but also adding new parameters that can define malware to effectively protect systems from advanced threats and 0-day vulnerabilities.

- Network monitoring is a tool employed for the analysis of networks data flow to identify issues affecting the availability or functionality of network devices, providing a more in-depth view of our network communications.

- Cloud computing based security software is used to protect cloud-based services and applications, and to provide better performance to end-user devices reducing their workload, providing better management, real-time analysis, and updates. Eliminating the need to manually scan and update end-user devices.

- Sandboxing is applied in common applications such as Antivirus Software or browsers. A sandbox is a safe isolated environment that replicates an end-user operating environment, where to execute and observe application behavior. Making easier to identify malware due to their activity.

However, the best practices for securing our networks are Network segmentation, least privilege over antivirus software detection and Virtualization:

- Network segmentation means to split our network in small networks, providing us higher security in an architectural way, making possible to isolate each of the segments and to stop any malicious activity from spreading to the whole network, preventing access to sensitive information, etc.

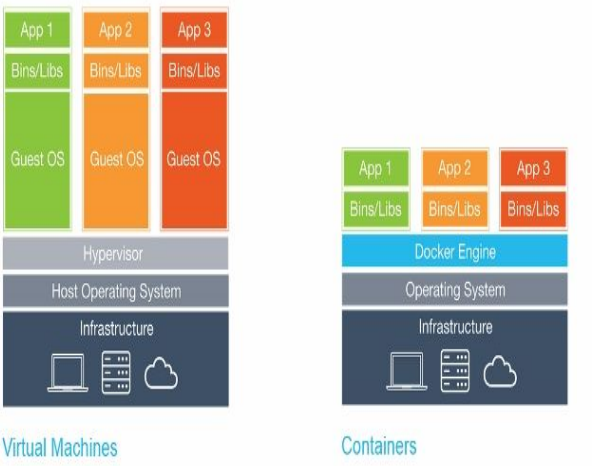
- Least privilege over antivirus software detection. Although antivirus software helps to protect systems, they mostly detect known malware or malware variants and other security measures should be taken. The principle of least privilege (PoLP) is the practice of limiting access rights for users to the bare minimum permissions needed to perform their work in order to reduce the attack surface of the organization.

  Under PoLP, users are granted permissions to read, write or execute only the files or resources they need to do their jobs. This kind of implementation also allows to restrict execution of unknown applications and have control over the installation of new ones.

  By applying this principle, we are able to protect systems from cyber attacks by shrinking the attack surface and to easily identify any suspicious activity related to application execution.

- Virtualization is the process of creating a software-based, or virtual, representation of something, such as applications, networks, servers, storage, etc. Nowadays, two different kinds of virtualization are used depending on requirements and costs, which are Virtual Machines (VM) and Containers.

  Virtual Machines [36] is an emulation of a complete computer system, hardware, and software. The Operating systems (OS) share the hardware resources from the server or servers available. Each VM requires its own underlying OS and a hypervisor which is software, firmware or hardware that creates and runs VMs.



  Containers [36] compare to VMs, they just virtualize OS. Each container shares the host OS kernel, binaries and libraries being these read-only. They are exceptionally lighter than VMs, and take just seconds to start, while VMs takes minutes to run and their size is much bigger. However, Containers are used to run specific programs while VMs are more versatile.

Virtualization provides the capability to analyze the behavior of malicious activities, quickly disrupt infected machines and quickly deploy a complete network in few seconds being all these systems isolated from each other.

Other tools, technologies, and practices used by businesses are:

- Software Defined Wireless Area Networks (SD-WAN) are a software-based WAN. It provides huge advantages such as resources and application optimization, simplified management, network segmentation and distributed security such as easily deploying of Firewalls, Intrusion Detection Systems, Intrusion Prevention Systems, Whitelisting and Blacklisting domains, etc.

- Code review and Bug Bounty programs are used by businesses to find the higher number of vulnerabilities or malfunctions in software, for preventing code decay and malicious attacks, to quickly provide a solution and keep their users safe.

- The creation of a cybersecurity department specialized in analyzing, attack simulation, protection of business IT infrastructure through the use of different security solutions such as the ones mentioned above and others such as Deep Packet Inspection, Security Information Event Management, Incident handling, defining secure network architectures, Honeypots, etc.

However, even if we are able to implement most of these measures, our efforts and work would be useless if we have not defined or updated our business security policies and procedures. At the same time, risk analysis should be performed to know the current status of the network. Being also especially important employees training making them able to easily identify basic and common malicious activities such as spam, phishing, social engineering, or suspicious files, due to the reason, they are the first line of defense and the main entry point of attacks.

4.4. Development of infection, evasion and resilience techniques

Cybercriminals trying to evade the security measures implemented by businesses and to improve their resilience in networks make use of different techniques and software improvements, being the most common ones, encoding, encryption, fast flux, cloud services, Polymorphism, 0-day vulnerabilities, Sandboxing and Virtualization, Antivirus analysis detection.

In order to avoid being detected by traditional antivirus software, basic and common techniques such as encoding, encryption, obfuscation are used.

- Encoding is the process of converting data into a specified format. Although it makes the text more difficult to read, anyone is able to read it if the correct format is found, there is no use of a password. It is used to evade signature-based detections. An example would be:

    - UTF - 8: This is a test message.
    - Base64: VGhpcyBpcyBhIHRlc3QgbWVzc2FnZS4=

- Encryption is encoding information in such a way that only authorized parties can access it. This access control is usually accomplished by implementing a password or key. An example of both types of encryption is SSL and AES-256. In the same way, as for encoding, it is used to evade signature-based detections, but keeping code safe from unwanted individuals. At the same time, it helps evade firewalls, IPS and IDS detections.

- Obfuscation is rendering an executable program or source code unreadable and hard to understand while keeping its functionality. It is used to bypass static based detections and to make researchers or analyst waste more time studying the program.

More advanced techniques are used to prevent being analyzed or being detected such as sandboxing, virtualization and antivirus detection, etc.

Advance malware tries to detect if they are being executed over a sandbox, virtualization system or being analyzed by antivirus software so they avoid executing themselves, acting as a benign file.

- In sandbox evasion, the malware keeps dormant until it detects user interactions and system interactions that a sandbox usually cannot replicate before execution, such as mouse actions or window scrolling, reboots, check installed programs, etc. An example of malware with these characteristics is Locky ransomware.

- For virtualization evasion, malware looks for specific properties which are characteristic of virtualization software such as registry keys values, certain processes, and services, virtual machine MAC address, memory structure, etc.

- For antivirus analysis evasion, cyber criminals make use of polymorphism or metamorphism to avoid pattern-matching detection. Polymorphic malware makes use of a polymorphic engine to mutate code while keeping the original algorithm intact. It is usually implemented with encryption providing a new encryptor/decryptor pair for each copy of the code.

  Metamorphic malware is the one that can change based on the ability to translate, edit and rewrite its own code. This kind of malware is difficult to write and to detect because the code is changed after a successful infection but keeps its original functionality.

In order to evade blacklisting, firewalls, IPS, and IDS and being able to download malware and communicate with the C&C control server, they make use of Fast Flux and Cloud Services.

- Cloud Services are the use of cloud computing to deploy and deliver a service over the Internet. To do so, cloud computing business such as Amazon Web Services or Microsoft Azure with each subscription provides an SSL certificate which is used by cybercriminals to avoid reputation based blacklisting and keeping communications secure.

- Fast flux is a DNS technique used to hide phishing and malware delivery sites behind an ever-changing network of compromised hosts by frequent and rapid changes of the IP addresses associated with a Fully Qualified Domain Name (FQDN). By using this technique, cybercriminals are able to hide their C&C server and do malware delivery by using the compromised hosts as proxies servers.

To keep resilience over a network, cybercriminals provides malware with self-replication capabilities, copying itself in case of being removed; backdoors, add itself to the list of programs to run on boot, modify antivirus rules to be considered as a benign file, etc.

Overall these techniques, the most effective way of infection, evasion and resilience is making use of unknown or unaddressed vulnerabilities called 0-day vulnerabilities, especially in IoT domain where devices take long periods for being updated or replaced.

5. Conclusions

5.1. Summary

The first steps of this project were to put in context the current situation of IoT security.
Next, looking at the evolution of IoT Botnet history and the impact they have produced in the different industries, we have seen the necessity of a regulation specifying how to achieve data privacy, including integrity and availability, a part of GDPR.

After watching this situation, we have analyzed botnets architecture which includes types of networks, protocols and operational used, to have a better understanding of these systems. At the same time, we have seen the services botnets provides analyzing the economic impact and damage they can produce in the different industries if we do not protect our systems correctly.

Afterward, we have explained the proposal of this project, trying to make a distributed high availability botnet to analyze the consequences it could produce, while putting in detail the different tools to be used to accomplish these requirements, specifying their use and operation. To prove the possibility of such a system, we define a scenario and make a proof of concept providing in detail its construction and operation.

Lastly, we analyze the results provided through the challenges found while developing the project and the future solutions that will be implemented to solve these issues, making possible a better and easier construction of this system. Later, we have explained the different detection measures and countermeasures that are being used or could be implemented to protect our devices. At the same time, we have defined several infection, evasion and resilience techniques used by cybercriminals to avoid the implemented security measures.

By reading this project, we could extract the huge impact botnets have, especially in IoT devices, the evolution of these systems and how difficult it would be to disrupt it, and the necessity to continuously improve our security measures, making special emphasis on the training in cybersecurity of any individual making use of our network.

5.2. Personal evaluation

This project has been very educational for its theoretical and technical knowledge. In the theoretical part by being able to analyze the operation and construction of botnets, the challenges and the risks these systems generates in networks, especially in IoT networks. Also, an approach of the different techniques used for infection, evasion, and resilience of malware providing an idea of cybercriminal tendencies and evolution. After, the different measures and countermeasures that are being taken to reduce the impact of botnets and malicious activities. From the technical point of view, being able to deploy a botnet using innovate technology such as IPFS, realizing the complexity of building a functional botnet, the challenges to avoid security measures, and to explain the risks of a distributed botnet system.

This project has provided me a better understanding of the challenges and difficulties to develop proper software that is able to detect and prevent malware, the necessity of a proper product life cycle for the IoT industry which considers security, the constant evolution of evasion techniques that although they are not new, cybercriminals apply it intelligently. At the same time, I feel proud that this project may contribute to the cybersecurity field.

5.3. Lines of development

In the case this project is continued, there are several lines of possible development:

Firstly, there are different improvements that could be made to the botnet, such as adding an identifier providing the capability to send a command to just a device, the addition of secondary communication channels, to apply evasion techniques that were explained in this project, or to provide the botnet the capability of using 0-day vulnerabilities for a posterior analysis of its impact.

Secondly, huge progress would be the development of an advanced tool to mitigate and detect botnets or malicious activities by the use of unsupervised artificial intelligence which nowadays it is still a challenge.

Bibliography and references

[1] Internet of Things:
https://www.cisco.com/c/dam/en/us/products/collateral/se/internet-of-things/at-a-glance-c45-731471.pdf

[2] New GSMA study: operators must look beyond connectivity to increase share of $1.1 trillion IoT revenue opportunity:
https://www.gsma.com/newsroom/press-release/new-gsma-study-operators-must-look-beyond-connectivity-to-increase-share/

[3] IoT Malware Comprehensive Survey, Analysis Framework and Case Studies:
https://i.blackhat.com/us-18/Thu-August-9/us-18-Costin-Zaddach-IoT-Malware-Comprehensive-Survey-Analysis-Framework-and-Case-Studies.pdf

[4] 2001 CERT Incident Notes:
https://resources.sei.cmu.edu/asset_files/WhitePaper/2001_019_001_496466.pdf

[5] RBOT Spybot:
https://www.trendmicro.com/vinfo/us/threat-encyclopedia/archive/malware/worm_rbot.zw

[6] ZLOB:
https://www.trendmicro.com/vinfo/us/threat-encyclopedia/malware/zlob

[7] Heads of Hydra. Malware for Network Devices:
https://securelist.com/heads-of-the-hydra-malware-for-network-devices/36396/

[8] Exploring Stuxnet's PLC Infection Process:
https://www.symantec.com/connect/blogs/exploring-stuxnet-s-plc-infection-process

[9] Carna Botnet (Census2012) :
http://census2012.sourceforge.net/paper.html

[10] Synolocker:
https://www.synology.com/es-es/company/news/article/Synology_Encourages_Users_to_Update_as_SynoLocker_Ransomware_Affects_Older_DSM_Versions

[11] Synology Dogecoin:
https://www.secureworks.com/blog/hacker-hijacks-synology-nas-boxes-for-dogecoin-mining-operation-reaping-half-million-dollars-in-two-months

[12] Moose Botnet:
https://www.welivesecurity.com/wp-content/uploads/2015/05/Dissecting-LinuxMoose.pdf

[13] Umbreon:
https://blog.trendmicro.com/trendlabs-security-intelligence/pokemon-themed-umbreon-linux-rootkit-hits-x86-arm-systems/

[14] Mirai:
https://www.incibe-cert.es/blog/ddos-actualidad-iot-y-los-dns-dyn

[15] Hajime:
https://securelist.com/hajime-the-mysterious-evolving-botnet/78160/

[16] Brickerbot:
https://www.trendmicro.com/vinfo/us/security/news/internet-of-things/brickerbot-malware-permanently-bricks-iot-devices

[17] IoT Reaper:
https://www.fortinet.com/blog/threat-research/reaper-the-next-evolution-of-iot-botnets.html

[18] VPNFilter:
https://www.symantec.com/blogs/threat-intelligence/vpnfilter-iot-malware

[19] Hide ‘N Seek (HNS):
https://labs.bitdefender.com/2018/05/hide-and-seek-iot-botnet-resurfaces-with-new-tricks-persistence/

[20] The State of IoT Security:
https://safenet.gemalto.com/iot-2018/

[21] CAP theorem:
https://en.wikipedia.org/wiki/CAP_theorem

[22] Botnet Communication Patterns:
https://ieeexplore.ieee.org/document/8026031

[23] PubSub pattern:
https://www.ibm.com/support/knowledgecenter/en/SSFKSJ_9.0.0/com.ibm.mq.explorer.doc/p_pubsubers.htm

[24] IPFS:

https://docs.ipfs.io/

[25] IPFS stack:
https://github.com/ipfs/specs/tree/master/architecture

[26] libp2p specification:
https://github.com/libp2p/specs#libp2p-specification

[27] IPLD:
https://ipld.io/

[28] Self-Certifying File System:
https://pdos.csail.mit.edu/papers/sfs:euresti-meng.pdf

[29] Raspberry Pi :
https://www.raspberrypi.org/help/what-%20is-a-raspberry-pi/

[30] Raspbian:
https://www.raspbian.org/FrontPage

[31] Raspberry Pi models and specifications:
https://en.wikipedia.org/wiki/Raspberry_Pi

[32] SSHpass package:
https://sourceforge.net/projects/sshpass/

[33] Mutt package:
http://www.mutt.org/download.html

[34] IPFS Requirements Roadmap:
https://github.com/ipfs/ipfs/blob/master/REQUIREMENTS.md

[35] Traditional Antivirus:
https://threatvector.cylance.com/en_us/home/how-traditional-antivirus-works.html

[36] Virtual Machine and Containers:
https://www.backblaze.com/blog/vm-vs-containers/

[37] PubSub IPFS:
https://docs.ipfs.io/reference/api/

[38] Denial of service consequences:
https://www.akamai.com/us/en/multimedia/documents/content/the-cost-of-denial-of-services-attacks.pdf

[39] HTTP Flooding:
https://www.imperva.com/learn/application-security/http-flood/

[40] GDPR:
https://gdpr.eu/what-is-gdpr/

[41] awk package:
https://packages.debian.org/jessie/awk

Appendix A

Impact analysis

This project by the use of different devices proceeds with the construction of an operational and distributed IoT botnet trying to demonstrate and analyze the huge impact it could produce on businesses and society if it is run by malicious actors. This botnet is on early stage, so there is not any real impact. However, if this project was developed and improved, the more relevant impacts it may produce are:

- Social impact: this project would have a social impact if certain entities suffered a denial of service attack. The worst case would be to influence critical infrastructures such as transport or health industries.

- Economic impact: this project would have some repercussions in the economic field if ransomware or denial of service were employed. As we have explained in this project, it would be able to produce great economic losses to businesses.

- Ethical impact: this impact is controlled as the proof of concept does not have any economic or ethical repercussion. Nevertheless, as it was explained if a denial of service affects healthcare, transport or another critical industry, it would have a high repercussion in people's life.

- Legal impact: all the technologies employed during the development of this project are properly licensed and legally acquired. All the activities of this project were made in a controlled and supervised environment. But the use of any physical or digital mean, to access or attack other network devices illegitimately is illegal which may incur in sanctions, police arrestment, etc.

- Environmental impact: this project does not have any environmental impact.

Appendix B

Project budget

This project has been developed for six months. The budget will be calculated taking into account human resources, software, and technical equipment.

- **Human resources costs**

  People involved in this project are the project manager (engineer) and the engineering student, author of the project.

	Cost per hour (€)	Working Hours	Total cost (€)
Project manager	90.00	30	2700.00
Engineering student	60.00	360	21,600.00
TOTAL			**24,300.00 €**

- **Software and technical equipment**

  The software used in this project was open source, so there is not a real cost. So, the hardware listed below has been used. The total costs are calculated by the product depreciation cost per month and the time used.

	Lifetime (years)	Units	Cost (€)	Depreciation (€/month)	Time used (months)	Total cost (€)
Laptop	3	1	700.00	19.44	20	388.8
Raspberry pi	3	2	80.00	2.22	5	11.1
Peripherals	3	4	48.00	1.33	5	6.65
TOTAL						**406.55**

Hence, taking into account the above items, the total expenses of this project is **24,706.55 €**.