Master's thesis

Master's Programme in Computer Science

Security issues and defences for Internet of Things

Yu Zhang

June 8, 2022

Faculty of Science

University of Helsinki

Supervisor(s)

Prof. Niemi, P Valtteri

Examiner(s)

Contact information

P. O. Box 68 (Pietari Kalmin katu 5)
00014 University of Helsinki,Finland

Email address: info@cs.helsinki.fi
URL: http://www.cs.helsinki.fi/

HELSINGIN YLIOPISTO – HELSINGFORS UNIVERSITET – UNIVERSITY OF HELSINKI

Tiedekunta — Fakultet — Faculty	Koulutusohjelma — Utbildningsprogram — Study programme
Faculty of Science	Master's Programme in Computer Science

Tekijä — Författare — Author

Yu Zhang

Työn nimi — Arbetets titel — Title

Security issues and defences for Internet of Things

Ohjaajat — Handledare — Supervisors

Prof. Niemi, P Valtteri

Työn laji — Arbetets art — Level	Aika — Datum — Month and year	Sivumäärä — Sidoantal — Number of pages
Master's thesis	June 8, 2022	61 pages

Tiivistelmä — Referat — Abstract

The Internet of Things (IoT) aims at linking billions of devices using the internet and other heterogeneous networks to share information. However, the issues of security in IoT environments are more challenging than with ordinary Internet. A vast number of devices are exposed to the attackers, and some of those devices contain sensitive personal and confidential data. For example, the sensitive flows of data such as autonomous vehicles, patient life support devices, traffic data in smart cities are extremely concerned by researchers from the security field. The IoT architecture needs to handle security and privacy requirements such as provision of authentication, access control, privacy and confidentiality.

This thesis presents the architecture of IoT and its security issues. Additionally, we introduce the concept of blockchain technology, and the role of blockchain in different security aspects of IoT is discussed through a literature review. In case study of Mirai, we explain how snort and iptables based approach can be used to prevent IoT botnet from finding IoT devices by port scanning.

ACM Computing Classification System (CCS)
Security and privacy → Network security → Mobile and wireless security
Security and privacy → Intrusion/anomaly detection and malware mitigation → Malware and its mitigation

Avainsanat — Nyckelord — Keywords

Security, Mirai, Internet of Things, BlockChain

Säilytyspaikka — Förvaringsställe — Where deposited

Helsinki University Library

Muita tietoja — övriga uppgifter — Additional information

Networking study track

Contents

1 Introduction

At present, the Internet of Things is developing rapidly and has gradually evolved from a concept to a part of human routines. More and more smart devices are joining people's lives. Many buildings and communities deploy sensors to save energy. Smart devices such as smart locks and smart meters are entering households. Traffic devices such as cars, taxis, and traffic lights are connected to the Internet to improve safety and transportation efficiency. More and more people use wearable devices and implantable medical devices cooperating with smartphones for detecting their own physical conditions. Industrial production improves management and production efficiency by connecting to the Internet.

The security issues of the Internet of Things gradually raised the general concern of the industry and users. In 2019, there were multiple news about hacking of Amazon's Ring home cameras, the voice of a stranger was suddenly heard from the cameras, and the hacker asked the owner to pay 50 bitcoins (valued at €350,000 back then). In the Black Hat USA conference 2018, the Keen Security Lab of Tencent showed the vulnerabilities which can achieve remote control of Tesla car (Tencent, 2016). According to a survey report by the HP Security Institute, 80% of IoT devices currently allow the use of weak keys, and 70% of communication between IoT devices and the Internet or local area networks has no encryption. From the user's perspective, because the Internet of Things is closely related to people's lives, the security issues of the IoT devices may pose a huge threat to the user's property and even life. For example, a security problem with a smart lock may allow a thief to enter the user's home easily and cause huge property damage to the user. If a networked car has a serious security problem, it may cause a traffic accident and endanger the user's life. If a heart rate detector is not working normally due to safety issues, it may not be able to inform the doctor in dangerous situations and miss the best rescue time.

In addition to the problems that can cause direct damage to users' property and lives, the data collected is inherently sensitive due to the correlation between IoT devices and personal information. Even data that does not seem important can potentially harm the privacy of users. For example, a smart meter which measure energy consumption allows eavesdroppers to determine whether the user is currently at home or not.

From an industrial point of view, there is a great demand for standard security solutions. A report published by the market research organisation Gartner's hype cycle (Camarinha-

Matos et al., 2013) in 2012 stated that the concept of the Internet of Things will take five to ten years to produce substantial productivity, the main factors are security challenges, privacy policies, data and wireless standards, and application and connection architecture for the realisation of IoT. Due to the trend of separation of IoT network infrastructure and applications, security risks are more urgent than architecture issues. For the Internet of Things, the goal is to connect things, and the basis for things to connect is that devices can communicate with each other. Among the many related network security technologies, ensuring the security of equipment communication are foundational.

This thesis presents the architecture of IoT and its corresponding security issues. Additionally, we introduced the concept of blockchain technology, and the role of blockchain in different secure aspects of IoT is discussed through a literature review. In case study of Mirai, we use snort and iptables linked approach to prevent IoT botnet from finding IoT devices by port scanning.

The remainder of this thesis is as follows. Section 2 gives an overview of IoT on fields of concepts, architecture, and limitations. The security issues of IoT and defence principles are discussed in section three and four, respectively. In section 4, we discuss the concept of blockchain technology and how to enhance the security of IoT by taking advantage of blockchain technology. In section 6, there is a study section on Mirai botnet, and we find a solution that using a snort-iptables linked approach to prevent IoT botnet from port scanning. The conclusion part is in the last section 7.

2 Background

The Internet of Things (IoT), as its name implies, is the things connect to other things via Internet. Firstly, the core and foundation of IoT is still the internet, while the Internet of Things is an extension of the traditional Internet. Secondly, network composition and communication of IoT extend to any objects or devices around us, such as RFID tags, smartphones, sensors, and kitchen applications. The entities in the IoT network can interact with each other at anytime and from anywhere to complete various tasks for information exchange and communication. As a connected network of huge number of things, IoT has been associated with great expectations. The expected application areas include smart logistics, smart transportation, precision agriculture, environmental protection, smart power grid, smart home, healthcare, public security, smart buildings and urban management. It can be said that the future life is closely related to Internet of Things.

2.1 History of IoT

Bill Gates, the founder of Microsoft is the first one to propose the term "Internet of Things" in his book "The Road Ahead" written in 1995 (Bill Gates and Rinearson, 1995). He described: "Today's internet only realised the interconnection between computer, not the interconnection of everything in the world. In the future ahead, you will be able to customise your favourites on television, your lost or stolen items can automatically send you a message to inform you where it is no". This is the very first vision of Internet of Thing although Gates did not mention the actual words in his book.

In 1999, Kevin Ashton, the vice president of Procter & Gamble defined the internet of Things as "connecting all items to the internet through information sensing devices such as radio frequency identification (RFID) to achieve the intelligent identification and management" (Ashton, 1999). This is allegedly the embryonic stage of concept of Internet of Things.

In March 2008, the world's first international conference on the "Internet of Things 2008" at Zurich discussed the new ideas and technologies about the means of continuation development of IoT. In the same year, the National Intelligence Council (NIC) listed the

Internet of Things technology as one of the six revolutionary technologies that have potential effects on the American national power.

2.2 Concepts of IoT

When designing a security solution for the Internet of Things, it is necessary to have a certain understanding of the concepts of Internet of Things. A security solution for the Internet of Things must also adapt to the characteristics of it. Although the Internet of Things has different characteristics from different perspectives, it can be said that the Internet of Things has the following most important characteristics (Chaudhary et al., 2019; Madakam et al., 2015):

- Scalability
  There are billions of nodes in the Internet of Things already now. The enormous network nodes or devices not only means that the network scale is unprecedently large, but also the massive data generated by these devices, the services and applications provided and the market are large-scale. In addition, during the development of the Internet of Things, the scale of the network will continue to increase, so scalability will also be an important feature in the development of the Internet of Things. The figure 2.1 from IoT Analytics Research 2020 shows the estimated number of active devices from 2015 to 2025. It clearly presents that the quantity of IoT devices is increasing much faster than that of non-IoT devices. More specifically, from year 2015 to 2025, the number of IoT devices will rise roughly ten times from 3.6 billion to 30.9 billion while the number of Non-IoT device is increased at much slower rate from 9.7 billion to 10.3 billion (Chaudhary et al., 2019; Madakam et al., 2015).

- Heterogeneity
  The Internet of Things connects various devices to the network, which will bring a wide range of heterogeneity. The heterogeneity of the Internet of Things will be reflected in many aspects. The hardware performance of the devices in the network varies a lot, not only including strong performance servers, personal computers, mobile phones, but also sensors, actuators and other equipment with limited resources. Communication means used by devices in the network are rich. Devices can access the network through Bluetooth, Wi-Fi, LTE, wired and other technologies. In addition, different devices are using different networking topology and transmission

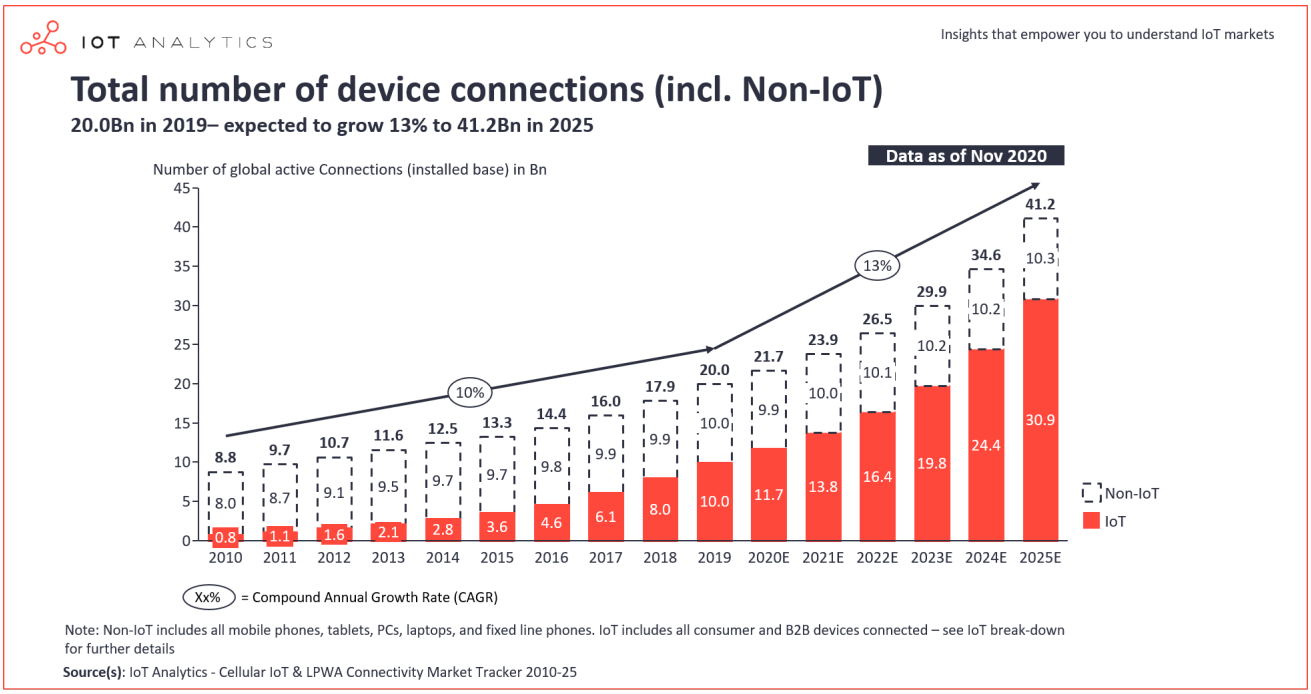

Figure 2.1: Total number of device connections (Lueth, 2020)

protocols. The heterogeneity will be seen in every aspect of the IoT domain. Table 2.1 lists some relevant information about IoT devices which shows the wide heterogeneity of the IoT. The data of the specification of hardware are getting from manufacture's website (Chaudhary et al., 2019; Madakam et al., 2015).

- Interoperability
  Although the Internet of Things is inherently heterogeneous, the IoT requires its components to interact with each other. For the IoT, simply connecting the device to the internet is not enough. It is more important to promote the interconnection and data integration of devices to provide automated and intelligent services.

Type of Device	Processor	RAM	ROM	Power supply	Communication
Iphone 13	Hexa-core 3.2G	4GB	256G	Battery	Wifi, 5G, NFC
Samsung Smart TV	1.3GHz	1GB	64/128G	AC	WiFi
Raspberry Pi 4	1.5GHz	1-8G	Micro-SD	USB-C	WiFi, Bluetooth
Netgear Router	1GHz	256M	128M	AC	WiFi/4G
Philips Smart Bulb	32MHz	8K	<256K	AC	Zigbee
Fitbit	32MHz	16K	128K	Battery	BLE

Table 2.1: Hardware of end-user devices

- Availability and reliability
  Availability and reliability are the fundamental requirements of communication and service system. The Internet of Things should not only be available, but also reliable. The Internet of Things must be flexible enough to guarantee a specific level of availability, and it also needs to provide reliable performance that can be customized to specific applications.

- Openness
  The definition of the Internet of Things gives the Internet of Things a concept of natural openness. In addition to providing raw data and other special services to Internet of Things devices, an IoT platform should be flexible enough to allow third-party organisations to develop complex applications based on the provided APIs.

- Security
  Security will be a very prominent feature for the Internet of Things. Since the Internet of Things is closely related to the real world, its security issues may bring serious consequences. In order to realize the security of the Internet of Things, many issues need to be considered when designing IoT system, such as how to protect the security of communications, how to manage the authentication, and how to protect the privacy of users.

IoT & Internet The common point of the Internet of Things and the Internet is that the technical basis is the same, that is, they are all built on the basis of packet-based data networks. The difference between the Internet of Things and the Internet is that their packet networks are different in terms of organisation, functions, performance, and requirements of network. The Internet, such as IPv4 and IPv6 protocols, mainly emphasises the openness and accessibility of specifications. The requirements for network performance are maximised transmission capacity and priority-based resource management. There are low requirements for security, credibility, controllability, and manageability. However, the Internet of Things has very high requirements on those. Currently there are several IoT systems, which have high requirements for real-time, security, credibility, and resource assurance. These requirements are beyond the capability of the current IP network could provide. Therefore, two points can be determined. One is that the Internet of Things does not necessarily use an IP network, where the current IP network can only provide "best-effort" transmission capabilities. The other point is that the lightweight communication protocols suit better for the Internet of Things, especially the Internet of Things for

small smart things. In other word, complex protocols such as TCP/IP should be avoided for small IoT smart objects. As a conclusion of these two aspects, the Internet of Things should have a network environment different from the traditional Internet, rather than a simple extension of the existing Internet.

2.3 Architecture of IoT

Before diving deep into the architecture of IoT, it is necessary to start on introducing the Open System Interconnection (OSI) model for better understanding of how IoT protocol works. OSI model is a layered server architecture system, and it was first proposed in 1973. Each layer is defined according to a specific function to be performed. There are 7 layers divided into upper/software layers (Application, Presentation and Session) and lower/hardware layers (Network, Data Link and Physical). The Transport layer is the heart of OSI model between upper and lower layers. All these seven layers work collaboratively to transmit data from one layer to another (Y. Li et al., 2011; Williams, 2022).

- Layer 1: Physical Layer. This is bottom layer of OSI model. This layer defines the physical and mechanical specifications to make the devices connect to a physical transmission medium. The example of hardware can be optical cable, radio, network adapters, ethernet, etc.
- Layer 2: Data Link Layer. The data is divided from bit into frames in the data link layer. This layer ensures that the frames received or transmitted from source to destination should be error-free. The layer also help you implement best routing path of packets through a network.
- Layer 3: Network Layer. The network layer provides the functional and procedural means of transmitting variable length data from A to B via one or more networks.
- Layer 4: Transport Layer. The transport layer is responsible for passing messages from one process to another. In this layer, the most important thing is confirmation. Confirmation is the process of successful of transmission data on the network.
- Layer 5: Session Layer. Responsible for establishing and maintaining the communication between two computers in the network during data transmission.

OSI model (Layers)	TCP/IP Four Layer Model	Corresponding Network Protocols
Application	Application	HTTP, TFTP, FTP, NFS, POP3, SMTP
Presentation		MPEG, Rlogin, MPEG, TLS
Session		NetBIOS, SAP
Transport	Transport	TCP, UDP
Network	Internet	IPV5, IPV6, ICMP, ARP, IPSEC
Data Link	Network Access	FDDI, Ethernet, Frame Relay, SLIP, PPP
Physical		IEEE 802.1A, IEEE 802.2-802.11

Table 2.2: Network protocol according to OSI and TCP/IP model (Williams, 2022)

- Layer 6: Presentation Layer. This layer is also called translation layer. The data is converted into a format compatible with receiver's system and suitable for transmission
- Layer 7: Application Layer. As the highest level of OSI model, this layer provides an interface for application to set up communication with another application. The example of application can be email, file transfer, SSH client, and instant chat program, etc.

Another approach is TCP/IP protocol, which gives a correspond simple model of four layers. The table 2.2 shows the representation of both models and the most common protocols in each layer.

In the filed of IoT, the architecture is not agreed in pace with the development of IoT system. A different researcher has defined different architecture in every stage of development. The most basic IoT architecture is three-layer model (Mashal et al., 2015). It has three layers, the perception, network, and application layers. On top of that, there are many new definitions of layers proposed in the literature. One example is a five-layer architecture, which adding business layer and processing layer. The processing layer is also regarded as a middleware layer, which stores, analyses and processing data comes from the perception layer. The business layer is responsible for managing and controlling IoT applications, business and profit models (Burhan et al., 2018). In the next section, the functions and security issues of each layer will be discussed.

2.3.1 Perception Layer

The perception layer is the most basic layer in the three-layer system of the Internet of Things. It mainly includes two parts: data acquisition and data short-distance transmission. Firstly, the perception layer collects data from the physical environment through sensor devices such as infrared, ultrasound, temperature and humidity. The most generic sensor is the smartphone nowadays. It has many types of sensors such as GPS, camera, light sensor, microphone, proximity sensor and movement sensors. The collected data will be transmitted through short-distance transmission technologies like RFID (Radio Frequency Identification), NFC (Near Field Communication), Bluetooth, and ZigBee. Some studies divided the security of the IoT perception layer into RFID security and wireless sensor network security (Kozlov et al., 2012). The perception layer is crucial to the protection of information in the process of sensing information. If the security protection of the sensing information is not implemented, the information can be easily obtained by attackers which may cause huge security risks in some cases.

2.3.2 Network layer

There is a huge amount of heterogeneous IoT devices connecting to the network. The network layer is a bridge between the perception and the application layer. It transmits the information gained by the perception layer to the required place according to the requirements of the application layer. An unauthorised node connected to the network will cause a lot of security problem. For example, a large number of unauthorised devices accessing the network in a short time will cause network congestion. There are two major security issues in the network layer: (1) The data is easily hacked. The data is extremely easy to be tampered or attacked when transmitting. Because the data will not be protected by complicate encryption algorithms due to the limited computer capability on IoT equipment. (2) Convergence of heterogeneous networks. The network layer of IoT is combined with multiple open networks. As the degree of network convergence increases, the network structure becomes more and more complex. When data is transmitted from one network to another, it often goes through multiple network protocols. The incompatibility of various networks provides a good chance to attackers (Jing et al., 2014).

2.3.3 Application Layer

The application layer is on the top of the three-layer structure of the Internet of Things, and its function is "processing". The application layer can perform calculation, processing, and data mining when the data flow collected from the perception layer. The application of IoT can be a smart home, smart city, E-health and Industry 4.0, etc. For each application, the service may has different dependence on the information collected by the sensor. For instance, the user account of IP camera is easily guessed and cracked by default password. Due to the devices used in smart homes have the weak computing power and storage capacity (Ali, 2018), they can introduce many threats and vulnerabilities from both inside and outside domain.

2.4 Limitation of IoT

From the perspective of hardware, the Internet of Things is composed of a large number of heterogeneous hardwares, which have more complicated setting than that of traditional Internet devices. According to the performance of devices, IoT devices are divided into two categories.

The first category consists of standard devices, including single-board computer unit such as smartphone, smart TV, laptop and Raspberry Pi. These equipments have enough resources to run traditional operating system (Windows, Linux, BSD, etc.) and corresponding network security protocols to their hardware and software environments.

Low-end equipment, such as Arduino or Wasmote, form the second category. These devices have very limited resources and are also called resource-constrained nodes. Resources are constrained in several ways:

- Code complexity is limited because of the size of ROM.
- System memory and cache are limited because of size of RAM.
- The performance of device is limited because of low frequency on processor.
- The available power is limited because of battery size.

Class	*RAM*	*Rom/Flash*
Class 0	<<10KB	<<100KB
Class 1	10KB	100KB
Class 2	50KB	250KB

Table 2.3: Classification of Resource-constrained node (Bormann and C. Gomez, 2016)

- The accessibility of user interface is limited.

The Internet-standards community defined the terminology of constrained nodes on document RFC 7228. Resource-constrained nodes are divided into three categories, as shown in Table 2.3.

In the above classification, the storage resources and processing capabilities of Class 0 devices are so constrained that they may not be able to directly connect to the Internet in a secure way. Class 0 devices are likely to need the help of gateways, proxies or servers to access the network. They usually cannot guarantee security or management in the traditional way, so they often need to be pre-configured with a very small set of security rules which never change after manufacture (Bormann and C. Gomez, 2016).

On Class 1 devices, the code space and processor capabilities are relatively limited. These devices cannot easily communicate with Internet nodes using full functionality protocols such as HTTP, TLS and TCP. Instead, they can use special design protocols which are suitable to resource-constrained devices, such as CoAP and UDO. Under normal circumstances, Class 1 nodes can support security functions and join the IP network as fully developed nodes (Bormann and C. Gomez, 2016).

Class 2 device resources are less limited and have the ability to support the same protocol stack as used on a laptop or server. However, this type of devices can benefit from lightweight, energy-efficient protocols. In addition, the applications running on the devices have more energy for their use when the network protocols consume less energy. Therefore, applying protocols designed for resource-constrained nodes to Class 2 devices may reduce deployment costs and increase interoperability. Devices whose capabilities and performance are significantly beyond Class 2 can directly use existing protocols without any changes, so there is no additional description on those devices in the definition of RFC 7228 (Bormann and C. Gomez, 2016).

2.5 Cyber Security

In recent years, cyber attacks have increasingly occurred in the Internet. Before discussing the security of IoT devices, it is necessary to understand the cyber attacks and their classification. The National Institute of Standards and Technology (NIST) defines a cyber-attack as: An attack, via cyberspace, targets an enterprise's use of cyberspace for the purpose of disrupting, disabling, destroying, or maliciously controlling a computing environment/infrastructure; or destroying the integrity of the data or stealing controlled information. (Standards and Technology, 2013).

In 2017, WikiLeaks announced a publication of new cyber attack tools released from CIA and National Security Agency (NSA), including a large number of remote attack tools, known vulnerabilities, and documents related to attacks. Under the openness of Internet, those tools are quickly spread through darknet, forums and blogs. Therefore, it becomes easier for hackers to achieve a proper tools.
Cyber attacks can be divided into classes of communication attacks and application attacks. In the following section, we will introduce the most common attacks applied in each field.

2.5.1 Communication Attack

The communication attack including DDos attack, Man-in-the-middle attack and spoofing attack. The details of those attacks are discuss as follows.

Denial-of-service (DoS) and Distribute Denial-of-service (DDos) attacks use brutal ways to exhaust the resources of the target. The purpose is to prevent the target computer or network from providing normal services or resource access. As a result, the target service system stops responding or even crashes. The service resources can be network bandwidth, file system capacity, open ports or connections. No matter how fast the computer processor is, how large the memory capacity is, or how fast the network bandwidth is, the consequences of this attack cannot be completely avoided (Darwish et al., 2013).

Man-in-the-middle attack (MITM) is a general term for when an attacker positions himself in communication between an user and application, disguised as normal exchange

of information is undergoing. The goal of such attack is to steal personal information, such as account details, login credentials, or credit card numbers. MITM brings a serious threat to online security because it provides a chance for attackers to capture and tamper information in real-time.

Spoofing is an act of impersonation. Deception is as old as humanity itself. When deception and information technology are combined, it generates a field of network attack. Hackers disguise themselves as someone or something who is authorized user to deceive the trust from system or administer (human). There are three main types of spoofing attacks (Raguvaran, 2014) :

- Email spoofing is the most common form of online spoofing. Scammers send emails to multiple addresses and use official logos and header images to pretend to be representatives of banks, companies, and law enforcement agencies. The emails they send contain links to malicious or other fraudulent websites, as well as attachments infected with malware. Some fraudsters may also use social engineering to trick victims into voluntarily divulging information. They usually tell the victim that their bank or game account has been stolen, thus one need to reset the password, and forge a connection for the victim to open. When an unsuspecting victim clicks on the link, they will be lead to a fake site where they must log in with their credentials, including their real username and passwords. Finally, these information will be forwarded to the attacker's back-site.

- DNS spoofing. Every computer and website on the Internet has an unique address called IP address. When the user enters an URL in the browser, the DNS server will find the matching IP address for the website. When a hacker uses the DNS redirection method, an user will be always redirected to a fake website designed by hacker no matter what URL user enters. This is called DNS spoofing. The information entered by the user after visiting the counterfeit website will eventually be recorded by the hacker in the background.

- IP spoofing. The IP data packet generated by the attacker has a forged source IP address in order to impersonate the identity of the sender. According to the definition of Internet Protocol, the data packet header contains source and destination information. IP address spoofing is to forge the header of a data packet so that the source of the displayed information is not the actual source, as if the data packet was

sent from another computer. The hacker interacts with the server using an borrowed IP address in order to pretend to be another machine.

2.5.2 Application Attack

The application attack are the treats mainly target on application, such as SQL injection, XSS attack and malicious code attack.

SQL Injecting is a code injection technique in which SQL code is attached to the input parameter and passed to server for execution. Specifically, the attacker injects (malicious) SQL commands into the back-end database. A successful injection means that the hacker can completely disclosure all data on the system, modify existing data and spoof himself all the time. SQL injecting generally happens in dynamic webpages with parameters inputs. Sometimes a dynamic webpage may only have one parameter or N parameters. If the programmer does not have security awareness and perform a necessary character filtering, the possibility of SQL injection is very high.
Below is an example of number inject through browser.
After entering 'learn.me/sql/article.php?id=1' in the browser, this command is equivalent to calling a query statement:

```
$sql = "SELECT * FROM article WHERE id =",$id
```

Here, the id parameter is accepting input through URL of browser. If you change the input value, for example, id =-1 OR 1 = 1, it will yield different output based on the given input. This is a SQL injection attack, and it may force the system return all information in the database.

Cross-site scripting (XSS) is a type of security vulnerability. Attackers can use this vulnerability to inject malicious code into web application. Different from other attack vectors (SQL injection), XSS does not target the application itself, instead, the users of the web application will be the victims. According to the Open Web Application Security Project (OWASP, 2017), XSS was recognised as one of the 7 most common web application vulnerabilities in 2017. XSS attacks are prone to occur in the following two situations:
i) Data were entered to a Web application from an unreliable link.

ii) Dynamic code without filtering out the malicious content is sent to web users. Malicious content generally includes JavaScript, while sometimes it also includes HTML, FLASH or other browser executable code. XSS attacks have many different forms, but usually they have following ways to launch the attack. For instance, the cookies from browser or other private information were sent to the attacker, the victim was redirected to a web page controlled by the attacker, or other malicious operations on the victim's machine via a malicious website.

XSS attacks can be divided into three categories: Reflective (AKA Non-Persistent or Type I), Stored (AKA Persistent or Type II) and DOM-Based (OWASP, 2005).

- Stored XSS
  An attacker uses Storded XSS to inject malicious script into the target server. If there is no input validation, this script is permanently stored by the application, such as database, chat-log, comment, and forum. For example, the attacker can enter malicious code into the comment filed in a blog and save those onto the back-end. When a victim opens the affected web page, the XSS malicious code will loaded as part of HTML code on the browser. Thus, the malicious code will be executed together with legitimate code on the web application when the user visit the page.

- Reflective XSS
  Reflictive XSS is most common type of XSS attack. Different from the storded XSS, the attacker's payload (malicious script) has to be a part of request sent to server. When an user clicks on a malicious link, submits a form, or phishing Emails, the reflected XSS is then executed in the broswer (Pranathi et al., 2018). Reflictive is not a persistent attck becasue the attacker need to lure the victim to download the payload.

- DOM Based XSS
  Document Object Model (DOM) Based XSS (also called type-0 XSS) is an XSS attack where the attack payload is executed as a result of modifying the DOM "environment" in the victim's broswer. This attack usually arises when JavaScript takes data from hacker-controllable source, and passes it to a sink that supports dynamic code execution. When the malicious JavaScript is exectued on the victim's device, the information from other users' account in the same device may be hijacked by attacker. The payload on the webpage itself does not change, this is in contrast to other XSS attacks (Reflective and Storded), where the payload is placed in the

response page (OWASP, 2005).

Malicious code Attack Malicious code is also known as malware or spyware. This code can be installed and run on a mobile device or a computer without explicitly prompting permissions. The malicious code can start by itself or run as part of an application. Especially on mobile phones, few users using anti-virus software to check application regularly. Therefore, illegal modification can be easily neglect by mobile users (Ali, 2018).

2.5.3 The evolution of threats

The types of attacks on the Web are continuously changing with the development of the Internet. Due to the wide usage of web application, security issue has gradually received attention and become a hot topic in the security field. Hackers have quietly shifted their focus to the weaknesses during web application development. Open Web Application Security Project (OWASP, 2017; OWASP, 2013) is an open community, non-profit organization. There are currently 82 divisions around the world with about 10,000 members. The task of OWASP is discussing and assisting in improving Web software security standards, tools and technical documents. The long-term goal of OWASP is to help governments or enterprises understanding the security of web applications and services. As an IoT developer, it is important and necessary to follow the instruction from OWASP before one starts designing an IoT Ecosystem. The table 2.4 is a list of Top 10 most critical web application security risks from 2013 to 2017 according to OWASP.
In the past few years, the basic technology and structure of the application have undergone major changes, traditional applicaitons are replaced by microservices written in Node.js and Spring boot. Javascript is now one of the main languages of the web application. Among them, Node.js runs on the server side while modern web frameworks such as Bootstrap, Electron, Angular, and React run on the client side. Applications written in the JavaScript framework allow the creation of highly modular and feature-rich front ends. In the next chapters, we will discuss the security issues of IoT systems and the defence methods to meet security requirements.

OWASP Top 10- 2013	OWASP Top 10-2017
A1- Injection	A1- Injection
A2- Broken Authentication and Session Management	A2- Broken Authentication
A3- Cross-Site Scripting (XSS)	A3- Sensitive Data Exposure
A4- Insecure Direct Object References	A4- XML External Entities (XXE)
A5- Security Misconfiguration	A5- Broken Access Control
A6- Sensitive Data Exposure	A6- Security Misconfiguration
A7- Missing Function Level Access	A7- Cross-Site Scripting (XSS)
A8- Cross-Site Request Forgery (CSRF)	A8- Insecure Deserialization
A9- Using Component with Known Vulnerabilites	A9- Using Component with Known Vulnerabilites
A10- Unvalidated Redirects and forwards	A10- Insufficient Logging&Monitoring

Table 2.4: Top 10 web-application threats from 2013 to 2017 (OWASP, 2017; OWASP, 2013)

3 Security issues of IoT

The Internet of Things implement on the integration of existing network architecture and application platforms and architecture. Most of the mechanisms on the Internet are still applicable to the Internet of Things. Certain security terms, such as authentication, verification and encryption apply also to IoT. However, some appropriate adjustments and supplements are needed for certain security mechanisms based on the characteristics of the Internet of Things. These adjustments and supplements are manifested in the following aspects (Razzaq et al., 2017):

- Local security of IoT devices
  IoT devices can complete complex and dangerous tasks instead of human labor. It is straightforward for attackers to reach and cause damage to these devices, or even to replace the software or hardware of the device without any notice. The local security of IoT devices is particularly critical.

- Transmission and security of the core network
  Due to the huge number of nodes, IoT device may lead to network congestion and Denial-of-Service (DoS) attack.

- Unattended in the site
  Most of the devices in IoT are deployed only once before connected to the network. It is tricky to realize remote monitoring of the IoT devices when the nodes are unattended in the site.

3.1 Security issues of IoT in three layers

The following security analysis is based on characteristics of each IoT layer.

Perception layer is a closed system composed of wireless sensor network (WSN) which completes all communications with external networks through gateway nodes. Different node has its own hardware configuration, so the overall computing and communication capabilities are limited. In the IoT architecture, it is hard to implement a single security principle to meet the demands of security requirements. Reducing the overhead of the

encryption protocol is a major security issue to solve. Authentication and confidentiality are both indispensables although there is a challenge to set a uniform standard for security services. In order to guarantee the security of the internal communication of the perception layer, this layer requires a proper key management mechanism. Confidentiality requires the setting of a temporary session key, and authentication is solved by symmetric or asymmetric cryptography schemes (Hameed and Alomary, 2019; Razzaq et al., 2017).

If the information obtained by the sensing node does not have security protection, the information will likely be illegally obtained by third parties. Factors that are taken into account such as the cost of implementing security protection or the convenience of usage, some nodes on the perception layer may only apply very simple information security protection. In that case, the information passing on that nodes are easily leaked.

The session key for communication between nodes in the perception layer is difficult to grasp, so even if an attacker captures the key of the node, he cannot actually control it. Generally speaking, the possibility of illegal control of the key node in the perception layer is very low. However, if an attacker obtains a shared key between a key node and other nodes, he can actually control the node and obtain all the information passing through the node.

The perception layer will eventually access the external network, so it will unavoidably be attacked by the external network, for example, DoS attacks. In the traditional Internet environment, if the node on perception layer cannot precisely identify the DoS attack, the network may be paralyzed. Therefore, it is necessary for the perception layer to have the capability to resist a small amount of DoS attacks (Hameed and Alomary, 2019).

Network layer is responsible for the safe and reliable transmission of information from the perception layer to the application layer. The key to maintain the security is the management of network infrastructure. Due to the diverse network architectures that need to be connected together, the challenge is to implement secure authentication mechanism across network. The design of the security architecture on network layer must prioritize efficiency, compatibility, and specificity. The IoT consists of a large number of networks that have diverse architectures with huge variations in the defence capabilities against network attacks. A smooth transition from heterogeneous networks to IoT must consider the

consistency and compatibility of security protocols. The network layer is easily accessed illegally if network access control is not utilized on it (Razzaq et al., 2017).

The characteristics of the network layer are summarized as follows:

- With varied application fields, the IoT network has distinct security and service requirements. It is impossible to directly copy the technical model of the traditional Internet. In addition, current communication networks are designed from the perspective of incoming communication, which does not fit well for device communication. Applying current security mechanisms will cause an inappropriate side effect between IoT devices.
- The network layer meets all the security issues from external network flow. However, its security problems will be more complicated than traditional networks because the data collected from the perception layer is massive and there is a big variety of heterogeneous data sources.
- The Internet of Things requires strict security and controllability. Most of the IoT applications are related to personal privacy or corporation secret information. Therefore, it must be capable to guard user's privacy and fight network attacks (Hameed and Alomary, 2019).

Application layer is directly facing users. Applications of the IoT are closely associated with the public and involves numerous domains and industries. Due to the tremendous volume of data information processing, it encounters challenges in reliability and security, which are control and management, middleware, and privacy protection (Razzaq et al., 2017).

- Control and management
  It is difficult to solve how to remotely control its equipment and accomplish the configuration of business information in IoT because end IoT devices are regularly deployed only once, after that these device are connected to the network unsupervised. Also, the IoT needs to establish a solid and consolidated security management platform, but this may impair the trust between the network and the management platform which generates a new round of security issues.
- Middleware
  When comparing Internet of Things to human body, the perception layer is like

the human limbs, the transmission layer is the human body and viscera, and the application layer is like the human brain. The middleware comprises the soul of the IoT, which plays a remarkably critical role in IoT.

- Privacy protection
  Unlike the perception layer or the network layer, privacy protection at the application layer is an issue that require more attention. With the networking of personal and business information, more and more information is considered to be private. Designing varying levels of privacy protection is becoming a popular research subject in IoT security. The popular privacy protection strategies is peer-to-peer computing, which is achieved by directly transacting and sharing computer-related services and resources (Hameed and Alomary, 2019; Razzaq et al., 2017).

4 Defence principles

Different from the Internet, the Internet of Things is mainly used to realize communication between people and things, or between things and other things. Thus, the scope of communication has expanded to cover things also. To solve the security problems of the Internet of Things, specific concepts are put forward in terms of security mechanisms as follows: Data encryption, user authentication and access control.

4.1 Data Encryption

Cryptography is an ancient technology that converts understandable information into incomprehensible information. In cryptography, the original information is called plaintext while the new information generated by transforming is called ciphertext. The process of converting plaintext into ciphertext is called encryption, and the mathematical part in the encryption process is called the encryption algorithm. Re-converting ciphertext into plaintext is called decryption, and the mathematical part in the decryption process is called the decryption algorithm. Keys are the most important parameters entered into the encryption and decryption algorithms. In addition to keys, other parameters such as initialization vectors and counters may be needed for cryptographic algorithms. When the encryption and decryption keys are same, the keys are called single-keys or symmetric keys. On the other hand, when encryption and decryption keys are not same, the keys are called double-keys or asymmetric keys.

Symmetric Key Cryptography Symmetric encryption refers to a cryptographic scheme that uses the same key for encryption and decryption. Before 1970, this method of information encryption was widely used in confidential communications for government and military. Nowadays, symmetric key encryption is also widely used in various computer systems to enhance data security. The encryption process is summarized as follows: the plaintext (as input) is encrypted with a key, using the encryption algorithm that generates the ciphertext (output). If the encryption scheme is strong enough, the only way for finding plaintext is to decrypt ciphertext with the correct key (Delfs and Knebl, 2015). The most straight-forward way to crack a symmetric encryption system is randomly guess-

ing the key. However, it takes billions of years to crack a 128-bit key by normal computer hardware. The longer the encryption key, the harder to crack it. The 256-bit key is generally considered to be extremely secure, and it can theoretically defend the brute-force attack by quantum computer (Delfs and Knebl, 2015).

On the other hand, when both parties use the same key and the key is shared over an insecure network, the key can easily be intercepted by malicious parties. If an unauthorized user gains access to a specific key, the security that is based on using that key to encrypt data will be compromised. To solve this problem, many web protocols use a combination of symmetric and asymmetric encryption to establish a secure connection. The most common example of this kind of protocol is the Transport Layer Security Protocol (TLS) (Rescorla, 2018), which is used to protect most network connections on the modern Internet.

Asymmetric Key Cryptography Asymmetric cryptography is also called public key cryptography. It is a method that uses two different keys: one for encryption and another one for decryption. Let us look at an example to illustrate how asymmetric cryptography works, see Figure 4.1. Imagine that Peter has a box with special lock which has three states instead of just two: A (locked), B (unlocked), C (locked). The lock has two keys: K1 can only turn to left, and K2 can only turn to right. This means that if the box is locked in the position A, only K2 can unlock it by turning right to position B. Only K1 can unlock the box from position C. In other words, either of two keys can lock the box, but only the other key can unlock the box later. Now, suppose that Peter made many copies of K2 (public key) that can only rotate to the right, and assigned these copies to the people who need it. At the same time, only Peter keeps the K1 (private key). What will happen next?

Option 1: Another person Ace can send confidential data to Peter through the box. Once Ace has locked the box with public key (from B to C), only the key rotates from right to left can unlock it. This means only Peter's private key can unlock the box.

Option 2: If the box is locked into position A, Ace can be sure that the content of the box is indeed from Peter. This is because the only way to turn the position from B to A is by using Peter's private key.

The example can be summarized as follows: Anyone can use the public key to encrypt data, but only the owner of the private key can decrypt it. Under above logic, anyone can safely send data to the private key owner. In addition, anyone can verify that the data received from the owner of the private key is indeed from that person (Cloudflare, 2020).

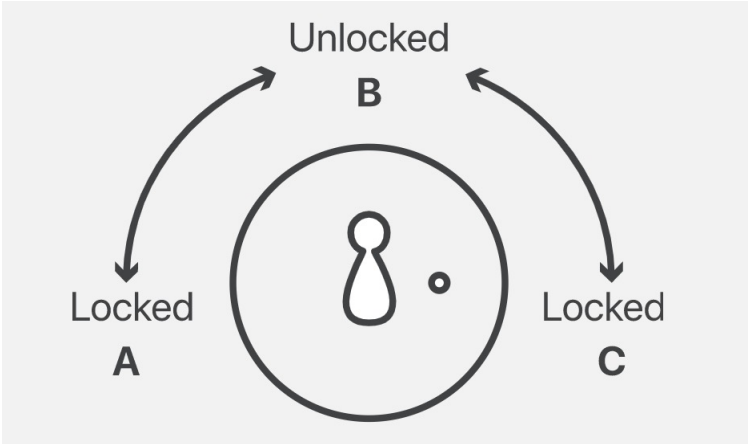

Figure 4.1: Key and Lock (Cloudflare, 2020)

For the reason of easier key management, asymmetric encryption is typically used in systems where a large number of users need to encrypt and decrypt messages at the same time, especially when devices of the users have sufficient computing resources. A common case is an encrypted email system, where use the public key originally from receiver to encrypt messages and the private key of the receiver can be used to decrypt.

Signature Cryptography and hash Digital signature refers to a string of characters attached to the data, or to the encrypted conversion of the data. The receiver uses this information to confirm the source of the data and its integrity. Digital signature serves the same purpose for a digital document as hand-written signature or seal serves for a physical document. Digital signatures are based on public key cryptographic algorithms and hash algorithms to sign data.
A hash function is a mathematical function that converts data of arbitrary size into a fixed length digest of the data. One application of hash is the verification of message integrity. For instance, the hash of document d is denoted by H(d). For two different documents, d1 and d2, there is very little chance that H(d1) is equal to H(d2). The longer the hash, the less likely it is that two different documents would have the same hash. (Knudsen and Preneel, 1999). For this reason, it is sufficient to digitally sign hash of a document instead of signing the whole document.

Digital signatures have very similar characteristics as hand signatures but achieved in a different way:

1. Origin authentication. To generate a legal digital signature, you must know the signer's private key. It is impossible to forge a digital signature without knowing the

private key.

2. Assurance of data integrity. Because the hash function has the characteristics of message integrity, the signer can digitally sign the value of the hash function instead of the message itself, and the signature is bound to the message at this time. If the message is modified, the value of the hash function will change and the verification of the signature will fail.

3. Signatory non-repudiation. The digital signature can be used to verify the identity of the signer. The public key of the signer can authenticate the digital signature, and the identity can be determined by the digital certificate. What is important is that the certificate also contains a signature of some party that vouches for the public key of the identity. The signer cannot deny the authenticity of their signature on the document in a later phase.

4.2 User Authentication

User authentication is a process that keeps unauthorized user from accessing system, network, or device. There are many technologies currently available to a system administrator to authenticate users. In here, password based authentication, hardware based authentication, and biometric authentication will be introduced in this section.

Password-based authentication Password-based authentication is the most common way to prove identity (Conklin et al., 2004). In the initial stage, the user registers user name and login password in the database of the system. Note that this password is typically valid for a long time. This login password is also called a static password. However, this method has serious security problems if the password is lost. All of the security of the user's account only depends on the protection of password. What is more, many people are using one or two username/password combinations on all web services. Once the password is leaked from one website provider, all services for which the same password is used are also in danger. The network application are constantly developing, and the password authentication technologies are developing as well. In order to prevent computer process from simulating automatic login, many platforms implement Completely Automated Public Turing test to tell Computers and Humans Apart (CAPTCHA) (Chen et al., 2014) technology which distinguish human input from computer simulating many human login attempts.

In addition to the "leaking password attack", the password-based authentication is vulnerable to the following attacks:

- Dictionary attack. The attacker can enumerate most probable passwords to generate a "dictionary". Then this attacker, or also some other attacker, can use the dictionary to attack user's account. After the attacker obtains verifiable information related to the password, such as name, birthday, ID number, he can perform a series of operations in conjunction with the dictionary to guess user's password and username based on the information obtained.

- Brute force attack. A brute force attack also known as brute force cracking uses trial-and-error to guess every possible combination of login info and password, in order to eventually to find the correct one. Although it is an old attack method, it is still simple way to break an account, especially as the computer resources are becoming more and more powerful.

- Keyboard monitoring. Keyboard logging often referred as keystroke logging or keyboard capturing, is the action of recording the keys struck by user on the keyboard. Most of time, a person using the keyboard is unaware that their activity is being logged. The program of logging keyboard can be legally used as statistics software to analyze user behavior under the user's control. However, most keyboard logging spyware is often used for stealing passwords. Once the spy application has been working on the computer, it will start to record every keyboard input from the typical user and send collected information to remote server. These processes usually happen without user noticing anything. In the future, an artificial intelligence (AI)-based logging spyware may be used to predict the identity of the user by analysing the speed of typing, strike frequency of each key on the keyboard and mouse behaviors (Moskovitch et al., 2009).

- Social Engineering. This is a method of obtaining secret information by setting up psychological traps for victims' psychological weaknesses, instinctive reaction, curiosity, trust, and greed. Most social engineering attacks rely on real communication between attacker and victim. The goal is to invite victim to provide sensitive information voluntarily instead of using brute force methods to steal the data. The process can happen in a single email or long period conversation over chatting. For example, one attacker can pretend to be your boss to obtain your trust through email

system. Once attackers build trust with their victim, they will ask you to provide them certain financial support or bank account to help them (Algarni et al., 2013).

In order to overcome various security risks caused by static password, dynamic password authentication has gradually become the mainstream technology of password authentication. A dynamic password means the user's password is different each time when they log in. Each password is used only once, and therefore it is also called one-time-password (OTP). The dynamic password is widely used in fields that require high level security guarantees, such as financial, health and game industry.
An enhanced password-based authentication strategy is the one that verifies user's identity using combination of static and dynamic authentication. For example, a static password that the user has to remember and a one-time password that is delivered to the user separately each time. By using this smart authentication, the hacker who has stolen the user's password meets another wall on the way to enter the account.

Hardware based authentication Identity authentication based on hardware tokens includes two elements: hardware devices and personal identification number (PIN) codes. The Finnish ID card is one example of smart cards. A typical smart card is a thin plastic card that contains an 8-bit micro-controller which can execute machine instructions at a speed of 1 MIPS. A co-processor could be included to improve the speed of encryption and decryption. The PIN code is the password of smart card which sets up a second wall to prevent untrusted use if the card is stolen. To read the data in smart card, a correct PIN code needs to be entered through smart card reader. The PIN code is typically stored in ROM of chip which means it cannot be changed once it is set up. After user entered wrong PIN code several times, the smart card will be locked.
In some application environment with higher security requirements, simply using one authentication method is not enough. Two-factor authentication uses two independent methods to verify the identity of the user. It could be a combination of password and hardware that is kept separately from the login device. The hardware could be an USB device or smartcard which contains users personal login certificate. The advantage of using unwriteable hardware is that no matter whether the computer used to login to the system has malware or not, the data in the hardware token is safe (Thomasson and Baldi, 1997).

Biometric authentication The biometrics of human are stable and unique. Biometrics can be used to replace traditional methods to enhance the security level of authorization or

authentication. Biometrics use scienced-based means to measure characteristics of human beings. Our life is filled with situation where we need to prove who we are: access to email account, bank account transfer, opening your own car, traveling cross the boarder, visiting hospital. Thus, fast and reliable authentication is essential for preventing fraud.
Biological characteristics are mainly divided into physical and behavioural characteristics. Physiological characteristics are inherent characteristics of the human body. They are basically hard to change under human wishes, therefore they are objective factors. Fingerprint, face shape and geometry, DNA, retina, and iris of eye are examples of physiological characteristics. Behaviour characteristics which mainly include sound recognition and signature recognition are formed by people in the long-term life process. Biometric-based identification technology has many advantages over traditional identity authentication, such as confidentiality, convenience, better anti-counterfeiting, being able to carry and use at anytime. Many countries are also gradually embedded the holder's biometric information, such as fingerprint and photo in the personal identity card and passport (Huixian and Liaojun, 2009).
Common biometric authentication methods include:

- Fingerprint scanners.
  Fingerprint is one of the most accessible biometric feature of human. The fingerprint information of a person is formed about 9 months after birth. The probability of different people having the same fingerprint is extremely low. Fingerprint is collected from the pattern information of the end of finger from human being. It is believed that when 13 points from two fingerprints are matching, then those two fingerprints can be considered to belong to the same person.

- Eye scanners.
  The eye scanners include technologies like iris recognition and retina scanners. The structure of human iris is very complex, with more than 260 variable items. It is considered to be the most reliable biometric technology because iris hardly changes in a lifetime.

- Facial recognition.
  The face like other biological characteristics of human body is unique and difficult to copy. Face recognition is highly valuable in access control, identification, traffic control, human computer interaction. The traditional face recognition is based on camera sensor to detect face image. This is a well-known recognition method that has been developed for more than 30 years. However, this method has insurmountable

defects, the effectiveness of recognition will drop sharply when the ambient light is constantly changing. Thus, traditional face recognition cannot meet all needs of daily life. To overcome the shortcomings of traditional face recognition, three-dimensional (3D) image recognition is implemented nowadays. There are two kinds of techniques for capturing 3D face models. The activate acquisition including triangulation and structure light, and passive acquisition technologies by using a stereo camera. In the example of triangulation technology, the scanner emits laser light on the face and the camera can capture the reflect light from face. The position of the laser point is determined by measuring the triangulation from laser spot, laser emitter and camera. The Microsoft Kinect is another example by using structure light. The scanner projects a pattern on the face, and camera getting the pattern deformed by the face. The shape of face finally calculated based on the deformation of face pattern in real time (S. Zhou and Xiao, 2018).

4.3 Access control

In the field of computer security, access control is a security technique that the system makes a decision whether the authenticated users, applications, or devices have the permission to view, create, modify and remove the data and resources of organization. The Internet of Things system is a multi-user, multi-tasking environment, which opens the door to the illegal use of system resources. Therefore, it is required to adopt effective security precautions for IoT networks to prevent users from illegally using system resources.

Access control can control user access permissions for critical resources in the system according to user types and attributes to prevent illegal intrusion and use of system resources by legitimate users. Commonly used access control models include discretionary access control (DAC), mandatory access control (MAC), role-based access control (RBAC) and attribute based access control (ABAC) (A. Younis et al., 2014).

- Discretionary access Control (DAC) is the principle of restricting access to object(resource/data) based on the identity of the subject(user/human). DAC appeared earlier than other access control models and it is a very common access control strategy. The controls are discretionary in the sense that subjects can directly or indirectly grant access rights to other subjects. (Balamurugan et al., 2015)(Vasilyevna, 2008).

- Mandatory access control (MAC)
  The difference between MAC and DAC is that the management of access control changes from user to system. The system directly manages access control to ensure that the transmission of information in the entire network is under system's control. The feature of mandatory access control lies in its mandatory decisions which means the system and security administrators assign the security attributes of the subject and object with certain rules. It is not possible for users to change access control of resource.
  The mandatory access control strategy compares the security attributes of the subject and objects to determine whether allowing access or not. If the system determines that the access is not established due to the security attribute, no one can change the decision made by system. Therefore, even if there is a trojan virus, it cannot change the fact that the subject cannot access the object. The design and implementation of MAC is commonly used in an organization where needs extremely high security level such as government and military. (Balamurugan et al., 2015)(Vasilyevna, 2008).

- Role-based access control (RBAC)
  The concept of Role-based access control (RBAC) has been proposed from the 1990s. In the RBAC model, it introduces the concept of roles and permissions to make right control easier. The core idea of RBAC is that access permissions are not directly related to users. Instead, the concept of 'roles' is introduced. Users can select different roles to achieve different access to objects. The system can modify or delete the permissions of a role at will. The biggest advantage of RBAC is that it realizes the logical separation of users and permissions (Balamurugan et al., 2015)(Vasilyevna, 2008).

- Attribute based access control (ABAC)
  ABAC is also known as policy-based access control. It defines an access control paradign where subject requests to perform operations on objects are granted or denied based on the use of policies with assigned attributes of the subject. This model introduces the concept of complex Bollean rule set that can evaluate many different attributes. The attributes can be user attributes, resource attributes, objects. The most significat benefit is ABAC's user-friendly nature. The user profiles is easy-to-understand, and any authorised system managner can update. (Balamurugan et al., 2015)(Vasilyevna, 2008)

5 BlockChain Enhanced IoT Security

In the chapters three and four, we have discussed the threats and the security factors to be considered when designing an Internet of Things system. In this chapter, we discuss how to enhance the security of the Internet of Things by combining it with blockchain technology. We also discuss the role of blockchain technology in different secure aspects of the Internet of Things, such as system security, privacy protection and access control.

5.1 What is BlockChain

People often called blockchain as Distributed Ledger Technology (DLT), but those two concepts are not exactly the same. Actually blockchain is based on DLT but combines it with other advanced technology from computational to mathematical techniques. Technologies such as encryption and cryptography methods, digital signatures, hash algorithms, distributed (peer-to-peer) networks, proof-of-work mechanisms are used to ensure security of Bitcoin transactions. Blockchains are immutable digital ledger systems implemented in a distributed environment without central authority. The community of users can record transactions in a shared ledger which is public to that community. Immutability, decentralisation are significant advantages of blockchain technology (Kumar and Mallick, 2018).

Bitcoin is the decentralized digital currency without a central bank powered by blockchain. Let us take bitcoin as an example to better understand how blockchain works. As the name blockchain implies, it consist of blocks linked together into a chain. Every blocks contains the information about every transaction such as buyer and seller, timestamp, unique identifying code for each exchange and total value. Once a block is added to the blockchain, it becomes public to anyone who wishes to view it within the community. Blockchain is not stored in one place, instead, it distributed across multiple computers within network. All the nodes have same copy and, every copy is updated whenever there is a validated new change to the blockchain.

Bitcoin mining is the process of validating transactions and adding new block to the blockchain. The blockchain creates the block hash (SHA256), a 256-bit number generated

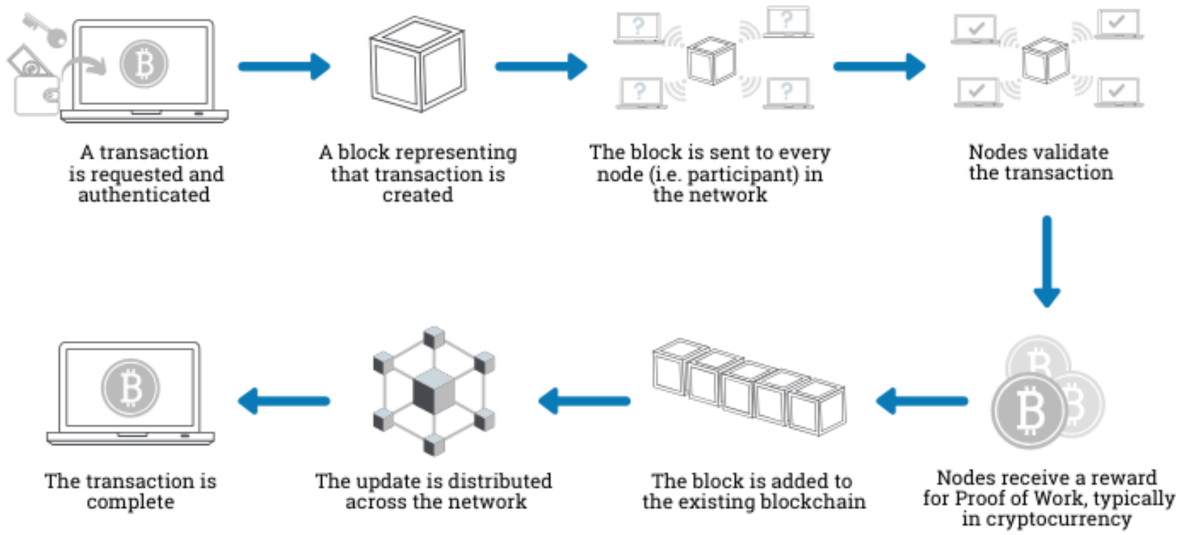

Figure 5.1: How does bitcoin work (Euromoney, 2020)

from the following information: block version, previous block's hash, transactions, block height number, timestamp, etc. A specific mining machine or general computer platform are deployed to generate a randomly hash number that matches the block hash. The miners compete to see which one will solve hash puzzles first, the winner will rewarded by Bitcoin after a new block is created on the chain. The process of mining consumes lots of computer resources and energy.

The workflow of Bitcoin is described in figure 5.1 (Euromoney, 2020). If node A transfers money to node B, a transaction request will be broadcast to all nodes in the network. Miners will pack all the transactions received in a certain period and start mining process. The first miner who solves a hash problem will broadcast the block to the entire network while other nodes verify the transaction using cryptographic algorithms. Finally, a new block contains the transaction is added to the end of existing blockchain. The features of blockchain are summarized as following subsections (Singh et al., 2018).

5.1.1 Decentralisation

Blockchain creates a shard public history of transactions packaged into blocks that are chained together to prevent tampering. Specifically, the data is packaged into blocks and all blocks are connected in chronological order to form a chain. From the perspective of data storage, blockchain can be summarised into two major characteristics: distributed data storage and immutability. The distributed data storage means the full data

of blockchain is not stored at a centralised node but stored at nodes distributed around the world. The chain does not belong to any computer or organisation. Each node has a complete copy of blockchain. At the same time, due to the use of cryptography in blockchain, even small modifications of any block can be detected by certain algorithms (Liu et al., 2020).

5.1.2 Consensus Mechanism

A consensus mechanism is a fault-tolerant mechanism that is used in blockchain system. This mechanism helps to solve the problem how blockchain keeps consistency under distributed scenarios. In any centralized system, like a database holding the citizen information in a country. A central administrator has the right to maintain and update the database such as adding, modifying and deleting records. On the contrast, blockchain that operate as decentralized systems has not any single administrator to achieve agreement, trust and security. In such a changing status of the blockchain, these publicly shared ledgers need an fair and secure mechanism to ensure that all transactions happening on the network are genuine. The most two common used mechanisms are Proof of Work (PoW) and Proof of Stake (PoS). PoW is a common consensus algorithm used by bitcoin and litecoin. (Gu et al., 2021).

5.1.3 Component of BlockChain

The system of blockchain is divided into the data layer, network layer, consensus layer, application and presentation layer. The data layer is based on cryptographic concepts such as hash algorithms, encryption algorithms, and Merkle trees. Some basic of components are introduced in the following.

- Account
  In Satoshi Nakamoto's original design, Bitcoin is an Unspent Transaction Output (UTXO) distributed in the network ledger. With the evolution of technology, the concept of account is used by more and more blockchain systems. The identity of each account is determined by private key, address and public key. The private key P_r is generated by hash (SHA256) from a randomly chosen numerical code. The corresponding public key P_u is generated by the Elliptic curve cryptography algorithm. The porcess of account creation is irreversible. The public key is used to

send cryptocurrency into a wallet, while the private key is used to prove ownership of a blockchain address. The private key should be kept secret as this is the only credentials to prove identity (Zhang and Lang, 2019; Lakshmi et al., 2019).

- Transaction
  In the blockchain ledger, the most basic data structure is transaction. The transaction records information about the deals of the participators, corresponding to the change of a certain value in the database. When an account creates a transaction, it needs to be signed with a private key. In essence, the signed transaction is just a piece of byte-code, the account broadcasts it to the network. Every nodes will verify transaction's signature to ensure its validity when nodes received transaction info. Finally, the transaction is packaged into blocks after confirmed by a mining node (Zhang and Lang, 2019; Kumar and Mallick, 2018).

Decentralization and encryption algorithms of Blockchain may be used for IoT security and trust issues. We will introduce some of usages in data storage, identity authentication, provable and traceable.

- Data storage
  The amount of IoT devices is continuously growing, and the data collected from them are becoming larger and larger. It is a big challenge to store massive data collected by sensors in most of centralised IoT system. Blockchain is a decentralised network and the nodes are completely equal. Using this feature can relief the storage problem that is created when all the data to be aggregated into a single server (Salman et al., 2019).

- Identity authentication
  The identity verification in the blockchain is implemented by using encrypted digital signatures and hashing. The decentralized identity recognition system is not controlled by any organization, which can ensure that the user has full control of their own identity information. Further, the blockchain verification and consensus mechanism helps to prevent illegal or malicious nodes from accessing the Internet of Things (Salman et al., 2019; Lakshmi et al., 2019).

- Provable and traceable
  It is almost impossible to modify block in blockchain, which means the data is immutable. It is difficult to tamper data in blockchain structure as long as the data

is written into the blockchain through consensus. Except for the first block, every other block contains the hash of the former one, and this feature can be used to trace the source of IoT operation.

5.2 Application

Here we discuss the role of blockchain technology in different secure aspects of IoT from current research articles, such as system security, privacy protection and access control.

5.2.1 System security

In 2017, Sharma (Sharma et al., 2017) proposed a distributed cloud architecture based on blockchain technology, using software defined network (SDN) to enable fog nodes controller at the edge of the network. The model includes the following four steps: selecting resource providers, providing services, registering transactions, and reward. In the first step, the cloud user must select a resource provider from a pool of service providers in a distributed cloud system based on the blockchain. Once a choice is made, the selected service provider will provide the user with the required services, such as task execution, data management, and server provision. After providing the requested service, the service provider registers transactions in the form of a blockchain and shares the transaction with all distributed peer-to-peer service providers. Finally, users will pay and reward provider. Blockchain provides low-cost, secure, and on-demand access to the most competitive computing infrastructure in IoT networks. By creating distributed Cloud infrastructure, this model achieves high performance and cost-effective computing. Kai (Fan et al., 2019) adopts an improved practical Byzantine fault tolerance (PBFT) consensus mechanism to achieve time synchronization of IoT devices. Thereby, this improved mechanism reducing external attacks and making the system efficient and safe. Minoli (Minoli and Occhiogrosso, 2018) pointed out that the deployment of IoT devices leads to increased number of attacks. The use of smart contracts in blockchain technology improves the security of the system. The blockchain mechanism has become a way of defence for the Internet of Things.

5.2.2 Privacy protection

Privacy protection refers to protecting the user's sensitive information, including identity information and personal account. We can protect our private data by changing the

authority policy when the sensitive data stored at a service provider. However, due to the existence of third parties, it is impossible to fully prevent information leakage. Blockchain technology contains different encryption algorithm to enhance the security and privacy in the IoT ecosystem. M.Singh (Singh et al., 2018) provides a model where use blockchain to protect the security of the Internet of Things. It introduced the Prove-of-work(PoW) consensus mechanism, PK (public key) to record the user's identity, while the private key is used to encrypt private data. Wang (Wang et al., 2018) analyzed the requirements of designing a decentralized PKI system for privacy and proposed a privacy-conscious blockchain-based PKI. In addition to a series of operations such as registration, revocation and restoration, they also introduced neighbour groups to improve the performance of privacy protection. Hardjono and Pentland (Hardjono and Pentland, 2019) introduce a blockchain-based privacy protection identity solution called ChainAnchor. In this scheme, the verified nodes have the authority to write or process transactions, and they are all built on tamper-resistant hardware to provide users with privacy protection services.

5.2.3 Access control

As a security mechanism, access control defines which resources and services in a computer system can be accessed. Traditional access control includes access control lists, roles based access control, attributes based access control. The traditional mechanism may not suitable to the current development of the Internet of Things. The introduction of blockchain technology makes the access control strategy transparent. X.Zhu (Zhu and Badr, 2018) proposed the FOCUS architecture. They used a three-dimensional social network to build a user-centric access control mechanism which can manage all types of access control. The entire access control mechanism is built on a blockchain-based identity in a trustless IoT environment. X.Zhu (Zhu and Badr, 2018) proposed a dynamic access control scheme to solve the problem of the direct data communication access control method between the devices in the dynamic environment of IoT. Ouaddah (Ouaddah et al., 2017) proposed the FairAccess framework. The advantage of this framework is the use of smart contracts to create decentralized pseudonyms and privacy protection authorization management frameworks. Prada-Delgado (Prada-Delgado et al., 2019) describes a novel zero-knowledge method for IoT devices, using the unique characteristics of microchips to obtain encryption, combined with blockchain for device identity verification.

6 Case study - Mirai

The IoT botnet is a new IoT security threat that combines virus, Trojan and worm.There is malware that can perform remote hijacking but also can act like worm to infect the whole network. This malware use to create and manage the botnets. Botnets are designed to infect millions of devices instead of just one single target. A more complex and intelligent botnet can self-propagate, search targets and infect devices automatically. Furthermore, the rapid development of Internet of Things has accelerated the spreading of botnets. The significant feature of IoT botnet is that the attacker can perform a series of attack operations on the botnet client without directly logging in to it. A large number of infected clients can complete tasks together, such as launching Distributed-Denial-of-Service (DDoS) attack on the same target (Mendes et al., 2019).

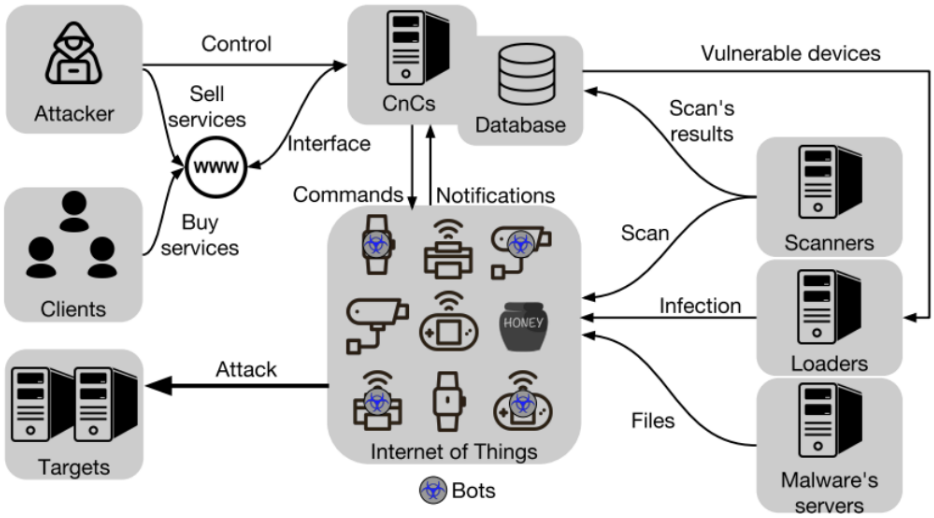

Figure 6.1: Overview of an IoT botnet (Marzano et al., 2018)

Figure 6.1 shows how the typical IoT botnet works. An IoT botnet network contains malware servers, infected clients (Bots), Command and Control Server (CnCs) and possibly some other components. Even a small botnet may have hundreds of infected clients. From the perspective of functional structure, botnet generally contains scanners module, loaders module, and other functional modules. The scanners module is used to scan the IoT devices and spread specific botnet viruses, then loaders module will hide within the IoT devices in a such way the user will not notice it. Finally, the botnets from infected

IoT devices are used to execute attack targets by receiving attacking command from Command and Control (CnC) Server (Han et al., 2012; Kolias et al., 2017).
Generally, attackers will use the vulnerabilities of the IoT system itself to crack IoT devices through phishing or automated cracking tools. Once the user name and password are successfully cracked, malicious code can be implanted from the botnet server to hijack the IoT device as an infected client. Botnets use a large number of zombie clients to launch flood attacks on other networks. At same time, Botnets have ability to infect other IoT systems to become new botnets (Han et al., 2012; Kolias et al., 2017).
The intrusion of IoT botnet is due to the security vulnerabilities of the IoT equipment where attackers have found effective methods to crack the devices against these vulnerabilities. The manufactures and users may not pay sufficient attention to security issues. For example, the device's password may never have been changed since the user started to use it. Table 6.1 from KrebsOnSecurity shows the most commonly used login passwords by IoT devices. As we can see from the list, most of devices in the list are networked-based cameras, printers and routers. Those devices are actually not representing just a single device but millions of devices that are sharing the same default password if the users have never changed it.

Username/Password	Manufacturer	Link to supporting evidence
admin/123456	ACTi IP Camera	https://ipvm.com/reports/ip-cameras-default-passwords-directory
root/anko	ANKO Products DVR	http://www.cctvforum.com/viewtopic.php?f=3&t=44250
root/pass	Axis IP Camera, et. al	http://www.cleancss.com/router-default/Axis/0543-001
root/vizxv	Dahua Camera	http://www.cam-it.org/index.php?topic=5192.0
root/888888	Dahua DVR	http://www.cam-it.org/index.php?topic=5035.0
root/666666	Dahua DVR	http://www.cam-it.org/index.php?topic=5035.0
root/7ujMko0vizxv	Dahua IP Camera	http://www.cam-it.org/index.php?topic=9396.0
root/7ujMko0admin	Dahua IP Camera	http://www.cam-it.org/index.php?topic=9396.0
666666/666666	Dahua IP Camera	http://www.cleancss.com/router-default/Dahua/DH-IPC-HDW4300C
root/dreambox	Dreambox TV receiver	https://www.satellites.co.uk/forums/threads/reset-root-password-plugin.101146/
root/zlxx	EV ZLX Two-way Speaker?	?
root/juantech	Guangzhou Juan Optical	https://news.ycombinator.com/item?id=11114012
root/xc3511	H.264 - Chinese DVR	http://www.cctvforum.com/viewtopic.php?f=56&t=34930&start=15
root/hi3518	HiSilicon IP Camera	https://acassis.wordpress.com/2014/08/10/i-got-a-new-hi3518-ip-camera-modules/
root/klv123	HiSilicon IP Camera	https://gist.github.com/gabonator/74cdd6ab4f733ff047356198c781f27d
root/klv1234	HiSilicon IP Camera	https://gist.github.com/gabonator/74cdd6ab4f733ff047356198c781f27d
root/jvbzd	HiSilicon IP Camera	https://gist.github.com/gabonator/74cdd6ab4f733ff047356198c781f27d
root/admin	IPX-DDK Network Camera	http://www.ipxinc.com/products/cameras-and-video-servers/network-cameras/
root/system	IQinVision Cameras, et. al	https://ipvm.com/reports/ip-cameras-default-passwords-directory
admin/meinsm	Mobotix Network Camera	http://www.forum.use-ip.co.uk/threads/mobotix-default-password.76/
root/54321	Packet8 VOIP Phone, et. al	http://webcache.googleusercontent.com/search?q=cache:W1phozQZURUJ:community.freepbx.org/t/packet8-atas-phones/411
root/00000000	Panasonic Printer	https://www.experts-exchange.com/questions/26194395/Default-User-Password-for-Panasonic-DP-C405-Web-Interface.html
root/realtek	RealTek Routers	
admin/1111111	Samsung IP Camera	https://ipvm.com/reports/ip-cameras-default-passwords-directory
root/xmhdipc	Shenzhen Anran Security Camera	https://www.amazon.com/MegaPixel-Wireless-Network-Surveillance-Camera/product-reviews/B00EB6FNDI
admin/smcadmin	SMC Routers	http://www.cleancss.com/router-default/SMC/ROUTER
root/ikwb	Toshiba Network Camera	http://faq.surveillixdvrsupport.com/index.php?action=artikel&cat=4&id=8&artlang=en
ubnt/ubnt	Ubiquiti AirOS Router	http://setuprouter.com/router/ubiquiti/airos-airgrid-m5hp/login.htm
supervisor/supervisor	VideoIQ	https://ipvm.com/reports/ip-cameras-default-passwords-directory
root/<none>	Vivotek IP Camera	https://ipvm.com/reports/ip-cameras-default-passwords-directory
admin/1111	Xerox printers, et. al	https://atyourservice.blogs.xerox.com/2012/08/28/logging-in-as-system-administrator-on-your-xerox-printer/
root/Zte521	ZTE Router	http://www.ironbugs.com/2016/02/hack-and-patch-your-zte-f660-routers.html

Table 6.1: Default username/password used by IoT device. (KrebsOnSecurity, 2016b)

6.1 Type of Botnet

Botnets can be divided into two types based on communication and control (CnC) protocols. These types are commonly called first and second generations. The first generation uses IRC and HTTP protocols for its CnC channels. The use of IRC channel provides faster response time which enables the always-on connection between botmaster and client. However, the HTTP protocol has been increasingly used during recent years since anti-malware companies have developed efficient ways to block the IRC botnets. By using HTTP protocol, the low-volume C&C traffic is hidden under the high volume of HTTP packages, which makes it harder for firewalls to detect such traffic. Still, centralised protocols such as IRC and HTTP allow detection and disabling of botnets relatively easily. The second generation is based on peer-to-peer (P2P) protocol and it overcome the shortcoming of single point of failure (Mendes et al., 2019). Thus, P2P protocol has been widely used in modern botnet attacks. In the following, the details of each of the above protocols will be explained.

- Internet Relay Chat (IRC) protocol
  The IRC protocol is a real-time network chat protocol that was widely used in the early days of the Internet. It enables Internet users from all over the world to join chat channels for text-based real-time discussions. The IRC protocol is based on the client-server model. The most notable feature of IRC is that it provides a channel where users can communicate with each other in unlimited fashion. The messages sent by each client to the IRC server will be forwarded to all clients connected to the channel. At the same time, the IRC protocol provides private chats between two users supported by two extended protocols, which are Direct Client-to-Client (DCC) (Irchelp.org, 2016) and Client-to-Client Protocol (CTCP). Since the IRC protocol provides a simple, low-latency, anonymous real-time communication method, it is commonly used by hackers for remote communication. In the early stage of the development of botnets, the IRC protocol naturally became a main method to control one-to-many streams. (Mendes et al., 2019).

- HTTP protocol
  The HTTP protocol is another botnet command and control protocol that has been popular in recent years. Compared with the IRC protocol, the advantages of using the HTTP protocol to build a botnet communication mechanism include two aspects:

- Botnets are more hidden and more difficult to detect.
  Since the IRC protocol used to be the main control protocol for botnets, the security industry has paid most attention to monitoring IRC communications to detect hidden botnet activities. Using the HTTP protocol to build a control channel allows the botnet control traffic to be submerged in a large number of Internet Web communications, thus making it more difficult to detect botnet activities based on the HTTP protocol (Mendes et al., 2019).

- Control flow can bypass the firewall.
  Most organizations deploy a firewall between organisation's network and internet. In many cases, the firewall filters out network communications on undesired ports. The ports used by the IRC protocol are usually filtered. On the other hand, a control channel built on HTTP protocol can bypass through the firewall. This is because the HTTP protocol is commonly used for many services.

- Peer-to-peer (P2P) protocols
  Compared to botnet that is based on distributed peer-to-peer architecture, the centralised control protocols such as IRC and HTTP make botnet based on client-server architecture easier to be tracked and detected. Once the defender observes the botnet, they can easily find the location of the botnet server and they are able to track the activities of the botnet. In order to make botnets more resilient and concealed, newly emerged bots have begun to use P2P protocols to build their controlling network. Botnets such as Sinit, Phatbot, SpamThru, Slapper, Nugache have implemented various P2P mechanisms. Some of those botnets show advanced design ideas (Zeidanloo et al., 2010; Mendes et al., 2019).

6.2 Mirai Botnet

There are huge number of viruses designed to attack IoT network, such as QBOT, Luabot, KTN-RM. The Mirai is the best known and most influential among them. In September 2016, hacker used name "Anna-senpai" released the source code of Mirai malware on the Github (KrebsOnSecurity, 2017). The DNS provider DyN in the United States suffered the most serious DDoS attack in history on October 2016. It has been confirmed that the source of attack was from devices infected by the Mirai virus. In early October 2016,

researchers from Imperva Incapsula analysed 49657 infected devices, and they found that majority of infected devices were CCTV cameras, DVRs, and routers. The IP addresses of these infected devices pointed to 164 countries where Vietnam (12.8%), Brazil (11.8%), United States (10.9%) and China (8.8%) shares big part of infected IoT devices (Ben, 2016). Analysis of Flashpoint and Akamai (KrebsOnSecurity, 2016a) confirmed that one of the sources of attack traffic was from devices infected with Mirai bots. The AdLab pointed that Mirai was using part of technology from QBOT to optimize the scanning and infection speed. (Kolias et al., 2017).

6.2.1 Working flow of Mirai

At present, most IoT botnets are derived from Mirai variants, their workflows are basically the same as in original Mirai. In this section, we will use Mirai as an example to analyse the principles of a botnet attack. The figure 6.2 shows the infection and attack process of Mirai.

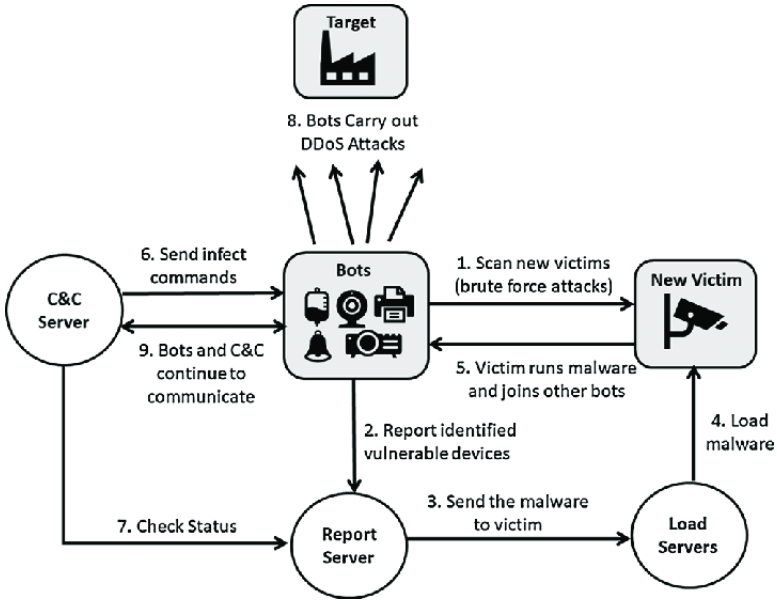

Figure 6.2: Mirai botnet operations. (Tuptuk and Hailes, 2018)

The Bot uses Telnet or SSH to make a brute force search for weak passwords throughout IoT devices on the internet (1). The login info which contains username, password, IP address and port number from attacked devices will pass to the report server for analysis (2). The report server will distribute the information about the vulnerable devices to load servers (3). Load servers uses this information to pull a downloader helper through one of

three methods: echo, wget or Trivial File Transfer Protocol (TFTP). Finally the Loader on the infected device downloads the malicious Bot program from the Mirai load server (4). Once the device runs the malware it becomes a new bot (5), and receives commands from the command and control server (6). The bot program on the infected device begins to scan the devices on the internet through a random strategy. Unlike ordinary botnets, Mirai's infection process can not only be initiated by the attacker's server, but can also be initiated by infected devices (5). Once the IoT device is infected by the botnet, the Bot module in the infected device will start to scan other IoT devices. This infection method is extremely effective (Tuptuk and Hailes, 2018).

6.2.2 Modules of Mirai

According to Mirai's source code (jgamblin, 2016), it can be divided into the following three modules:

- Bot module: The module is written by C programming language. Once it has been started, Mirai will delete its exe file, and keep running on the RAM only. This main program of Mirai, running in the infected device, receive attack command issued from CnCs server. Besides the main module, it has three sub-modules, with following tasks:
  - Attack: Attack module can parse ten different attack methods from ten different functions. When the module receives the command of attack, the module will decide which attack to launch.
  - Scanner: Scanner module continuously scans the randomly generated IP addresses to check the possible vulnerable IoT devices by telnet. The usernames and passwords are picked from the table which contains most common factory default combinations. If the telnet is successful, the scanner module will send observed usernames and passwords of the devices to the report servers.
  - Killer: Killer module is running at background in order to prevent Mirai from being killed by other worms running on the same device. Firstly, it closes processes holding ports using Telnet (23), SSH (22), and HTTP (80) and reserves these ports. The other function is to delete a specific file and kill the corresponding process to achieve monopoly. The purpose of this function is to maximize the controllability of system resources. (De Donno et al., 2018; Sinanović and Mrdovic, 2017).

- Command-and Control (CNC) module: CNC module is written in GO language. It manages the infected devices and issues DDoS attacks to the zombie clients. There are two types of account in the CnC servers, Admin and User. Each of sub-account is able to perform different operation level according to its connected port.
  - Admin: Admin has highest privilege right. It can add new users(account) to the system, report available zombies client to the CnC server, and schedule new attacks.
  - User: After certain clients buy the services from attackers through internet or dark-net, they will get an user account. This account can launch limited attacks and its permissions are limited by the administrator.
- Loader module: The loader module creates a server for downloading payloads using wget, echo, or TFTP from busybox. After that, it become as a reporting server to receive information of vulnerable IoT devices (De Donno et al., 2018; Sinanović and Mrdovic, 2017).

6.2.3 Highlights of Mirai

In addition to the characteristics of traditional botnet, Mirai has some exceptional aspects which deserve special attention. Through the study of Mirai source code, we gain insight into the characteristics of this type of botnet. The following summarizes the six unique characteristics of Mirai (jgamblin, 2016; Ben, 2016) and the source code was acquired from the following GitHub repository:
https://github.com/rosgos/Mirai-Source-Code.

- Process mask: In order to prevent the name of process from being exposed, Mirai deletes itself from the file system and hides its name with a random string.

```
// Hide process name
    name_buf_len = ((rand_next() % 6) + 3) * 4;
    rand_alphastr(name_buf, name_buf_len);
    name_buf[name_buf_len] = 0;
    prctl(PR_SET_NAME, name_buf);
```

- Exclusivity: Once the device is infected, it will close the Telnet (23), SSH (22), HTTP (80) services and prevents these services from restarting. The killer module will be used to forcibly close these three processes, and prevent these three ports from restarting again.

```
 // Kill telnet service and prevent it from restarting
    #ifdef KILLER_REBIND_TELNET
        #ifdef DEBUG
        printf("[killer] Trying to kill port 23\n");
        #endif
    if (killer_kill_by_port(htons(23)))
        {
    #ifdef DEBUG
        printf("[killer] Killed tcp/23 (telnet)\n");
    #endif
// Kill SSH service and prevent it from restarting
......
// Kill HTTP service and prevent it from restarting
......
```

- Territorial predator: Mirai will use a memory scraping technology to kill other malware in a same device. Mirai searches the memory of device for the features of QBOT, UPX, Zollard, and Remaiten bot. After that, it will kill these opponents' processes in order to achieve the purpose of monopolizing resources. This aggressive behavior helps Mirai maximize the usage of resources and attack potential from the botnet devices.

```
    #DEFINE TABLE_MEM_QBOT            // REPORT %S:%S
    #DEFINE TABLE_MEM_QBOT2           // HTTPFLOOD
    #DEFINE TABLE_MEM_QBOT3           // LOLNOGTFO
    #DEFINE TABLE_MEM_UPX             // \X58\X4D\X4E\X4E\X43\X50\X46\X22
    #DEFINE TABLE_MEM_ZOLLARD         // ZOLLARD
```

- Avoid attacking sensitive targets: Mirai filters out the IP addresses of companies and

institutions such as General Electric, Hewlett-Packard, US national postal service and Department of Defense to prevent unwanted infections.

```
    while
(o1 == 127 ||         // 127.0.0.0/8      - Loopback
(o1 == 0) ||          // 0.0.0.0/8        - Invalid address space
(o1 == 3) ||      // 3.0.0.0/8            - General Electric Company
(o1 == 15 || o1 == 16) || // 15.0.0.0/7   -Hewlett-Packard Company
(o1 == 56) ||     // 56.0.0.0/8           - US Postal Service
(o1 == 10) ||   // 10.0.0.0/8             - Internal network
(o1 == 192 && o2 == 168) ||   // 192.168.0.0/16   - Internal network
(o1 == 172 && o2 >= 16 && o2 < 32) ||    // 172.16.0.0/14  - Internal network
(o1 == 100 && o2 >= 64 && o2 < 127) ||   // 100.64.0.0/10  - IANA NAT reserved
(o1 == 169 && o2 > 254) ||                // 169.254.0.0/16 - IANA NAT reserved
(o1 == 198 && o2 >= 18 && o2 < 20) ||    // 198.18.0.0/15  - IANA Special use
(o1 >= 224) ||                            // 224.*.*.*+     - Multicast
(o1 == 6 || o1 == 7 || o1 == 11 || o1 == 21 || o1 == 22 || o1 == 26 ||
o1 == 28 || o1 == 29 || o1 == 30 || o1 == 33
|| o1 == 55 || o1 == 214 || o1 == 215) // Department of Defence
    );
```

- There are more than 60 username and password combinations built in Mirai already. An example of combinations of admin/password is shown below.

```
add_auth_entry("\x50\x4D\x4D\x56", "\x5A\x41\x11\x17\x13\x13", 10);
// root/xc3511
add_auth_entry("\x50\x4D\x4D\x56", "\x54\x4B\x58\x5A\x54", 9);
// root/vizxv
add_auth_entry("\x50\x4D\x4D\x56", "\x43\x46\x4F\x4B\x4C", 8);
// root/admin
add_auth_entry("\x43\x46\x4F\x4B\x4C", "\x43\x46\x4F\x4B\x4C", 7);
// admin/admin
add_auth_entry("\x50\x4D\x4D\x56", "\x1A\x1A\x1A\x1A\x1A\x1A", 6);
// root/888888
```

The list of these username and password pairs are encrypted by XOR operations.

6.3 IoT design and defence

From the analysis of Mirai attack in the previous section, it can be seen that attacks using IoT botnet to break IoT terminal devices and launch DDoS attacks can be divided into three stages.
In the first stage, the attacker uses port scanning tool to find an IoT device which is exposed to the internet. The protocols SSH (port 22), Telnet (port 23) and HTTP/HTTPS (Port 80/443) become the breakthrough points for botnet to infect equipment.
In the second stage, the bot continues to penetrate the terminal and discover whether weak passwords are used in the device. Because of user negligence, the username/password may not have changed since device was manufactured, which enables the bot to crack the device using brute force dictionary attack.
In the third stage, the terminal becomes a part of the botnet controlled by the attacker, receiving instructions from command and control (CnC) server to launch at attack.
From the review of the process of Mirai attack above, the keys to design the security protection strategies for IoT terminals are to defend against the port scanning in the first stage and the brute force dictionary cracking in the second stage. It is difficult to find a protection solution if the terminal already reach the third stage. This section will focus on port scanning and brute force cracking.

6.3.1 Protection against port scanning

Port scanning is an important step for the attacker to collect information about the target. Usually, the ports of the target host are scanned to determine which ports are open. Attackers can guess the services enabled by the target host from the open ports, and then find possible vulnerabilities in the target host. Common port scanning methods include full TCP connection scanning, SYN scanning, ACK scanning, and UDP scanning. The botnet viruses often use SYN scanning method when scanning the available ports of the device. The figure 6.3 shows the working steps of SYN flooding attack.

1. The hostile client will send a connection request synchronization (SYN) packet to the port of server, as if to establish a three-way handshake to the server.

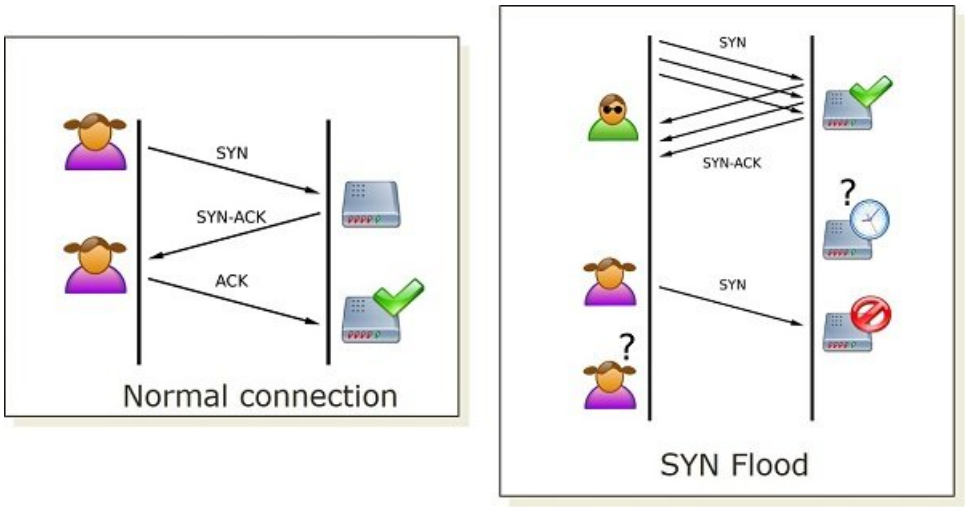

Figure 6.3: SYN Flood Attack (Thewindowsclub, 2020)

2. The server will reply to the scanning host with SYN-ACK confirmation packet.

3. After hostile client receives the confirmation packet, it will keep slient instead of an acknowledgment (ACK) response. The hostile client could also spoof the source IP address in step (1) in which case the server sends the SYN-ACK to a fake IP address. (Thewindowsclub, 2020).

A SYN flood is a form of denial-of-service attack in which the attacker establish a connection to a server without finalizing the connection. Thus, the server has to wait half-opened connections, which consumes computing resources from the server.

6.3.2 Snort and Iptables

Snort is a powerful and lightweight network intrusion detection system (NIDS), which can detect a variety of different attacks and provide real-time alarms on attacks (Rehman and Regina, 2003; Baker and Esler, 2007). In addition, Snort has good scalability and portability. Any organization and individual that complies with GNU General Public License (GPL) can freely use it. Snort has the ability of real-time traffic analysis and logging of IP network data packets, so that snort can quickly detect network attacks and issue alarms in time. Snort's alarm mechanism is very rich, including Syslog, user-specified files, a Unix socket, and Samba protocol to send WinPoup messages to Windows clients. Using the XML plug-in, Snort can use Simple Network Markup Language (SNML) (Rehman and Regina, 2003) to store the log in a file. Snort with XML plug-in can detect various attacks, such as buffer overflow, Common Gateway Interface (CGI) attack, port brute force

cracking, Server Message Block (SMB) and web application attacks. The Snort system is generally composed of two parts: Rule set and Snort Intrusion Detection System (IDS) (Z. Zhou et al., 2010).

1) Snort rule set

The snort rule set is a database which contains the attack characteristics created by several research teams according to the previous behavior of attacks. Each rule is an identifier for a particular attack type. Snort uses rule sets to identify attack behaviours and generates alerts for users. Snort rules are distributed in two sets: The "Community Ruleset" and the "Snort Subscriber Ruleset." (Snort.org, 2021)

2) Snort executable program

The executable program is composed of four important subsystems, see figure 6.4: Packet Decoder (Sniffer), preprocessor, detection engine, and log/alarm subsystem (Output) (Z. Zhou et al., 2010; Firoz et al., 2020).

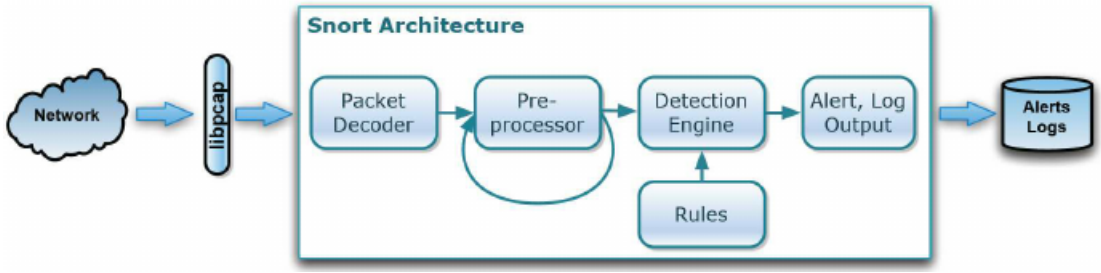

Figure 6.4: Workflow and architecture of Snort (Schütte et al., 2012)

- Packet Decoder (Sniffer)
  Decoding is the first process for a packet that goes through Snort. The job of the decoder is to determine which basic protocols (Ethernet, IP, TCP, etc.) are used in the data packet.

- Preprocessors
  The module uses the corresponding plug-in to check the original data packet, discovers the purpose of the original data such as port scanning, IP fragmentation, etc. Preprocessors module works in following ways:

  1. Prepare data for detection engine.
  2. Detect anomalies in packet headers:

3. Packet defragmentation: The smaller pieces of data packages are re-assembled by the receiving system to form the original data packet.
4. Decode HTTP URI: Detect Unicode characters inserted into the Uniform Resource Identifier (URI) which the browser regard as legal codes.

- Detection Engine
  Detection Engine is the core module of Snort. When the data packet is sent from the preprocessor, the detection engine checks the data packet according to the preset rules. Once it finds that the content in the data packet matches a certain rule, it will notify the alarm module.

- Logging and Alerting System
  The data examined by the detection engine needs to be output in some way. If a rule in the detection engine is matched, an alarm will be triggered. This alarm information will be sent to the log file using trap commands of the network, UNIX socket, Windows Popup message, and Simple Network Management Protocol (SNMP) protocol. The alarm can be sent to a third party Plug-ins (such as SnortSam). It can also be recorded in the SQL database (Firoz et al., 2020; G.-z. Zhou and J.-y. Li, 2012).

Although the Snort has flexible rules setup and great power to detect port scanning, it cannot take active defence role against attacks. When considering the shortcomings of Snort, the idea of associating it with Iptables firewall in Linux system has been raised (Guerrero and R. Gomez, 2005). Similarly as a firewall based on security rules, the Iptables can implement network packet filtering, network address translation (NAT), and datagram processing. However, because its security rules are preset static rules, Iptables does not have the ability to dynamically respond to intrusions (Mirzaie et al., 2010). Therefore, the method of linking Snort and Iptables can overcome their respective shortcomings and form a defence system with both intrusion detection and attack protection capabilities. The overall workflow is shown in the figure 6.5.

All external data will be inspected by Snort detection engine. If the data traffic matches to the behaviour of defined SYN port scanning, Snort will output the alarm information to the alarm processing module for analysing. When the analysis module detects new alarm information in the alarm log, it will analyse and extract various information related to the suspected attack. The module may generate a new rule based on the behaviour of attack

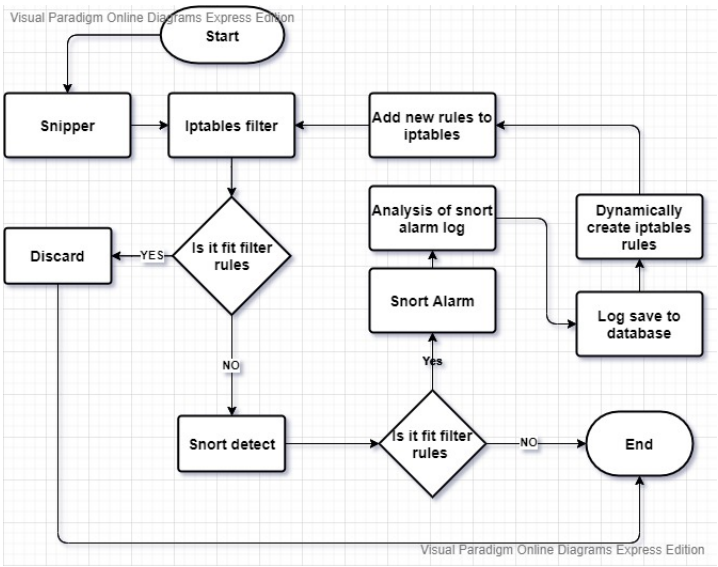

Figure 6.5: Workflow of Snort-Iptables (Guerrero and R. Gomez, 2005)

and write it into Iptables filter library. Thus, when the attacker returns, the data packets will be filtered by Iptables to ensure network security.

6.3.3 Brute-force attack

Brute-force attack means that attacker is systematically trying all combinations of usernames and passwords to check whether they can log in to the system. Most of IoT botnets, including Mirai botnets, use Telnet brute-forcing in the process of infecting IoT devices. If the Telnet port is enabled on the device, Mirai will attempt to log in to the device with built-in password dictionary. The following list summarizes the points to protect against the brute-forcing attack (Bošnjak et al., 2018).

- Enhancing the length and complexity of usernames and passwords can force the attacker to consume more time on guessing the passwords. However, this requires manufactures of IoT devices and users to have a higher awareness of the security issues. Furthermore, the management of large number of usernames and passwords for different equipment is a challenge.
- Limit the number of password attempts in the system.
- During password verification, result is returned after some delay.
- Limit the scope of clients allowed to initiate requests.

- The system administrator will be notified by Email after large number of password attempts from the same source.

Some commonly used approaches to defend against brute-force cracking are difficult to use against Mirai. For example, increasing the complexity of device usernames/passwords requires manufacturers and users to have a high level of security awareness, and may require users to modify passwords regularly. Changing the port number for disabling remote services such as SSH and Telnet can limit small-scale brute force attacks. However, this kind of defence has only small effect against large-scale attacks launched by botnets like Mirai. Moreover, disabling the port also limits the services that the device can provide.
For the purpose of defending against Mirai's attack, iptables firewall can be used as a tool to prevent Mirai brute force cracking attacks. Within a certain period of time, iptables can limit the number of login attempts to SSH, Telnet and other services. Brute force cracking attacks obtain a certain success rate through a large number of attempts. Frequent login attempts will leave a record of login failures on the target device. This can be used as a basis for detecting such brute force cracking attacks. When an attacker attempts to log in to SSH, Telnet and other services, as soon as the preset number of login failures is reached within a certain period of time, the iptables will block the IP addresses from suspected attacker where the many login attempts have arrived. In order to avoid false blocking of legitimate users and system administrators, a whitelist of legitimate users or system administrators should be added. The IP addresses in the whitelist can have an unlimited number of login attempts while IP addresses in the non-whitelist can only have a certain number of login attempts within the set time, and after that, the attempts will be blocked.

In this chapter, we first discussed the Internet of Things botnet viruses represented by Mirai. We conclude that the main methods of such viruses infecting Internet of Things terminal devices are port scanning and brute force dictionary cracking. The method of defence against Mirai attack is implemented as follows. Use snort and iptables linked approach to prevent IoT botnet from finding IoT devices by port scanning. The second discussion was about enhancing the complexity of the password to increase the difficulty of Mirai brute force cracking.

7 Conclusion

The purpose of this thesis is to find security issues and defence principles for the IoT platform. This was done by first looking at various cyber security threats and their evolution on the internet. We discussed the security issues of IoT platform in a three layer model, which consists of perception layer, network layer and application layer. In the perception layer, it is hard to implement a single security protocol to meet the demands of security requirements because different nodes have their own hardware configurations. Network layer is responsible for keeping safe and reliable transmission of information from perception layer to application layer. The design of security architecture on network may prioritize efficiency and compatibility. Application layer is directly interacting with users, it encounters challenges in reliability and security which relate to control and management, middleware and privacy protection.
In the defence principles, data encryption, user authentication and access control are put forward to secure IoT platforms. When taking the blockchain technology into practice, we discuss how to enhance the security of IoT by combining it with blockchain. Blockchain is based on distributed ledger technology but combines it with other state-of-art technologies from computational and mathematical techniques, such as decentralization, Byzantine fault tolerance consensus mechanisms (Fan et al., 2019) and smart contracts (Ouaddah et al., 2017).

The case study of Mirai is conducted in the last section of the thesis. From the studies, we analyzed the working flow of Mirai and functional modules of Mirai. Some highlights of Mirai, for example, process mask, exclusivity and territorial predator, are discussed as well. After the analysis of the Mirai attack, we find that attacks using an IoT botnet to break an IoT terminal device can be divided into three stages. In the first stage, the attacker uses a port scanning tool to find IoT devices exposed to the internet. In the second stage, the bot continues to penetrate the terminal by guessing the password. In the third stage, the IoT device becomes a part of the botnet controlled by the attacker and it starts receiving instructions from the Command and Control server of the botnet.

The method of defence against Mirai attack is to implement snort and iptables based approach to prevent port scanning. This combination can overcome shortcomings that exist if either snort or iptables is used alone and form a defence system with intrusion

detection and attack capabilities.

Bibliography

A. Younis, Y., Kifayat, K., and Merabti, M. (2014). "An access control model for cloud computing". In: *Journal of Information Security and Applications* 19.1, pp. 45–60. ISSN: 2214-2126. DOI: https://doi.org/10.1016/j.jisa.2014.04.003. URL: https://www.sciencedirect.com/science/article/pii/S2214212614000222.

Algarni, A., Xu, Y., Chan, T., and Tian, Y.-C. (2013). "Social engineering in social networking sites: Affect-based model". In: *8th International Conference for Internet Technology and Secured Transactions (ICITST-2013)*, pp. 508–515. DOI: 10.1109/ICITST.2013.6750253.

Ali B.; Awad, A. (2018). "Cyber and Physical Security Vulnerability Assessment for IoT-Based Smart Homes". In: *Sensors* 18(3),817, pp. 1–17.

Ashton, K. (1999). *Beginning the Internet of Things*. Blogpost. URL: https://medium.com/@kevin_ashton/beginning-the-internet-of-things-6d5ab6178801. (accessed: 22.5.2022).

Baker, A. R. and Esler, J. (2007). "Chapter 1 - Intrusion Detection Systems". In: *Snort Intrusion Detection and Prevention Toolkit*. Rockland: Syngress, pp. 1–30. ISBN: 978-1-59749-099-3. DOI: https://doi.org/10.1016/B978-159749099-3/50006-9. URL: https://www.sciencedirect.com/science/article/pii/B9781597490993500069.

Balamurugan, B., Shivitha, N. G., Monisha, V., and Saranya, V. (2015). "Survey of access control models for cloud based real-time applications". In: *International Confernce on Innovation Information in Computing Technologies*, pp. 1–6. DOI: 10.1109/ICIICT.2015.7396065.

Ben Igal, D. (2016). *Breaking Down Mirai: An IoT DDoS Botnet Analysis*. Blogpost. URL: https://www.imperva.com/blog/malware-analysis-mirai-ddos-botnet/. (accessed: 22.5.2022).

Bill Gates, w. N. M. and Rinearson, P. (1995). *The Road Ahead (Gates book)*. Viking Penguin.

Bormann, C. and Gomez, C. (2016). *Terminology for Constrained-Node Networks*. RFC document. URL: https://www.hjp.at/doc/rfc/rfc7228.html. (accessed: 22.5.2022).

Bošnjak, L., Sreš, J., and Brumen, B. (2018). "Brute-force and dictionary attack on hashed real-world passwords". In: *2018 41st International Convention on Information and Com-*

munication Technology, Electronics and Microelectronics (MIPRO), pp. 1161–1166. DOI: 10.23919/MIPRO.2018.8400211.

Burhan, M., Rehman, R. A., Khan, B., and Kim, B.-S. (2018). "IoT elements, layered architectures and security issues: A comprehensive survey". In: *Sensors* 18.9,2796, pp. 1–37.

Camarinha-Matos, L., Goes, J., Gomes, L., and Martins, J. (Jan. 2013). "Contributing to the internet of things". In: 394, pp. 3–12.

Chaudhary, S., Johari, R., Bhatia, R., Gupta, K., and Bhatnagar, A. (2019). "CRAIoT: Concept, Review and Application(s) of IoT". In: *2019 4th International Conference on Internet of Things: Smart Innovation and Usages (IoT-SIU)*, pp. 1–4. DOI: 10.1109/IoT-SIU.2019.8777467.

Chen, C. J., Wang, Y. W., and Fang, W. P. (2014). "A Study on Captcha Recognition". In: *2014 Tenth International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, pp. 395–398. DOI: 10.1109/IIH-MSP.2014.105.

Cloudflare (2020). *How does public key encryption work?* Web article. URL: https://www.cloudflare.com/en-gb/learning/ssl/how-does-public-key-encryption-work/. (accessed: 22.5.2022).

Conklin, A., Dietrich, G., and Walz, D. (2004). "Password-based authentication: a system perspective". In: *37th Annual Hawaii International Conference on System Sciences, 2004. Proceedings of the*. IEEE, pp. 10–27.

Darwish, M., Ouda, A., and Capretz, L. F. (2013). "Cloud-based DDoS attacks and defenses". In: *International Conference on Information Society (i-Society 2013)*, pp. 67–71.

De Donno, M., Dragoni, N., Giaretta, A., and Spognardi, A. (2018). "DDoS-capable IoT malwares: Comparative analysis and Mirai investigation". In: *Security and Communication Networks* 2018, pp. 1–30.

Delfs, H. and Knebl, H. (2015). "Symmetric-key cryptography". In: *Introduction to Cryptography*. Springer, pp. 11–48.

Euromoney (2020). *How does a transaction get into the blockchain?* Web article. URL: https://www.euromoney.com/learning/blockchain-explained/how-transactions-get-into-the-blockchain. (accessed: 22.5.2022).

Fan, K., Wang, S., Ren, Y., Yang, K., Yan, Z., Li, H., and Yang, Y. (2019). "Blockchain-Based Secure Time Protection Scheme in IoT". In: *IEEE Internet of Things Journal* 6.3, pp. 4671–4679.

Firoz, N. F., Arefin, M. T., and Uddin, M. R. (2020). "Performance Optimization of Layered Signature Based Intrusion Detection System Using Snort". In: *Cyber Security and Computer Science*. Ed. by T. Bhuiyan, M. M. Rahman, and M. A. Ali. Cham: Springer International Publishing, pp. 14–27.

Gu, W., Li, J., and Tang, Z. (2021). "A Survey on Consensus Mechanisms for Blockchain Technology". In: *2021 International Conference on Artificial Intelligence, Big Data and Algorithms (CAIBDA)*, pp. 46–49. DOI: 10.1109/CAIBDA53561.2021.00017.

Guerrero, J. and Gomez, R. (2005). "An example of communication between security tools: Iptables - Snort". In: *Operating Systems Review* 39, pp. 34–43. DOI: 10.1145/1075395.1075398.

Hameed, A. and Alomary, A. (2019). "Security Issues in IoT: A Survey". In: *2019 International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies (3ICT)*, pp. 1–5. DOI: 10.1109/3ICT.2019.8910320.

Han, F., Chen, Z., Xu, H., and Liang, Y. (2012). "A Collaborative Botnets Suppression System Based on Overlay Network". In: *The International Journal of Security and Networks* 7, pp. 211–219. DOI: 10.1504/IJSN.2012.053459.

Hardjono, T. and Pentland, A. (2019). "Verifiable anonymous identities and access control in permissioned blockchains". In: *arXiv preprint arXiv:1903.04584*.

Huixian, L. and Liaojun, P. (2009). "A Novel Biometric-Based Authentication Scheme with Privacy Protection". In: *2009 Fifth International Conference on Information Assurance and Security*. Vol. 2, pp. 295–298. DOI: 10.1109/IAS.2009.304.

Irchelp.org (2016). *A description of the DCC protocol*. Specification doc. URL: https://www.irchelp.org/protocol/dccspec.html. (accessed: 22.5.2022).

jgamblin (2016). *Leaked Mirai Source Code for Research/IoT Development Purposes*. Source code. URL: https://github.com/jgamblin/Mirai-Source-Code. (accessed: 22.5.2022).

Jing, Q., Vasilakos, A. V., Wan, J., Lu, J., and Qiu, D. (2014). "Security of the Internet of Things: perspectives and challenges". In: *Wireless Networks* 20.8, pp. 2481–2501.

Knudsen, L. and Preneel, B. (1999). "Error correcting codes and collision-resistant hashing". In: *Proceedings of the 1999 IEEE Information Theory and Communications Workshop (Cat. No. 99EX253)*, pp. 55–57. DOI: 10.1109/ITCOM.1999.781407.

Kolias, C., Kambourakis, G., Stavrou, A., and Voas, J. (2017). "DDoS in the IoT: Mirai and Other Botnets". In: *Computer* 50.7, pp. 80–84. DOI: 10.1109/MC.2017.201.

Kozlov, D., Veijalainen, J., and Ali, Y. (2012). "Security and privacy threats in IoT architectures." In: *BODYNETS*, pp. 256–262.

KrebsOnSecurity (2016a). *IoT Device Maker Vows Product Recall, Legal Action Against Western Accusers*. Report. URL: https://krebsonsecurity.com/2016/10/iot-device-maker-vows-product-recall-legal-action-against-western-accusers/. (accessed: 22.5.2022).

– (2016b). *Who Makes the IoT Things Under Attack?* Blogpost. URL: https://krebsonsecurity.com/2016/10/who-makes-the-iot-things-under-attack/. (accessed: 22.5.2022).

– (2017). *Who is Anna-Senpai, the Mirai Worm Author*. Blogpost. URL: https://krebsonsecurity.com/2017/01/who-is-anna-senpai-the-mirai-worm-author/. (accessed: 22.5.2022).

Kumar, N. M. and Mallick, P. K. (2018). "Blockchain technology for security issues and challenges in IoT". In: *Procedia Computer Science* 132. International Conference on Computational Intelligence and Data Science, pp. 1815–1823. ISSN: 1877-0509. DOI: https://doi.org/10.1016/j.procs.2018.05.140. URL: https://www.sciencedirect.com/science/article/pii/S187705091830872X.

Lakshmi, K. P., Archana, K., Prasanthi, Y., and Padma, V. (2019). "A Study on Internet of Things with Blockchain Technology". In: *2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI)*, pp. 372–376.

Li, Y., Li, D., Cui, W., and Zhang, R. (2011). "Research based on OSI model". In: *2011 IEEE 3rd International Conference on Communication Software and Networks*, pp. 554–557. DOI: 10.1109/ICCSN.2011.6014631.

Liu, Y., Wang, K., Qian, K., Du, M., and Guo, S. (2020). "Tornado: Enabling Blockchain in Heterogeneous Internet of Things Through a Space-Structured Approach". In: *IEEE Internet of Things Journal* 7.2, pp. 1273–1286.

Lueth, K. L. (2020). *State of the IoT 2020: 12 billion IoT connections*. Blogpost. URL: https://iot-analytics.com/state-of-the-iot-2020-12-billion-iot-connections-surpassing-non-iot-for-the-first-time. (accessed: 22.5.2022).

Madakam, S., Lake, V., Lake, V., Lake, V., et al. (2015). "Internet of Things (IoT): A literature review". In: *Journal of Computer and Communications* 3.05, p. 164.

Marzano, A., Alexander, D., Fonseca, O., Fazzion, E., Hoepers, C., Steding-Jessen, K., Chaves, M. H. P. C., Cunha, Í., Guedes, D., and Meira, W. (2018). "The Evolution of Bashlite and Mirai IoT Botnets". In: *2018 IEEE Symposium on Computers and Communications (ISCC)*, pp. 813–818. DOI: 10.1109/ISCC.2018.8538636.

Mashal, I., Alsaryrah, O., Chung, T. y., Yang, C.-Z., Kuo, W.-H., and Agrawal, D. (2015). "Choices for Interaction with Things on Internet and Underlying Issues". In: *Ad Hoc Networks* 28, pp. 68–90. DOI: 10.1016/j.adhoc.2014.12.006.

Mendes, L. D. P., Aloi, J., and Pimenta, T. C. (2019). "Analysis of IoT Botnet Architectures and Recent Defense Proposals". In: *2019 31st International Conference on Microelectronics (ICM)*, pp. 186–189. DOI: 10.1109/ICM48031.2019.9021715.

Minoli, D. and Occhiogrosso, B. (2018). "Blockchain mechanisms for IoT security". In: *Internet of Things* 1-2, pp. 1–13. DOI: 10.1016/j.iot.2018.05.002.

Mirzaie, S., Elyato, A. K., and Sarram, M. A. (2010). "Preventing of SYN Flood Attack with Iptables Firewall". In: *2010 Second International Conference on Communication Software and Networks*, pp. 532–535. DOI: 10.1109/ICCSN.2010.74.

Moskovitch, R., Feher, C., Messerman, A., Kirschnick, N., Mustafic, T., Camtepe, A., Lohlein, B., Heister, U., Moller, S., Rokach, L., and Elovici, Y. (2009). "Identity theft, computers and behavioral biometrics". In: *2009 IEEE International Conference on Intelligence and Security Informatics*, pp. 155–160. DOI: 10.1109/ISI.2009.5137288.

Ouaddah, A., Abou El Kalam, A., and Ouahman, A. A. (2017). "Harnessing the power of blockchain technology to solve IoT security & privacy issues." In: *Second International Conference on Internet of Things, Data and Cloud Computing*, pp. 1–7. DOI: 10.1145/3018896.3018901.

OWASP (2005). *Types of XSS*. Web article. URL: https://owasp.org/www-community/Types_of_Cross-Site_Scripting#DOM_Based_XSS_.28AKA_Type-0.29. (accessed: 22.5.2022).

– (2013). *What changed from 2013 to 2017*. Web article. URL: https://owasp.org/www-project-top-ten/2017/Release_Notes. (accessed: 22.5.2022).

– (2017). *OWASP Top 10 Application Security Risks - 2017*. Web article. URL: https://owasp.org/www-project-top-ten/2017/Top_10. (accessed: 22.5.2022).

Prada-Delgado, M., Baturone, I., Dittmann, G., Jelitto, J., and Kind, A. (2019). "PUF-derived IoT identities in a zero-knowledge protocol for blockchain". In: *Internet of Things* 9:100057, pp. 1–16. DOI: 10.1016/j.iot.2019.100057.

Pranathi, K., Kranthi, S., Srisaila, A., and Madhavilatha, P. (2018). "Attacks on Web Application Caused by Cross Site Scripting". In: *2018 Second International Conference on Electronics, Communication and Aerospace Technology (ICECA)*, pp. 1754–1759. DOI: 10.1109/ICECA.2018.8474765.

Raguvaran, S. (2014). "Spoofing attack: Preventing in wireless networks". In: *2014 International Conference on Communication and Signal Processing*, pp. 117–121. DOI: 10.1109/ICCSP.2014.6949811.

Razzaq, M. A., Gill, S. H., Qureshi, M. A., and Ullah, S. (2017). "Security issues in the Internet of Things (IoT): A comprehensive study". In: *International Journal of Advanced Computer Science and Applications* 8.6, p. 383.

Rehman, R. U. and Regina, N. (2003). *Intrusion Detection with SNORT (Bruce Perens' Open Source Series): Advanced IDS Techniques Using Snort, Apache, MySQL, PHP, and ACID*. Pearson Education. ISBN: 0131407333.

Rescorla, E. (2018). *The Transport Layer Security (TLS) Protocol Version 1.3*. RFC 8446. DOI: 10.17487/RFC8446. URL: https://rfc-editor.org/rfc/rfc8446.txt.

Salman, T., Zolanvari, M., Erbad, A., Jain, R., and Samaka, M. (2019). "Security Services Using Blockchains: A State of the Art Survey". In: *IEEE Communications Surveys Tutorials* 21.1, pp. 858–880.

Schütte, M., Scheffler, T., and Schnor, B. (2012). "Development of a Snort IPv6 Plugin-Detection of Attacks on the Neighbor Discovery Protocol." In: *SECRYPT*, pp. 399–402. DOI: 10.5220/0004073303990402.

Sharma, P. K., Singh, S., Jeong, Y., and Park, J. H. (2017). "DistBlockNet: A Distributed Blockchains-Based Secure SDN Architecture for IoT Networks". In: *IEEE Communications Magazine* 55.9, pp. 78–85.

Sinanović, H. and Mrdovic, S. (2017). "Analysis of Mirai malicious software". In: *2017 25th International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*, pp. 1–5. DOI: 10.23919/SOFTCOM.2017.8115504.

Singh, M., Singh, A., and Kim, S. (2018). "Blockchain: A game changer for securing IoT data". In: *2018 IEEE 4th World Forum on Internet of Things (WF-IoT)*, pp. 51–55. DOI: 10.1109/WF-IoT.2018.8355182.

Snort.org (2021). Organization Website. URL: https://www.snort.org/.

Standards, N. I. of and Technology (2013). *Security and Privacy Controls for Federal Information Systems and Organizations*. Tech. rep. URL: https://nvlpubs.nist.gov/nistpubs/specialpublications/nist.sp.800-53r4.pdf.

Tencent (2016). *Car Hacking Research: Remote Attack Tesla Motors*. Web News. URL: https://keenlab.tencent.com/en/2016/09/19/Keen-Security-Lab-of-Tencent-Car-Hacking-Research-Remote-Attack-to-Tesla-Cars/. (accessed: 22.5.2022).

Thewindowsclub (2020). *Denial of Service (DoS) Attack: What it is and how to prevent it*. Web article. URL: https://www.thewindowsclub.com/dos-denial-of-service-attack. (accessed: 22.5.2022).

Thomasson, J.-P. and Baldi, L. (1997). "Smartcards: portable security". In: *1997 Proceedings Second Annual IEEE International Conference on Innovative Systems in Silicon*, pp. 259–265. DOI: 10.1109/ICISS.1997.630268.

Tuptuk, N. and Hailes, S. (2018). "Security of smart manufacturing systems". In: *Journal of Manufacturing Systems* 47, pp. 93–106. DOI: 10.1016/j.jmsy.2018.04.007.

Vasilyevna, N. B. (2008). "An RBAC Design with Discretionary and Mandatory Features". In: *2008 International Symposium on Ubiquitous Multimedia Computing*, pp. 260–263. DOI: 10.1109/UMC.2008.60.

Wang, R., He, J., Liu, C., Li, Q., Tsai, W., and Deng, E. (2018). "A Privacy-Aware PKI System Based on Permissioned Blockchains". In: *2018 IEEE 9th International Conference on Software Engineering and Service Science (ICSESS)*, pp. 928–931.

Williams, L. (2022). *OSI Model Layers and Protocols in Computer Network*. Web article. URL: https://www.guru99.com/layers-of-osi-model.html. (accessed: 22.5.2022).

Zeidanloo, H. R., Tabatabaei, F., Amoli, P. V., and Tajpour, A. (2010). "All About Malwares (Malicious Codes)." In: *Security and Management*, pp. 342–348.

Zhang, H. and Lang, W. (2019). "Research on the Blockchain Technology in the Security of Internet of things". In: *2019 IEEE 4th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*. Vol. 1, pp. 764–768.

Zhou, G.-z. and Li, J.-y. (2012). "Research on Snort Intrusion Detection System and Key Match Algorithm". In: *Electrical, Information Engineering and Mechatronics 2011*. Ed. by X. Wang, F. Wang, and S. Zhong. Springer London, pp. 623–629. ISBN: 978-1-4471-2467-2.

Zhou, S. and Xiao, S. (2018). "3D face recognition: a survey". In: *Human-centric Computing and Information Sciences* 8, pp. 1–27. DOI: 10.1186/s13673-018-0157-2.

Zhou, Z., Chen, Z., Zhou, T., and Guan, X. (2010). "The study on network intrusion detection system of Snort". In: *2010 International Conference on Networking and Digital Society*. Vol. 2, pp. 194–196. DOI: 10.1109/ICNDS.2010.5479341.

Zhu, X. and Badr, Y. (2018). "Fog Computing Security Architecture for the Internet of Things Using Blockchain-Based Social Networks". In: *the 2018 IEEE Symposium on Blockchain*, pp. 1361–1366.