

**UNIVERSITY OF HERTFORDSHIRE  
School of Physics, Engineering and Computer Scienc**e

**FINAL PROJECT REPORT**

**7COM1039 and Advanced Computer Science Masters Project**

**July 22, 2022.**

**AN INVESTIGATION ON HOW DISTRIBUTED DENIAL OF SERVICE ATTACKS SYSTEMS UTILIZING INTERNET OF THINGS IN THE CLOUD.**

**Name: Ozurumba Chinedu Francis**

**Student ID: 19019481**

**Supervisor: Mr Ed Edris.**

## **PROJECT DECLARATION**

This report is submitted in partial fulfilment of the requirement for the degree of Master of Science in Advanced Computer Science at the University of Hertfordshire (UH).

It is my own work except were indicated in the report.

I did not use human participants in my MSc Project.

I hereby give permission for the report to be made available on the university website provided the source is acknowledged.

Ozurumba Chinedu Francis

Student ID: 19019481

**ABSTRACT**

In the last decade, the term "internet of things" (IoT) has grown in popularity, which can be defined as various connected devices via the internet. IoT has rapidly expanded to encompass all aspects of our lives. Smart homes, smart cities, and various wearable devices are examples. IoT devices function to achieve their intended goals, which are to develop a person's life with minimal human intervention. At the same time, IoT devices have numerous flaws that attackers can exploit to compromise their security. The most common attacks on IoT security are denial of service (DoS) and distributed denial of service (DDoS). The main goal of these attacks is to use malicious malware to bring down victim systems and make them inaccessible to legitimate users.

The primary goal of this paper is to investigate how DDoS attacks affect system (web-server) performance via IoT in the Cloud environment.

**ACKNOWLEDGEMENTS**

I would like to thank the Almighty God, for providing sufficient strength to begin and finish this research work, even when the odds were against me. I wish to thank my project supervisor Mr Ed Edris for his wise and homely advise every week, and for pushing me to work harder in other to complete this work. His support and belief in me were immense in adding value to this research work.

To my parents, my siblings and my friends: you put up with me being distracted and missing many events. I am forever grateful for your patience and understanding. I hope to have time now to reconnect with each of you.

TABLE OF CONTENTS

## PROJECT DECLARATION --------------------------------------------------------------------------- 2

ABSTRACT----------------------------------------------------------------------------------------------- 3

ACKNOWLEDGEMENTS----------------------------------------------------------------------------- 4

CHAPTER ONE------------------------------------------------------------------------------------------ 7

INTRODUCTION---------------------------------------------------------------------------------------- 7

1.1 Background Of The Study -------------------------------------------------------------------- 7

1.2 Research Question------------------------------------------------------------------------------ 9

1.3 Research Objective------------------------------------------------------------------------------9

1.4 Research Aim------------------------------------------------------------------------------------ 9

CHAPTER TWO ----------------------------------------------------------------------------------------10

LITERATURE REVIEW-------------------------------------------------------------------------------10

2.1 Introduction ------------------------------------------------------------------------------------10

2.2 Overview of IoT -------------------------------------------------------------------------------11

2.2.1 Applications of IoT----------------------------------------------------------------------------11

2.2.2 The IoT Architecture---------------------------------------------------------------------------13

2.2.3 Challenges in Internet of Things (IoT)------------------------------------------------------14

2.2.3.1 Security Challenges------------------------------------------------------------------14

2.2.3.2 Design Challenges--------------------------------------------------------------------15

2.2.3.3 Deployment Challenges-------------------------------------------------------------15

2.3 Overview of DDoS Attack and Threat------------------------------------------------------15

2.3.1 What is DDoS Attack--------------------------------------------------------------------------15

2.3.2 How does a DDoS attack work---------------------------------------------------------------16

2.3.3 How are DoS/DDoS attack tools categorized----------------------------------------------17

2.3.4 DoS/DDoS Attack tools-----------------------------------------------------------------------17

2.4 IoT DoS/DDoS attack classification based on layered structure ------------------------19

2.4.1 Perception Layer-------------------------------------------------------------------------------19

2.4.1.1 Security Vulnerabilities at Perception Layer-----------------------------------------------19

2.4.1.2 Famous types of attacks on this layer:------------------------------------------------------19

2.4.1.3 Security Solution at Perception Layer------------------------------------------------------19

2.4.2 Network Layer---------------------------------------------------------------------------------20

2.4.2.1 Security Vulnerabilities at Network Layer-------------------------------------------------20

2.4.2.2 Famous types of attacks on this layer-------------------------------------------------------20

2.4.2.3 Security Solution at Network Layer--------------------------------------------------------20

2.4.3 Application Layer-----------------------------------------------------------------------------21

2.4.3.1 Security Vulnerabilities at Application Layer---------------------------------------------21

2.4.3.2 Famous types of attack on this layer--------------------------------------------------------21

2.4.3.3 Security Solutions at Application Layer----------------------------------------------------21

2.5 Overview of Amazon Web Services (AWS)-----------------------------------------------21

2.5.1 AWS IoT Core----------------------------------------------------------------------------------22

2.5.2 How your devices and applications connect to AWS IoT--------------------------------22

2.5.3 How AWS IoT Works-------------------------------------------------------------------------23

2.6 What is Raspberry Pi--------------------------------------------------------------------------25

CHAPTER THREE--------------------------------------------------------------------------------------27

METHODOLOGY---------------------------------------------------------------------------------------27

3.1 Introduction-------------------------------------------------------------------------------------27

3.3 Raspberry Pi and Ultrasonic Ranging Sensor connectivity-------------------------------27

3.3.1 What is HC-SR04 Ultrasonic Sensor--------------------------------------------------------27

3.3.2 How HC-SR04 Ultrasonic Sensor works---------------------------------------------------28

3.3.3 Circuit Design----------------------------------------------------------------------------------29

3.4 Deploying Raspberry Pi to AWS cloud-----------------------------------------------------29

3.4 Attack Generation Tools and Technology used--------------------------------------------32

3.4.1 Nmap---------------------------------------------------------------------------------------------32

3.4.2 Hping3-------------------------------------------------------------------------------------------32

3.4.3 Slowloris----------------------------------------------------------------------------------------32

3.4.4 UFONet-----------------------------------------------------------------------------------------33

3.4.5 Wireshark---------------------------------------------------------------------------------------33

## 3.5 Legal, Social, and Professional Considerations-----------------------------------------------33

3.5.1 Legal Obligations and Liability--------------------------------------------------------------33

3.5.2 Professional Issues-----------------------------------------------------------------------------34

3.5.3 Social Issues------------------------------------------------------------------------------------34

CHAPTER FOUR---------------------------------------------------------------------------------------35

EXPERIMENT, RESULTS and ANALYSIS-------------------------------------------------------35

4.1 Introduction-------------------------------------------------------------------------------------35

4.2 Experiment--------------------------------------------------------------------------------------35

4.2.1 Phase 1: Port Scanning------------------------------------------------------------------------36

4.2.2 Phase 2: DoS Attack Launch-----------------------------------------------------------------38

4.2.2.1 DoS attack using Slowloris-------------------------------------------------------------------39

4.2.2.2 DoS attack using UFONet--------------------------------------------------------------------39

4.2.2.3 DoS attack using Hping3---------------------------------------------------------------------40

4.3 Result--------------------------------------------------------------------------------------------40

4.4 Analysis-----------------------------------------------------------------------------------------42

CHAPTER FIVE-----------------------------------------------------------------------------------------43

DISCUSSION and CONCLUSION-------------------------------------------------------------------43

5.1 Discussion---------------------------------------------------------------------------------------43

5.2 Conclusion--------------------------------------------------------------------------------------43

5.3 DDoS Protection Techniques in the Cloud-------------------------------------------------44

REFERENCES-------------------------------------------------------------------------------------------46

**CHAPTER ONE**

**INTRODUCTION**

**1.1 Background Of The Study**

The "Internet of Things" (IoT) is becoming a popular topic of discussion both in and out of the workplace. It's a concept that has the potential to influence not only how we live, but also how we work. But what exactly is the “ Internet of Things? ”.

The term "Internet of Things" was coined by (Ashton, 1999) to describe a technological revolution that represents the future of computing and communications. It represents a network of devices that can connect and communicate with billions of things at the same time. This type of network does not require expensive components and can be built with inexpensive sensors and interconnected objects that collect data from the environment and enable improvements in how we live. The Internet of Things is the next step on the path to the fourth industrial revolution, in which the Internet is expanded to include more of the physical world, bringing more intelligence to everyday objects and, as a result, more control over the physical world. The *I* in IoT refers to the universal Inter-connectivity between all perceivable objects, including the traditional computing objects such as computers and smart phones, and the new generation of smart objects. The *T* in IoT enabled by *sensors*, *actuators*, and all embedded computers into everyday objects, from toys and wearable to home appliances, manufacturing equipment, vehicles and up to buildings, power grids and the entire urban city.

According to a study conducted by Fujino, Ogawa, and Minowa (2016), the global population uses approximately 10 billion IoT devices in their daily lives. Furthermore, according to dataprot.net, it is estimated that by 2025, there will be 152,200 IoT devices connecting to the internet per minute. The devices are based on a centralized Client/Server model and have three layers of architecture with some additional helper layers: *Perception* (collects information from the environment), *Network* (controls information processing), and *Application layer* (contains business logic). Certain elements of this architecture are vulnerable to specific attacks that endanger the security of IoT devices. One example is a Distributed Denial of Service (*DDoS*) attack.

Because of the rapid expansion of smart appliances that are all connected to a network, the increased use of IoT devices poses multiple security risks on these devices. Home automation appliances, thermostats, printers, and even personal refrigerators are among the IoT devices that can be controlled with other systems such as Google Assistant, Alexa, and Siri (Celosia, 2020). Thus, hijacking these devices can be achieved easily, such as sending a spam email conscripted into a botnet and leaking personal information, pinging and scanning the network to discover open and vulnerable ports. However, the use of IoT devices has skyrocketed and data transmission has been uncountable. Because of the critical data connection and billions of networks all over the world, managing and tracking data security has become difficult (Kimani, Oduol, and Langat, 2019). Indeed, the proliferation of connected and unsecured devices in today's world implies an increase in attack vectors and opportunities for hackers to target these devices, access sensitive data, and control our devices, compromising personal privacy.

According to De Donno et al. (2018), This insecurity trend has pushed Distributed Denial of Service (DDoS) attacks to the forefront, making them more powerful and complex than ever (though easier to achieve, as they are even *offered as a service*), and thus much more difficult to identify and characterize. As a result, DDoS popularity has grown significantly in recent years, precisely when the Internet of Things revolution flooded the Internet with poorly protected devices ready to engage in criminal activities. The issue of DDoS attacks was brought to the fore in 2016 when IoT was subjected to the most severe DDoS attack ever. According to Vignau et al. (2021), DDoS attacks and Malware affected thousands of connected devices, with an offensive capability of approximately 1.2 terabits per second. According to Somani et al. (2017), more than 20% of enterprises worldwide experienced at least one DDoS attack on their infrastructure.

Cloud computing is currently regarded as the most recent computing paradigm, providing numerous flexible and consistent services via virtualization technology, which is used in the next generation of data centers. Through cloud computing infrastructure, private companies, individuals, and government departments are attempting to increase service availability. Cloud computing, with its capacity, resilience, and cost minimization, enables the sharing of resources in a pervasive and transparent manner, as well as the ability to perform procedures that meet various needs. Furthermore, cloud computing provides users with on-demand services and the ability to access shared infrastructure. The National Institute of Standards and Technology (NIST) identifies five fundamental cloud computing specifications: on-demand self-service, broad network, access Resource pooling, measured service, and rapid elasticity.

According to Wang et al. (2017), cloud computing has grown in popularity due to activities and services such as pay per usage, system flexibility, reduced software and system maintenance costs, demand access, and virtualization. These features have transformed data and information management in every institution, including businesses.

Software as a Service (SaaS), Platform as a Service (PaaS) and Infrastructure as a Service (IaaS) is cloud computing software models that use virtualization technology to provide infrastructure by sharing hardware with multiple users. Visualisation, according to Wanjun (2016), is an important tool in cloud computing. Cloud computing virtualization has recently been successfully integrated with stages such as Networks, CPU, Database, and Storage, greatly improving the system available to make the software a more superior flexible system.

DDoS and malware attacks, on the other hand, have posed a significant threat to the performance of cloud computing systems (El-Sofany and Abou El-Seoud, 2019). An attacker using DDoS and Malware attacks can severely degrade or completely disrupt the victim's network connectivity. A DDoS attack's primary goal is to prevent the user from accessing cloud computing services. By targeting web services, CPU, overall storage, and other network resources, the attacker hopes to prevent the victim from accessing various resources. Furthermore, both malware and DDoS attacks degrade cloud environment service performance by destroying critical virtual servers. Malware on cloud services seeks out a bug or existing weakness in the software in order to disrupt cloud functionality. In some instances, the attacks deplete all the bandwidth and resources of the victim’s system. Ali (2018), highlights that DDoS attacks in the past years were initially remotely controlled, well-executed, and highly

spread in nodes identified as Zombies. The attacks are usually launched using the assistance of Zombies (secondary victims).

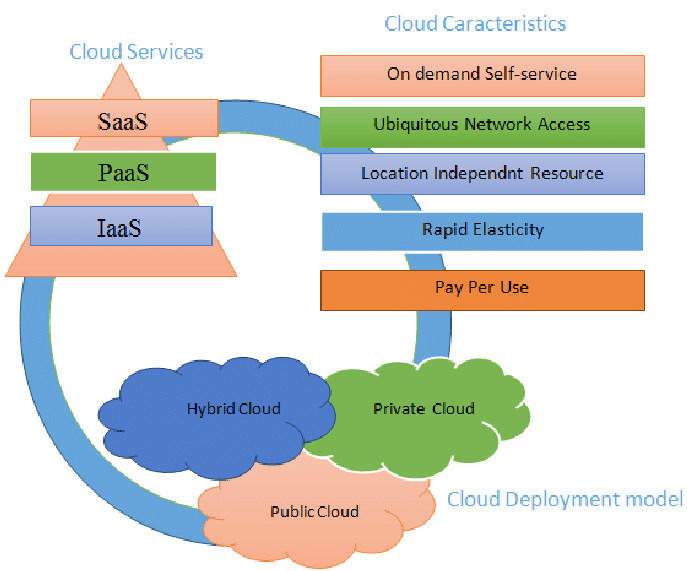


Figure 1.0: Cloud Deployment Model

**1.2 Research Question**

This study will seek to answer the question, "How does DDoS Attack threaten the performance of cloud computing services via Internet of Things?".

**1.3 Research Objective**

IoT makes use of the internet and real-time to provide the end-user with a resourceful and effective environment. They also rely heavily on cloud computing systems for data storage. As a result, any IoT security threats directed at cloud computing systems have a significant impact on confidentiality and performance, and can result in massive economic losses.

This research will specifically look into how DDoS attacks affect the performance of the system via IoT in the Cloud environment. This will be accomplished by launching a DDoS attack on the hardware platform (Raspberry pi 4) which is deployed to the cloud, and determining the level of damage caused by the DDoS attacks. Another goal of the research is to look into the design and inspection features of DDoS attacks, as well as effective approaches to mitigating the security threats.

**1.4 Research Aim**

Information communication and technology are critical and necessary components of our daily lives and businesses. Technology has resulted in the development of more efficient methods for carrying out various processes, organising, and storing information. IoT and cloud computing are two well-known and significant technologies. According to Singh and Sharma (2019), cloud computing and IoT work together as a collaboration and are used to store IoT data. Cloud computing has attracted a large number of users from academic institutions, entities, and cooperating organisations due to its capabilities and cost-effectiveness. Cloud computing provides high availability, which is critical in computer science. Despite the underlying benefits, however, these systems face security threats relating to user information credibility and service. Because of the cloud's resource multi-tenancy and sharing capabilities, these security threats can have an impact on the environment.

Distributed Denial of Service (DDoS) are significant threats to the performance of IoT and Cloud computing. This study's goal will be to identify, analyze and investigate the significant DDoS attacks and the security threats that affects the availability and use of IoT and Cloud computing services. This project will also look into how these attacks affect various components of IoT and cloud computing.

Furthermore, this project will use Amazon Web Service (AWS) as a Cloud Service (CSP), Raspberry pi 4 as the IoT Hardware platform.

**CHAPTER TWO**

**LITERATURE REVIEW**

**2.1 Introduction**

This chapter provides background information on overview of IoT, DoS/DDoS attack types based on the layered structure after an extensive literature review was conducted, with the results presented in the following section. In addition, some relevant solutions are presented here to provide a broad overview of the existing approaches on handling IoT security challenges. Farooq et al. examine the layered architecture of IoT and the main security goals of data confidentiality, data integrity, and data availability. They also discuss security issues and potential solutions at each layer. Ning et al propose an IPM-based systematic security architecture that consists of three security aspects: information, physical, and management. This security model incorporates a social layer, as well as intelligence and compatibility, for security consideration.

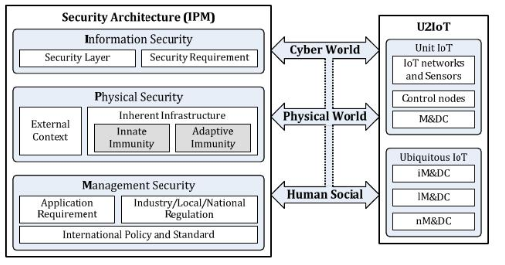


Figure 2.0: An IPM-based systematic security architecture

(Eskandari, M. et al. 2020) developed Passban, an Intelligent Intrusion Detection System (IDS). Passban is designed in such a way that it protects IoT devices that are directly connected to it. The benefit of their work is that it can be used on low-cost IoT gateways using the edge computing paradigm to detect cyber threats as close to the corresponding data source as possible. Passban has been shown to detect various types of malicious intrusion with high accuracy, including SSH brute force, port scanning, and SYN flood attacks. M. Eskandari et al., 2020

In their paper, authors Perakovi et al analyse data collected from a domestic company regarding DDoS attack trends. According to their findings, the Simple Service Discovery Protocol (SSDP) is the most commonly used when conducting DDoS attacks. They also examined the number of IoT-related attacks and concluded that an increase in IoT devices also means an increase in DDoS attacks.

Oh et al. conducted a simple study on dealing with DDoS attacks, in which they presented four major approaches that can be considered in order to defend against these attacks. Defense against DDoS attacks can be accomplished using a rate limit framework, defence by offence, active filtering, or IP traceback.

**2.2 Overview of IoT**

What is Internet of Things?

The term “Internet of Things” has been defined by various authors in many different ways. Let's consider two of the most common definitions.

The Internet of Things, according to Vermesan et al., is simply an interaction between the physical and digital worlds. A plethora of sensors and actuators allows the digital and physical worlds to interact. According to Pena-L'opez et al., the Internet of Things is a paradigm in which computing and networking capabilities are embedded in any conceivable object. These capabilities are used to query the object's state and, if possible, change its state.

IoT devices are outfitted with embedded sensors, actuators, processors, and transceivers to enable this intelligence and interconnection. IoT is not a single technology; it is a collection of technologies that work in tandem.

Sensors and actuators are devices that aid in interacting with the physical environment. The data collected by the sensors must be intelligently stored and processed in order to yield useful inferences. It is important to note that we define "sensor" broadly; a mobile phone or even a microwave oven can count as a sensor if it provides inputs about its current state (internal state + environment). An actuator is a device that causes a change in the environment, such as the temperature controller of an air conditioner.

Data storage and processing can take place at the network's edge or on a remote server. If data preprocessing is possible, it is typically performed at the sensor or another nearby device. Typically, the processed data is then sent to a remote server. The storage and processing capabilities of an IoT object are also constrained by the resources available, which are frequently limited due to size, energy, power, and computational capability constraints.

**2.2.1 Applications of IoT**

As illustrated in Fig. 2.1, recent advances in sensor development, lightweight communication protocols, open source server programs, web development tools, and IoT dashboards are being applied to a variety of societal problems.

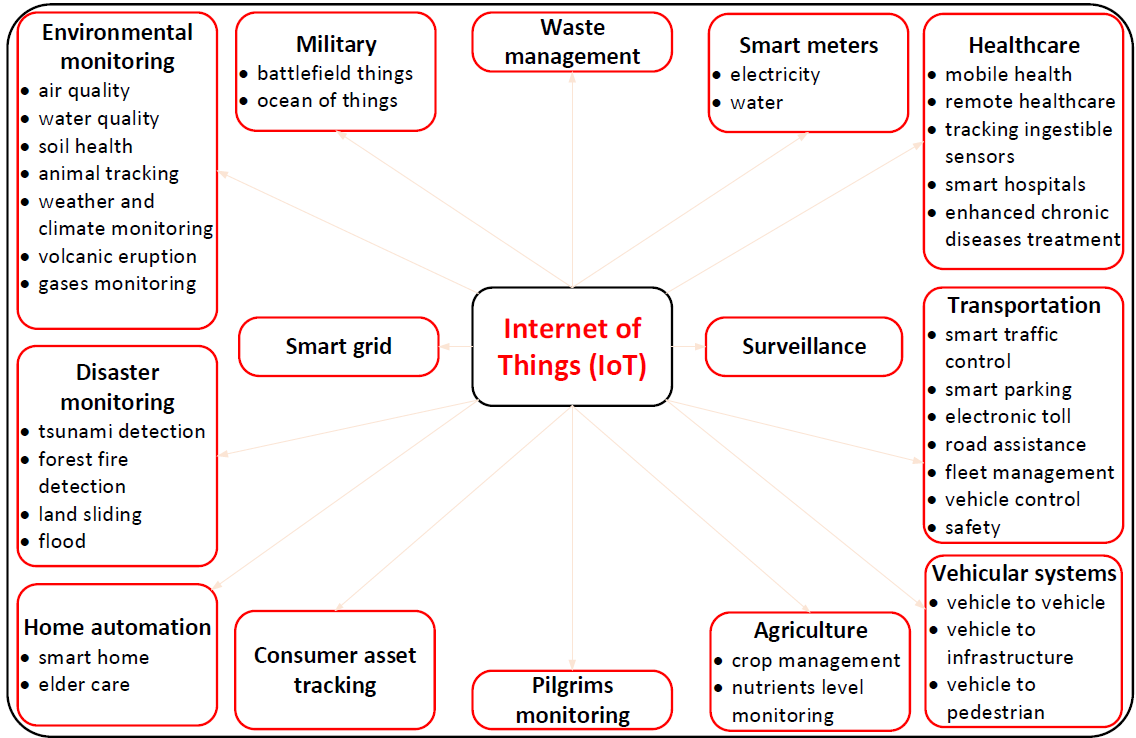


Figure 2.1: Applications of IoT.

1. Military

The Internet of Things has critical applications in the military. Some of the wireless network technologies used for armed personnel health monitoring and tracking include ZigBee and GSM networks. However, these technologies have drawbacks such as limited bandwidth, high operational costs, complex infrastructure, and so on. These disadvantages can be overcome by incorporating an IoT architecture into the health monitoring and tracking system for armed personnel. The deployment of IoT on the battlefield can improve military authorities' situational awareness, risk assessment, and response time. IoT-based systems can help the military identify enemies, monitor the physical and mental health of armed personnel, and synchronize armed personnel with defense systems. Military uniforms and helmets with embedded sensor devices can be used to track armed personnel's locations and vital status. The command center can take immediate action to save the lives of armed personnel. Sensors such as temperature sensors, pulse rate sensors, accelerometers, ECG sensors, mobility sensors, and oxygen level sensors can be used to create smart military uniforms and helmets.

1. Home Automation

The Internet of Things has changed the way electronic appliances are controlled in the home. A relay switch, a microcontroller, and a network device can be used to connect any appliance to the internet, including lights, air conditioners, media, security systems, refrigerators, and ovens. The user can control the appliances from a remote location by using a graphical user interface. A wide range of sensors are being developed to implement an IoT-based home automation system. Temperature sensors (DHT11/22, DS18B20, LM35), lux sensors (TSL2591, BH1750), water level sensors (HC-SR04, LM1830), air composition sensors (MiCS-5525, MQ-8, MQ135), and humidity sensors are among those included (HIH6100, Dig RH).

1. Consumer Asset Tracking

One of the characteristics of IoT is the ability to give physical objects a unique identity. This feature employs IoT in the development of a consumer asset tracking system. Most mobile phones now include a GPS tracking system that makes it simple to locate misplaced phones. Similarly, the IoT enables this facility for consumer assets by providing them with a unique identity and internet connectivity. E-commerce businesses can track the status of their products in transit to ensure proper delivery. The system's sensing components include an accelerometer, GPS, vibration sensors, and RFID. Nano network processors are used to connect the internet to cloud services.

1. Surveillance

Monitoring human activities and behaviour is critical in many ways. A surveillance technique is required for tracking the activities of any labor force, observing workplace facilities, and observing areas where security is a concern. In other words, surveillance enables the user to manage and direct people while also preserving the event as evidence during an investigation. In, an IoT-based surveillance system is presented. The internet-connected surveillance device continuously captures and collects data. Data can be in audio or video format. The captured data is distributed remotely via the cloud server. The main benefits are that continuous monitoring can be done from a remote location; real-time visibility of a person entering and leaving a location is possible; and security alert messages will be delivered to a specific mobile device based on motion detection, face detection, and vehicle number plate to take immediate action if there is any theft process. It also manages and directs workers at workplaces in real time.

1. Smart Metering

Every home requires access to water, gas, and electricity. It is common practice to receive bills by mail at the end of each month. Smart metering based on IoT allows the service provider to monitor the usage of essential resources from a central station while also allowing the user to monitor day-to-day usage. The smart metre collects usage data and transmits it to a central server via the internet. Central monitoring stations, such as electricity distribution centres, gas distribution centres, and water distribution centres, process the data and transmit it to customers via the internet.

Customers can view their daily usage, and when they receive paperless bills via the internet, they can log into their electronic account and pay the bills electronically.

**2.2.2 The IoT Architecture**

The flow of information or data from sensors to large server clouds is referred to as IoT architecture. The sensors are attached to the "things" and collect data from their surroundings. Large cloud servers perceive, store and process incoming data to produce required outputs. Data is sent back to the "things" via the clouds, causing a chain reaction. When large amounts of data are involved in a network framework, a proper architecture is required to control the data flow, send and receive signals, perform computations, and handle storage.

When it comes to the IoT ecosystem, there is no universally agreed-upon architecture. Every production company has its own set of private guidelines based on its own needs and requirements. The generic IoT architecture is divided into three basic layers.

When designing a model architecture, we must usually keep in mind that we must accommodate a wide range of possibilities. Different industries have different requirements, and their architecture is built around these.

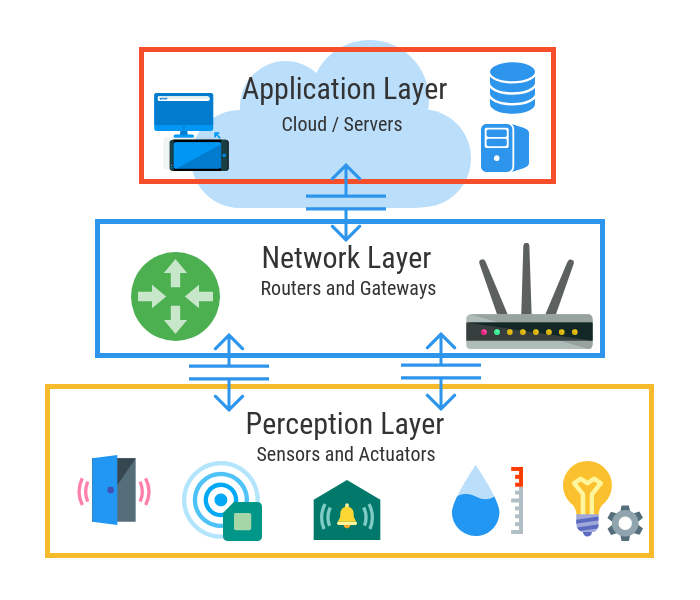


Figure 2.2: Three Layers of IoT Architecture

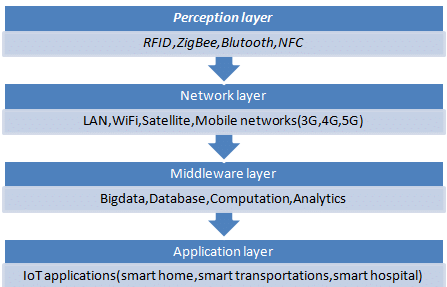


Figure 2.2.1: IoT Layers and Used Protocols

**2.2.3 Challenges in Internet of Things (IoT)**

**2.2.3.1 Security Challenges:**

1. Bandwidth and Power Consumption: IoT devices are designed to be small, with limited computing power and memory capacity. As a result, advanced cryptographic algorithms cannot be used in IoT systems due to the high computing and memory requirements. Meanwhile, IoT contains many connected sensors to perform the desired task while maintaining security concerns, which may consume a large amount of bandwidth. As a result, security mechanisms should be implemented on IoT systems with minimal overhead.
2. Insecure Network Services: All IoT systems rely heavily on network communications, so these networks must be secure. Otherwise, network services will be jeopardized due to buffer overflows, fuzzing, DDoS, and other types of attacks.
3. Lack of Encryption: Although encryption is an excellent method of preventing hackers from accessing data, it is also one of the most significant IoT security challenges. These drives have the same storage and processing capabilities as a traditional computer. As a result, there is an increase in attacks where hackers can easily manipulate the algorithms designed to protect.
4. Inadequate Authentication and Authorization Mechanism: The majority of IoT devices have weak and default passwords, insecure credentials, and no access control. This exposes the devices to harm, as it's connected to the internet. As a result, an attacker could use this to threaten privacy and data integrity. Because of weak credentials and login information, nearly all IoT devices are vulnerable to password hacking and brute force.
5. Insufficient Testing and Lack of Update: With the increasing use of IoT (internet of things) devices, manufacturers are more eager to produce and deliver their devices quickly without worrying about security. Despite the rapid expansion of the Internet of Things (IoT) platform, many manufacturers don't give marketing research much thought before swooping into the market. Websites that offer shorter periods of time rarely update their content consistently. These updates are being made due to an increase in device shortages. As a result, they create a new generation of devices and ask users to begin using them without adequate security measures.

Having outdated software can leave the IoT device vulnerable to hackers and other security breaches. IoT devices that send data to the cloud may also experience downtime. It is possible for some parts of the app to stop working during an upgrade, while others search for updated app components and submit the updated data. It is recommended that firmware versions A49 and B06 be updated as soon as possible because they contain bugs that may compromise the security of the router. Given the numerous "internet of things" security risks, proper automatic updates are critical.

The manufacturer of IoT devices are responsible for updating their devices to the most recent software as soon as a vulnerability or malware attack is discovered.

1. Home Invasion: The most distressing scenario of Internet of Things (IoT) security challenges is when IoT device users are subjected to home invasions. Smart devices are now permeating every aspect of our lives. Because many of our homes are now connected to the Internet, the concept of smart homes has emerged. The biggest issue with smart home systems is that if there is a security flaw, they can broadcast IP addresses to nearby hackers. The search engine Shodan can assist hackers in determining the location of the device's user. In terms of abuse, it is possible to conclude that this technology can reach criminal circles and be sold to the general public. As an introduction, the best way to avoid that IoT security flaw is to configure each device to connect via VPNs, consider the importance of passwords, and secure the login details.
2. Hijacking and Ransomware: Ransomware can infect smart-home devices, wearables, and audio-video devices that are linked to cloud services. Malicious malware that infiltrates a user's files, encrypts them, and prevents the user from accessing sensitive files. When a hacker infects an IoT device with ransomware, they can take control of the device and demand money in exchange for the victim's encrypted files to be unlocked.
3. Fortunately, it is still a rare reality. However, this is a significant challenge in the hacking community and among hackers themselves.

**2.2.3.2 Design Challenges:**

1. Security of the System: Systems must be designed and implemented to be robust and reliable, as well as secure through the use of cryptographic algorithms and security procedures. It entails various approaches to securing all embedded system components from prototype to deployment.
2. Battery Life is a Limitation: Packaging and integration issues with small-sized chips with low weight and power consumption. Computers are becoming smaller, but battery energy remains constant.
3. Increased Cost and Time to Market: Cost is a minor constraint for embedded systems. The need for better approaches when designing IoT devices to handle cost modeling or cost optimally with digital electronic components arises.

Designers must also solve the design-time issue and bring the embedded device to market on time.

**2.2.3.3 Deployment Challenges:**

1. Data Collection and Processing: Data is crucial in the development of IoT. What is more important in this case is the processing or usefulness of the stored data. Along with security and privacy, development teams must plan carefully for how data is collected, stored, and processed within an environment.
2. Connectivity: It is the most important consideration when connecting devices, applications, and cloud platforms. Devices that are connected and provide useful front and information are extremely valuable. When IoT sensors are required to monitor process data and supply information, poor connectivity becomes a challenge.
3. Lack of Skillset: All of the development challenges listed above can only be addressed if an appropriately skilled resource is working on the IoT application development. The right talent will always get you past the major challenges and will be a valuable asset in the development of IoT applications.
4. Cross Platform Capability: IoT applications must be created with future technological changes in mind. Its development necessitates a delicate balance of hardware and software functions. It is difficult for IoT application developers to ensure that the device and IoT platform drivers perform optimally despite high device rates and fixes.

**2.3 Overview of DDoS Attack and Threat**

**2.3.1 What is DDoS Attack?**

A distributed denial-of-service (DDoS) attack is a malicious attempt to disrupt normal traffic to a specific server, service, or network by flooding the target or its surrounding infrastructure with Internet traffic or packets or it can be defined as a brute-force attempt to slow down a server.

DDoS attacks are effective because they use multiple compromised computer systems as attack traffic sources. Computers and other networked resources, such as IoT devices, can be exploited.

A DDoS attack is analogous to an unexpected traffic jam clogging the highway, preventing regular traffic from reaching its destination.

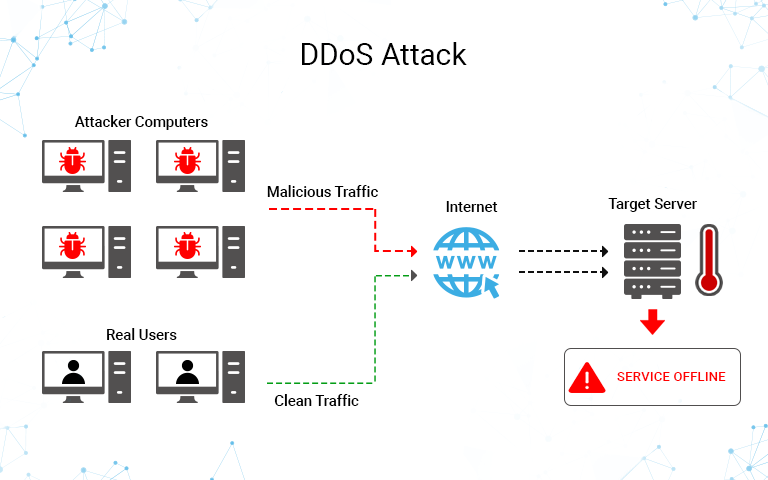


Figure 2.3: A typical setup of a DDoS attack.

**2.3.2 How does a DDoS attack work?**

DDoS attacks are carried out using networks of machines connected to the Internet. These networks are made up of computers and other devices (such as IoT devices) that have been infected with malware, allowing an attacker to control them remotely. Individual devices are known as bots (or zombies), and a network of bots is known as a botnet.

After establishing a botnet, the attacker can direct an attack by sending remote commands to each bot.

When the botnet targets a victim's server or network, each bot sends requests to the target's IP address, potentially overloading the server or network and causing a denial of service to normal traffic. Because each bot is a legitimate Internet device, distinguishing between attack and normal traffic can be difficult.

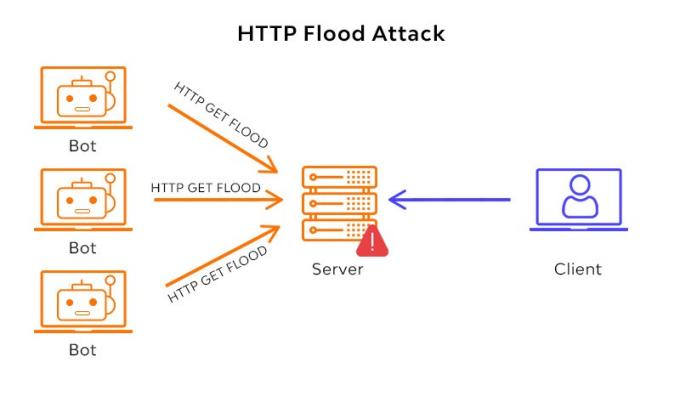


Figure 2.3.1: DDoS HTTP Flood attack.

What is the difference between a DoS attack and a DDoS attack?

The number of connections used in the attack is the distinctive feature between DDoS and DoS. Some DoS attacks, such as Slowloris, derive their power from the simplicity and minimal requirements required to be effective.

DoS attacks use a single connection, whereas DDoS attacks use multiple sources of attack traffic, often in the form of a botnet. Many of the attacks are fundamentally similar and can be attempted using a variety of malicious traffic sources. Learn how Cloudflare's DDoS protection prevents distributed denial-of-service attacks.

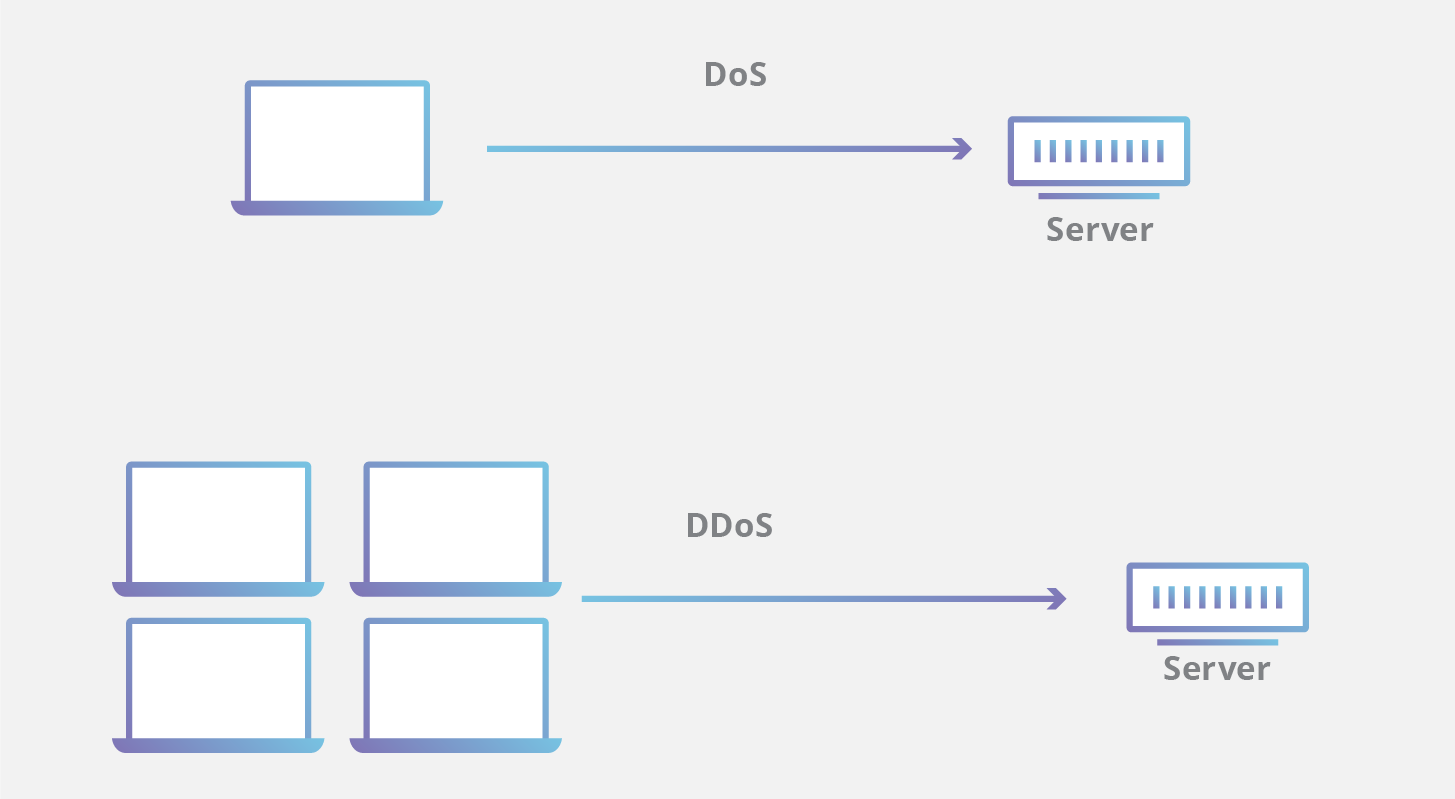


Figure 2.3.2: Difference between DoS and DDoS attack.

**2.3.3 How are DoS/DDoS attack tools categorized?**

There are quite a lot of tools out there that has be explicitly designed and can be used to launch a successful DoS/DDoS attack. The former category frequently includes "stressors"—tools designed to assist security researchers and network engineers in performing stress tests on their networks but can also be used to launch genuine attacks.

Some are specialized and only target a single layer of the OSI model, whereas others are designed to support multiple attack vectors. Attack tools are classified as follows:

Low and Slow attack tools

As the name implies, these attack tools use a small amount of data and operate slowly. These tools are designed to send small amounts of data across multiple connections to keep ports on a specific server open for as long as possible. They continue to consume the server's resources until the server is unable to maintain additional connections. Low and slow attacks, on the other hand, can be effective even when not using a distributed system like a botnet and are commonly used by a single machine.

Application Layer (L7) attack tools

These tools are aimed at layer 7 of the OSI model, which is where Internet-based requests like HTTP occur. A malicious actor can launch attack traffic that is difficult to distinguish from normal requests made by actual visitors by using an HTTP flood attack to overwhelm a target with HTTP GET and POST requests.

Protocol and Transport Layer (L3/L4) attack tools

Moving down the protocol stack, these tools use protocols such as UDP to send large amounts of traffic to a specific server, such as during a UDP flood. While these attacks are often ineffective on their own, they are most commonly found in the form of DDoS attacks, where the benefit of additional attacking machines increases the effect.

**2.3.4 DoS/DDoS Attack tools**

1. LOIC

This DDoS attack tool is short for Low Orbit Ion Cannon. It's really useful in DDOS attacks that take advantage of large packets. Simply put, it directs computer network connections to specific server frameworks. Because no computer can send requests powerful enough to overwhelm server bandwidth, they cause computer networks to send unnecessary packets to pre-selected servers. These are the types of packets that LOIC support:

\* TCP

\* UDP

\* HTTP GET (HTTP Floods attack)

1. HOIC

This attack tool stands for High Orbit Ion Cannon. This tool was created by hackers for more refined bulk attacks. This tool can generate more request at a faster interval rate than LOIC can, and the types of packets sent are different. HOIC can execute DDoS attacks by sending HTTP POST and HTTP GET requests.

This makes it possible to carry out large-scale volume web-based attacks in less time. HOICs have been known to generate up to 250 garbage packets and send them as requests to servers. As previously stated, attacks with this tool do not generate more bulk, but they do become more difficult to detect as their sophistication increases. Essentially, some firewall systems (particularly traditional ones) would be incapable of filtering out these attacks.

1. HULK

This is an abbreviation for HTTP Unbearable Load King. In this day and age, where transfer protocols are the standard means for computers to communicate with servers, it would be simple for attacks to overwhelm servers with a large number of seemingly legitimate requests. That's precisely what HULK does. HULK is very efficient at carrying out denial of service attacks because it can send obfuscated traffic, making it harder to detect. Finally, these tools can be used to create attacks that are difficult (and sometimes impossible) to detect. Of course, this is to be expected given that the execution vehicle is HTTP.

1. Pyloris

This tool is extremely useful for carrying out stealth DDOS attacks. It is yet another tool used for pen-testing by performing slow and hidden attacks. It is used to launch denial-of-service attacks against the service. It makes use of server communication frameworks that employ encryption and anonymity. SSL and SOCK proxies are notable examples. It applies these techniques to popular server protocols such as HTTP, UDP, and others.

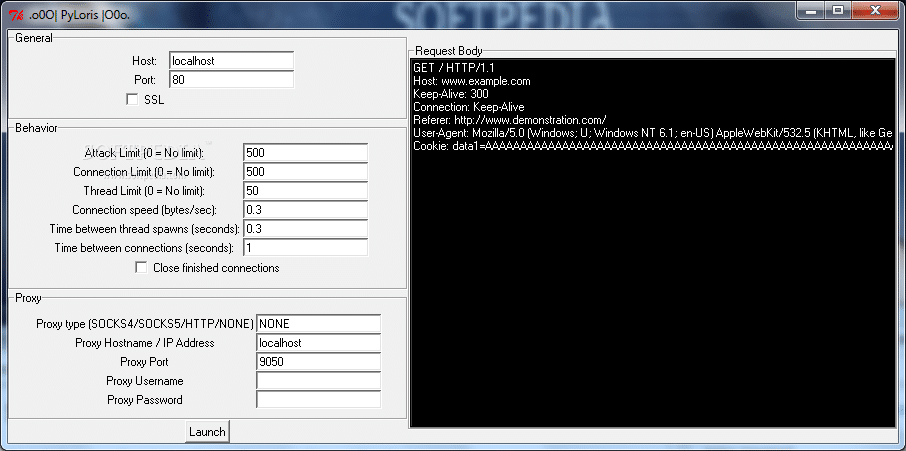


Figure 2.3.4: Pyloris DDoS attack.

1. SLOWLORIS

This tool is one of the most effective DDOS attack tools. It is even referred to as the most effective tool available. It operates by sending legitimate HTTP requests, albeit inadvertently. As a result, with limited bandwidth, the server becomes overburdened with requests. Additionally, this tool enables attackers to establish a connection with the victim server and keep these connections open for as long as necessary. Because the server is anticipating several uncompleted HTTP requests, it has limited or no space to attend to the users. It is particularly effective because the HTTP requests are not spoofed, but are genuine.

1. R.U.D.Y (R-U-Dead-Yet)

R.U.D.Y. is another low and slow attack tool that allows the user to launch attacks with a simple point-and-click interface. The attack aims to gradually overwhelm the targeted server by opening multiple HTTP POST requests and then keeping those connections open for as long as possible.

**2.4 IoT DoS/DDoS attack classification based on layered structure**

**2.4.1 Perception Layer**

**2.4.1.1 Security Vulnerabilities at Perception Layer**

This layer, also known as the sensing layer, is dependent on physical resources as part of the IoT. It collects data using various sensing technologies and devices, converts it to digital signals, and sends it to the network layer. RFID tags, cameras, wireless sensor networks (WSN), GPS, and Bluetooth are examples of perception layer technologies. These devices were chosen based on the functionality of IoT applications. Data collected from the surrounding environment can take many forms, including motion, light, temperature change, and location.

End users are presented with sensors and devices in the perception layer. These devices are designed to increase flexibility, lower costs, and have limited computing and storage resources. Furthermore, they have a data transmission rate limit. Intruders take advantage of this constraint.

**2.4.1.2 Famous types of attacks on this layer:**

\* RF Jamming Attack: Because most wireless devices communicate using radio frequency (RF) signals, their signals can be jammed by other stronger signals. The attacker intercepts and blocks communication between the sensor, or tag, and the data reader.

\* Eavesdropping: mainly affects the confidentiality part of IoT devices. It is a dangerous attack because an attacker can read and collect secret information committed between the tag and the reader of data, and take advantage of the information gathered. This confident information could be phone calls, text messages, or video conferences.

**2.4.1.3 Security Solution at Perception Layer**

Porambage et al. have discussed a lightweight pervasive authentication (PAuthKey) method. This algorithm was created with the idea of sensor resource scarcity in mind. Furthermore, the key establishment process was refined as a result of it. This PAuthKey system enables users to establish secure connections to sensor nodes at a lower cost.

1. Rani et al. discuss possible countermeasures against RFID and WSN attacks. One proposed anti-jamming countermeasure in the paper is to regulate transmitted power and Frequency Hopping Spread Spectrum (FHSS). It is a powerful solution for avoiding interference and multi-path fading (distortion), lowering narrowband interference, increasing signal capacity, and improving the signal-to-noise ratio.

Lin Hu et al. investigated secrecy-enhancing techniques to reduce Secrecy Outage Probability (SOP) caused by eavesdropping at the perception layer. It resulted in improved security at a lower cost than other methods available.

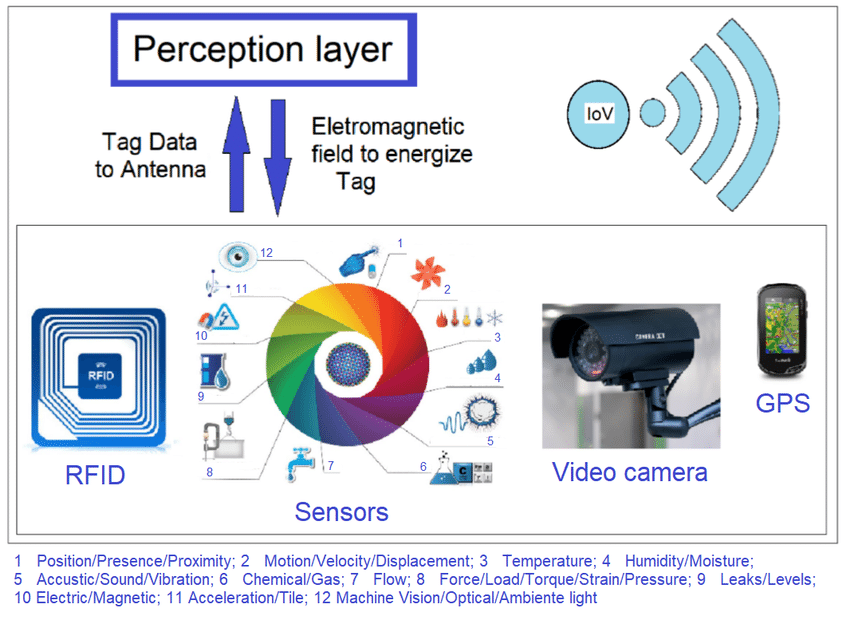


Figure 2.4: IoT Perception Layer

**2.4.2 Network Layer**

**2.4.2.1 Security Vulnerabilities at Network Layer**

This layer operates similarly to the TCP/IP network layer and faces the same communication network security issues that affect data confidentiality, availability, and integrity. It is in charge of transmitting data collected from perception layer devices and sensors.

**2.4.2.2 Famous types of attacks on this layer:**

\* Flooding Attacks: Many useless packets are sent through the network in this type of attack, rendering the target system unreachable. The system is drained specifically by a large number of requests from the attacker, such as a UDP flood. Because the attacker floods different UDP (User Datagram Protocol) packets on different victim ports, the server host will repeatedly inspect these ports for incoming requests, causing victim resources to be depleted.

\* Reflection-based Flooding Attacks: In this type of attack, the attacker intercepts the authentic connection and sends repeated bogus requests to reflectors. The target system is rendered unreachable because all of these reflectors react to it at once.

**2.4.2.3 Security Solution at Network Layer**

In the case of traditional IPv6, it is a tried-and-true method for securing networks known as IPsec. Since IoT devices are added to the Internet using IPv6 over Low-power Wireless Personal Area Networks (6LoWPAN), Raza et al. introduced a method to secure the IoT based on the tested IPsec extension added to LoWPAN. Furthermore, Encapsulation Security Payload (ESP) and Authentication Header (AH) techniques are used to secure communication from application layer devices to network layer.

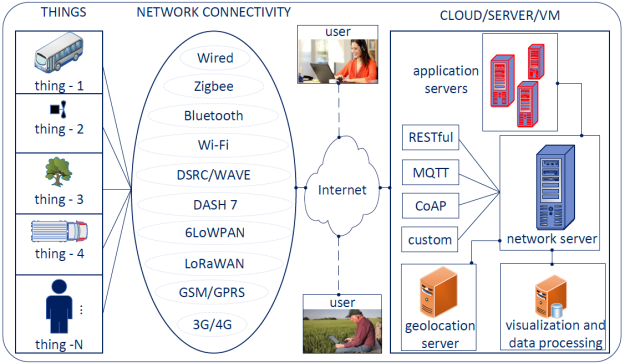


Figure 2.4.1: Network architecture of IoT

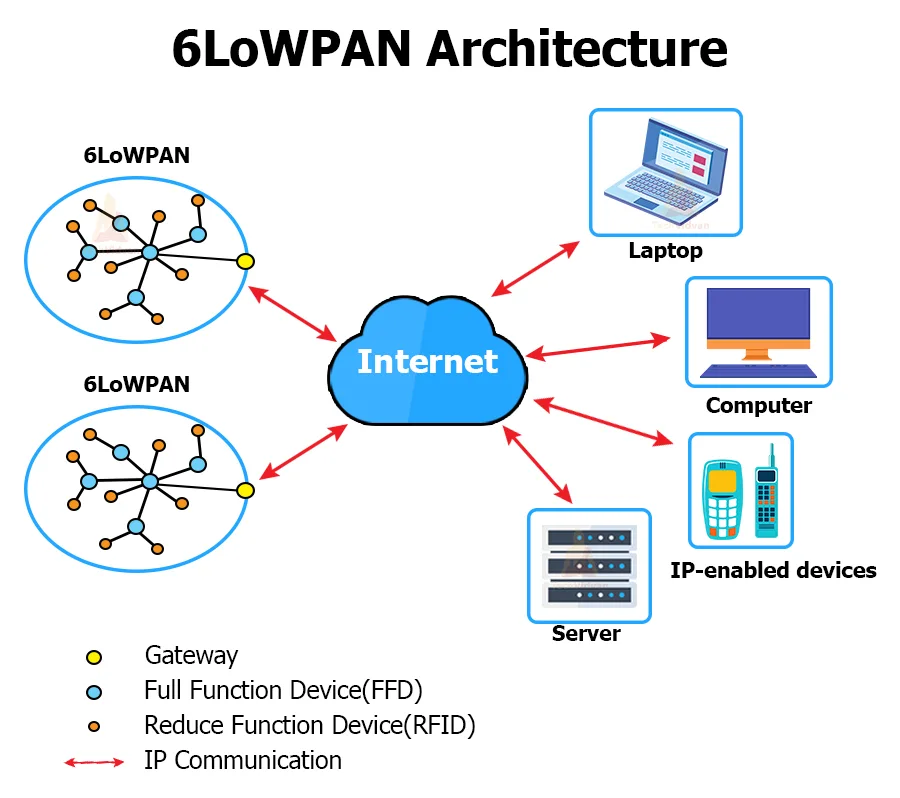


Figure 2.4.2: Low-power Wireless Personal Area Networks (6LoWPAN) Architecture

**2.4.3 Application Layer**

**2.4.3.1 Security Vulnerabilities at Application Layer**

This is the top layer; it is in charge of the logical part of an IoT application. In other words, this layer will manipulate data and display it to end users via a user interface (UI). This layer faces various security challenges, such as access permissions and authentication, which are very likely to be hacked because they are difficult to maintain across various types of applications and users. Furthermore, hackers may exploit application layer vulnerabilities such as buffer overflow, cross-site scripting, and SQL injection, making data privacy and protection difficult to maintain.

**2.4.3.2 Famous types of attack on this layer:**

Reprogramming Attack: If an attacker has unauthorized access, they may change the program code, resulting in data leakage. They can change the code to suit their needs if they have access to the program's source code. Furthermore, if they use an infinite loop in the code, the server resources will be depleted.

Path-Based DoS Attack: A Path-based DoS attack is carried out by flooding multi-hop end-to-end communication paths with data packets.

**2.4.3.3 Security Solutions at Application Layer**

Cirani et al. proposed an authorization framework based on integration with an external Open Authorization Service to address the issue of authentication in the application layer (OAS). The entire solution is known as IoT-OAS, and it focuses on HTTP and CAP (Constrained Application Protocol) services. This method allows for flexible and simple integration with existing services while also reducing processing load.

**2.5 Overview of Amazon Web Services (AWS)**

Amazon Web Services (AWS) is comprised of numerous cloud computing products and services. Amazon's extremely profitable division offers servers, storage, networking, remote computing, email, mobile development, and security. AWS's main products are Elastic Cloud Compute (EC2), Amazon's virtual machine service; Glacier, a low-cost cloud storage service; S3, Amazon's storage system; and AWS IoT Core for Internet of Things connectivity.

AWS has surpassed its competitors in terms of size and presence in the computing world. According to one independent analyst, AWS had 32.4% percent of the market in the first quarter of 2021, with Azure at 20.2% percent and Google Cloud at 9 percent.

**2.5.1 AWS IoT Core**

AWS IoT offers cloud services for connecting your IoT devices to other devices and AWS cloud services. AWS IoT offers device software to assist you in integrating your IoT devices into AWS IoT-based solutions. If your devices can connect to AWS IoT, AWS IoT can connect them to AWS's cloud services. This service lets you select the most suitable and up-to-date technologies for your IoT solutions.

AWS IoT Core supports the following protocols to assist you in managing and supporting your IoT devices in the field:

\* MQTT (Message Queuing and Telemetry Transport): MQTT is a lightweight and widely used messaging protocol designed for devices with limited resources. With some exceptions, AWS IoT MQTT support is based on the MQTT v3.1.1 specification.

\* MQTT over WSS (Websockets Secure): AWS IoT Core supports device connections identified by a client ID that use the MQTT protocol and MQTT over WSS.

\* HTTPS (Hypertext Transfer Protocol - Secure): Clients can publish messages by sending HTTP 1.0 or 1.1 requests to the REST API. HTTP requests use authentication and port mappings. HTTPS, unlike MQTT, does not support a clientID value. So, while a clientID is available with MQTT, it is not available with HTTPS.

\* LoRaWAN (Long Range Wide Area Network): AWS IoT Core for LoRaWAN is a fully managed LoRaWAN network server (LNS) that supports gateway management via Configuration and Update Server (CUPS) and Firmware Updates Over-The-Air (FUOTA). You can use AWS IoT Core for LoRaWAN to replace your private LNS and connect your LoRaWAN devices and gateways to AWS IoT Core. This reduces maintenance, operational costs, setup time, and overhead costs.

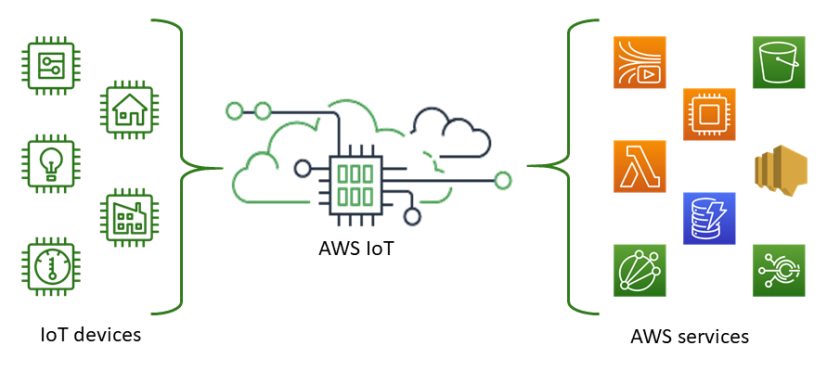


Figure 2.5: AWS IoT Core

To publish and subscribe to messages, the AWS IoT Core message broker supports devices and clients that use the MQTT and MQTT over WSS protocols. It also supports devices and clients that publish messages using the HTTPS protocol.

AWS IoT Core for LoRaWAN connects and manages wireless LoRaWAN (low-power long-distance Wide Area Network) devices. AWS IoT Core for LoRaWAN eliminates the need for you to build and maintain a LoRaWAN Network Server (LNS).

**2.5.2 How your devices and applications connect to AWS IoT**

AWS IoT service provides the following interfaces for accessing your Apps and Devices.

\* AWS IoT Device SDKs: You can create applications on your devices that send and receive messages from AWS IoT.

\* AWS IoT API: Allows you to create IoT applications that use HTTP or HTTPS requests. These API actions enable you to create and manage thing objects, certificates, rules, and policies programmatically.

\* AWS SDKs: You can create IoT applications by utilizing language-specific APIs. These SDKs wrap the HTTP/HTTPS API and allow you to program in any of the supported languages.

\* AWS Command Line Interface (AWS CLI): You can run commands for AWS IoT on Windows, macOS, and Linux. You can use these commands to create and manage things like objects, certificates, rules, jobs, and policies.

\* AWS IoT Core for LoRaWAN: Using AWS IoT Core for LoRaWAN, you can connect and manage your long-range WAN (LoRaWAN) devices and gateways.

**2.5.3 How AWS IoT Works:**

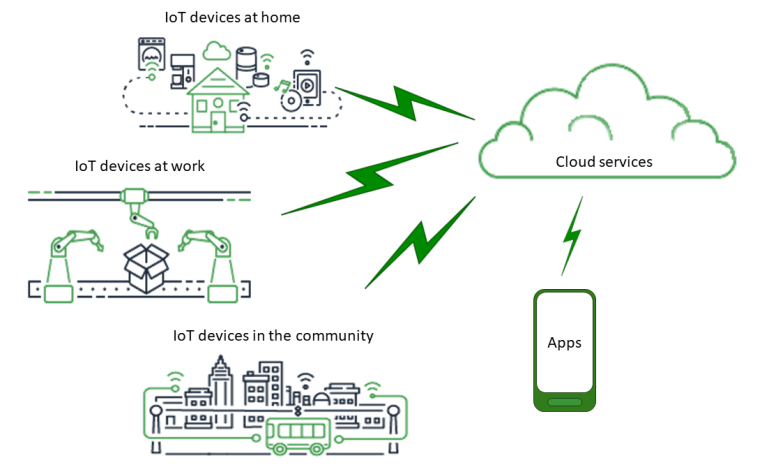


Figure 2.5.1: AWS IoT Universse

Apps:

Apps provide end users with access to IoT devices as well as the features offered by the cloud services to which those devices are linked.

Cloud Services:

Cloud services are internet-connected, distributed, large-scale data storage and processing services. Examples include:

1. IoT connection and management services
2. AWS IoT is an example of an IoT connection and management service. Compute services, such as Amazon Elastic Compute Cloud and AWS Lambda.
3. Database services, such as Amazon DynamoDB

Communications

Different technologies and protocols are used by devices to communicate with cloud services. Examples include:

1. Wi-Fi/Broadband internet.
2. Narrow-band cellular data
3. Broadband cellular data.
4. Long-range Wide Area Network (LoRaWAN)
5. Proprietary RF communications.

Devices

Devices are pieces of hardware that manage interfaces and communications. Devices are typically placed near the real-world interfaces they constantly monitor. Microcontrollers, CPUs, and memory are examples of computing and storage resources that can be found in devices. Examples include:

1. Raspberry Pi.
2. Arduino.
3. Voice-interface assistants.
4. LoRaWAN and devices.
5. Amazon Sidewalk devices.
6. Custom IoT devices.

Interfaces

An interface is a component that connects a device to the physical world.

User Interfaces: These are components that allow devices and users to communicate with each other.

Input interfaces: Allow a user to interact with a device.

Examples include keypads and buttons.

Output interfaces: These interfaces allow a device to communicate with a user.

Examples: Alpha-numeric display, graphical display, indicator light, and alarm bell are some examples.

Sensors

Sensors are Input components that measure or sense something in the real world in a way that a device can understand. Examples include:

\* Temperature sensor: This sensor converts temperature to an analog or digital signal.

\* Humidity Sensor: This sensor converts relative humidity to an analog or digital signal.

\* Analog to Digital converter sensor: This is used to convert analog voltage to a numeric value.

\* Ultrasonic distance measuring unit: This sensor converts a distance to a numeric value.

\* Optical sensors: This sensor converts a light-level to a numeric value.

\* Camera: Converts an image data to a digital data.

Actuators

An actuator is a machine component that's in charge of moving and controlling a mechanism or system, such as opening a valve. In layman's terms, it is a "mover."

Example include:

\* Stepper Motors: Is used to convert electric signals to movement.

\* Relays: Is used to control high electric voltages and currents.

**2.6 What is Raspberry Pi?**

The Raspberry Pi is a low-cost, credit card-sized computer that connects to a computer monitor or television via HDMI and operates with a standard keyboard and mouse. It is capable of running a variety of operating systems, including Raspbian (Debian Linux), Android, Windows 10, IoT Core, and others. It is a capable little device that allows people of all ages to experiment with computing and learn to programme in languages such as Scratch and Python. It can do everything a desktop computer can, from browsing the internet and watching high-definition video to creating spreadsheets, word processing, and playing games.

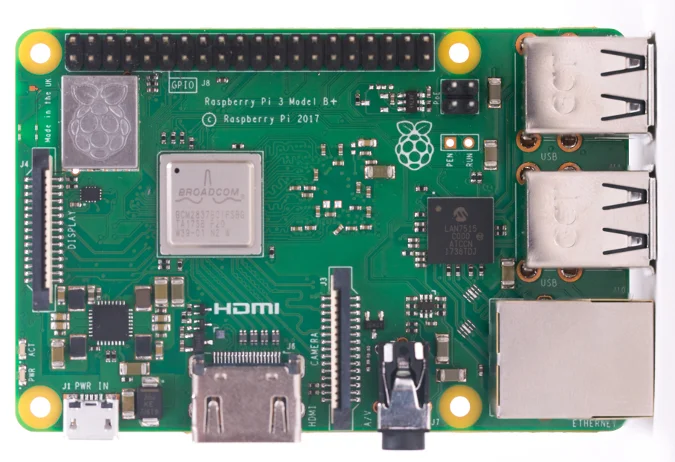


Figure 2.6: A Raspberry Pi 3.

The Raspberry Pi line has gone through several generations, beginning with the Pi 1 and ending with the Pi 400. Almost every generation has had a Model A and a Model B. Model A has been a less expensive variant, with less RAM and fewer ports (such as USB and Ethernet). The Pi Zero is a smaller and less expensive version of the original (Pi 1) generation. So far, here's the lineup:

* Pi 1 Model B (2012)
* Pi 1 Model A (2013)
* Pi 1 Model B+ (2014)
* Pi 1 Model A+ (2014)
* Pi 2 Model B (2015)
* Pi Zero (2015)
* Pi 3 Model B (2016)
* Pi Zero W (2017)
* Pi 3 Model B+ (2018)
* Pi 3 Model A+ (2019)
* Pi 4 Model A (2019)
* Pi 4 Model B (2020)
* Pi 400 (2021)

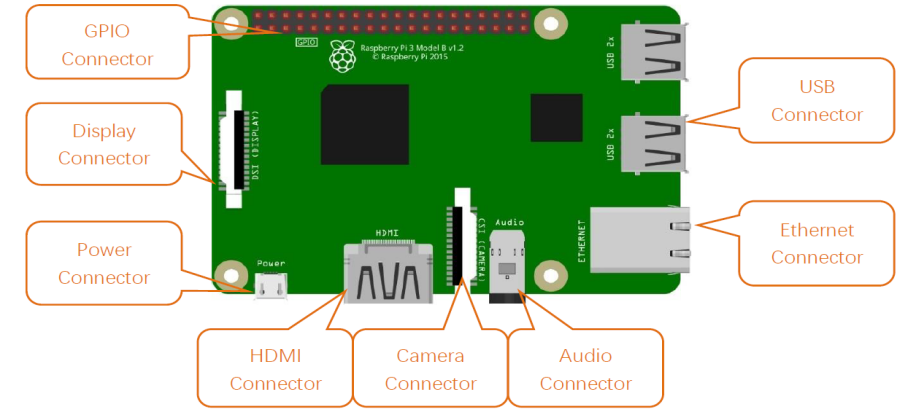


Figure 2.6.1: Hardware interface diagram of RPi3B+/3B/2B/1B+.

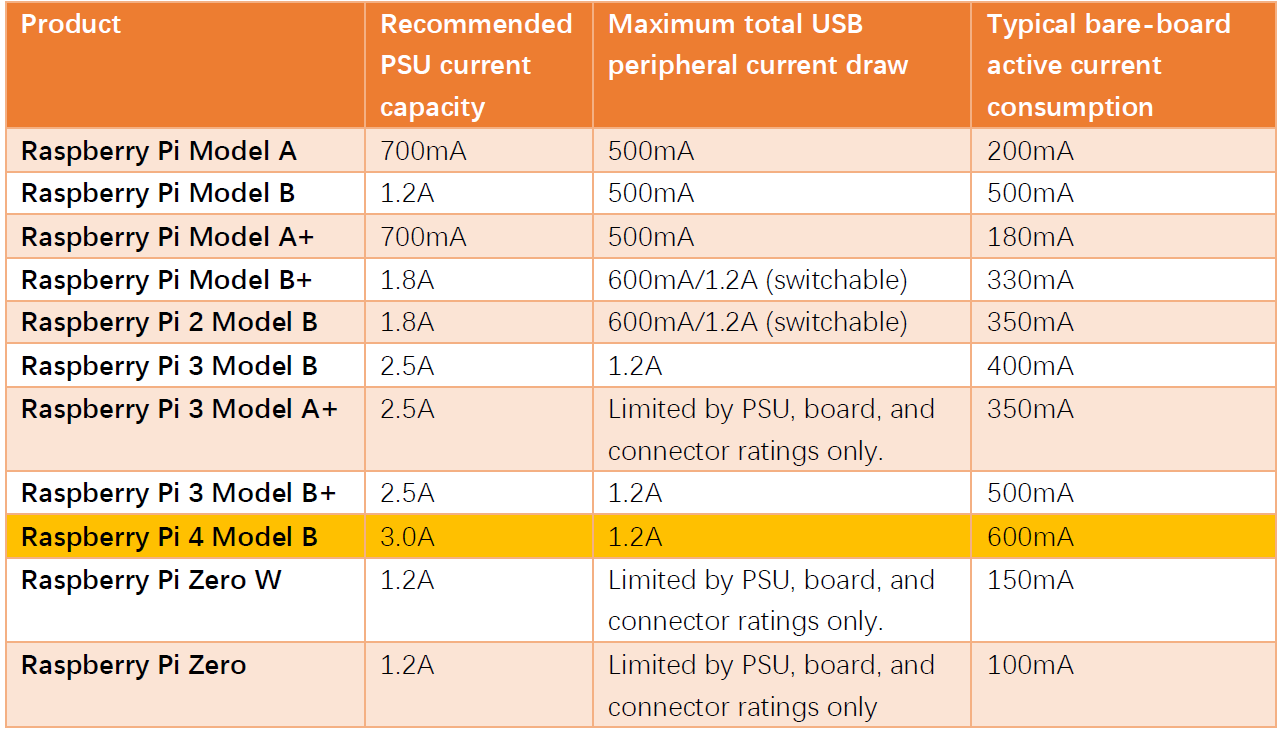


Figure 2.6.2: Power requirements of various versions of Raspberry Pi.

CHAPTER THREE

METHODOLOGY

**3.1 Introduction**

This chapter discusses developing the research methodology required to complete the study's experimentation portion. As a result, this study investigated the extent to which a DoS/DDoS attack could impair the performance of cloud computing services via IoT.

This chapter is further divided into 3 sections. The Raspberry Pi and Ultra Sonic Ranging Sensor connectivity (hardware and software) requirements, setup and connections are described in Section 1. Section 2 describes how Raspberry Pi is deployed to the AWS cloud through the AWS IoT Core service. Section 3 goes over brief description of the tools and technologies that we will be using to replicate and investigate this attack for this project, and also discuss about the legal and professional considerations when carrying out this type of attack .

**3.2 System Requirements**

|  |  |  |
| --- | --- | --- |
| Operating System Description | Hardware Description | Software Description |
| Server  Debian OS  (Raspberry Pi 4 models)  (Victim System) | 4 GB RAM memory,  36GB Hard Disk, 2 Core processor. | AWSIoTPythonSDK  Bash,  Python,  SSH,  VNC  Wireshark  Apache Server 2.4 (Raspian) |
| Kali Linux  (Attacker) | 4 GB RAM memory, 100 GB Hard Disk, 2 Core Processor | Nmap, Slowloris, Hping3,  Pyloris, Wireshark |
| Windows 10 Pro  Version 21H1  OS build 19043.1826 | Intel(R) Core(TM) i5-8250U CPU @ 1.60GHz 1.80 GHz | VirtualBox |

## Table 3.0: Software and Hardware Requirement for experiment

**3.3 Raspberry Pi and Ultrasonic Ranging Sensor connectivity:**

**3.3.1 What is HC-SR04 Ultrasonic Sensor?**

The Ultrasonic Ranging Module operates on the same principle as RADAR and SONOR, in that ultrasonic waves are reflected when they encounter obstacles. When an ultrasonic wave is received, time interval counting stops, and the time difference (delta) is the total time of the ultrasonic wave's journey from the transmission to reception. Because the speed of sound in air is constant and around *v=340m/s*, we can use the *formula s=vt/2* to calculate the distance between the Ultrasonic Ranging Module and the obstacle.

This is accomplished by counting the time between when the ultrasonic wave is transmitted and when it reflects back after encountering an obstacle. The HC-SR04 Ultrasonic Ranging Module incorporates an ultrasonic transmitter as well as a receiver. The transmitter converts electrical signals (electrical energy) into high frequency (beyond human hearing) sound waves (mechanical energy), while the receiver does the opposite.



Figure 3.0: HC-SR04 Ultrasonic Ranging Sensor.

**3.3.2 How HC-SR04 Ultrasonic Sensor works?**

Before we wet our pants into the workings of an Ultrasonic Sensor, we need to understand the parts and pins of that sensor.

The HC-SR04 Ultrasonic Sensor is made up of three components: an ultrasonic transmitter, a control circuit, and an ultrasonic receiver. The HC-SR04 Sensor has only four pins, which are VCC, TRIG (Trigger), ECHO (Echo), and GND.

The Sensor's Ultrasonic Transmitter generates a 40 kHz Ultrasound. This signal then travels through the air, and if it encounters an obstacle, it hits the object and bounces back.

The Ultrasonic Receiver then collects the bounced signal. Because you already know the speed of sound, you can calculate the distance of the object based on the time of travel of the signal.

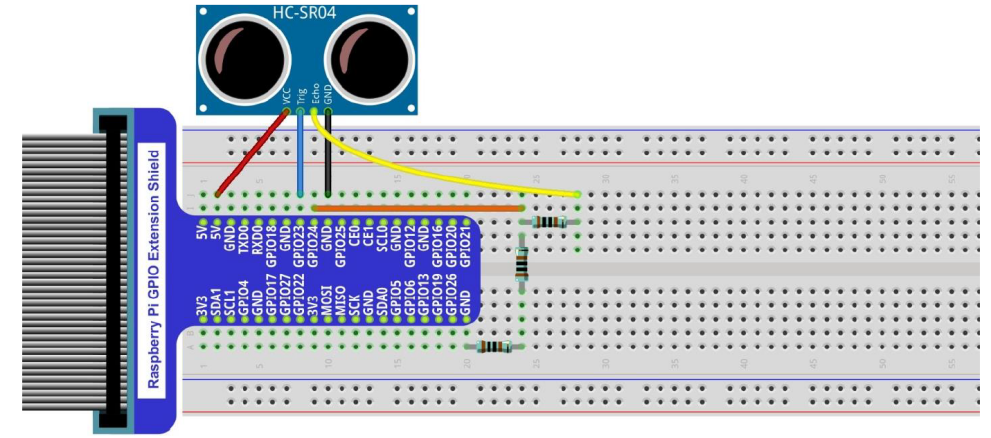


Figure 3.1: Circuit Diagram

Component Required

* Raspberry Pi 3+/4
* HC-SR04 Ultrasonic Sensor
* Mini Breadboard (with 40 pin GPIO)
* Jumper Wires x4
* 1kΩ Resistor x3.
* Power Supply
* Computer with SSH or VNC software enabled.

**3.3.3 Circuit Design**

Connect the HC-SR04 Ultrasonic Sensor's Trig Pin to the Raspberry Pi's Physical Pin 16, or GPIO23. To convert the Echo pin to 3.3V logic (approximately), use a combination of resistors and connect it to Physical Pin 18 (GPIO24) of the Raspberry Pi.

Finally, connect the Raspberry Pi's +5V and GND pins to the Ultrasonic Sensor. The diagram can be found in Fig 3.1.

**3.4 Deploying Raspberry Pi to AWS cloud**

In this section, we will discuss in details on how to deploy a raspberry pi system with ultrasonic range IoT sensor connected to AWS cloud using AWS IoT Core services.

To complete the above mentioned process, we will follow the procedures listed below:

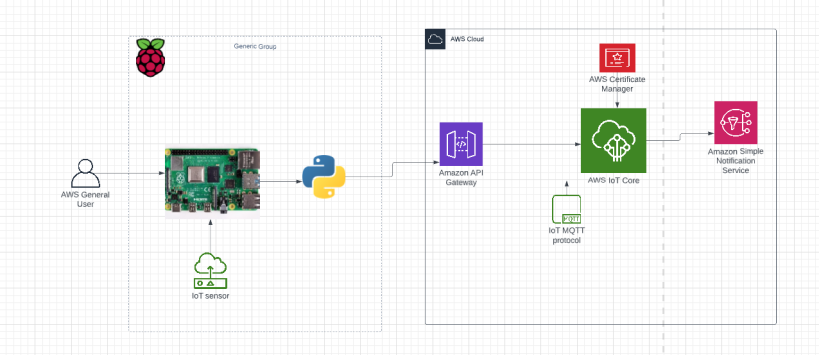


Figure 3.2: Connecting RPi to AWS cloud architecture.

\* Set up your device. (Already done in last section)

To complete this stage user must:

1. User must have an AWS account.
2. Create, register and activate the Device Certificate with AWS IoT.

\* The certificate is frequently created alongside and attached to an AWS IoT thing object. While a thing object is not required for a device to connect to AWS IoT, it does provide the device with additional AWS IoT features.

\* A policy that's attached to the device certificate and authorizes it to connect to AWS IoT Core and perform all of the actions you specify.

\* An internet connection that can access your AWS account’s device endpoints.

\* Download the necessary certificates from AWS IoT

\* Download and Install the required tools and libraries for the AWS IoT device SDK (Software Development Kit).

sudo apt-get update

sudo apt-get upgrade

sudo apt-get install cmake

sudo apt-get install libssl-dev

sudo apt install python3-pip

Pip install AWSIoTPythonSDK==1.0.0

\* Review the MQTT protocol.

MQTT communicates using a publish/subscribe model. The MQTT protocol communicates with its host via a publish/subscribe model. This model differs from the HTTP request and response model. Devices use MQTT to establish a session with the host, which is identified by a unique client ID. To send data, devices publish messages identified by topics to the host's message broker. Devices subscribe to topics to receive messages from the message broker by sending topic filters in subscription requests to the message broker.

\* Create a script to communicate and connect your device with AWS IoT Core.

Below script is used to communicate with AWS IoT Core services and the Raspberry Pi. This script was written in python3 programming language.

*import RPi.GPIO as GPIO*

*import time*

*from AWSIoTPythonSDK.MQTTLib import AWSIoTMQTTClient*

*GPIO.setwarnings(False)*

*#GPIO Mode (BOARD / BCM)*

*GPIO.setmode(GPIO.BCM)*

*GPIO.cleanup()*

*myMQTTClient = AWSIoTMQTTClient("FrancisClientID") #random key, if another connection using the same key is opened the previous one is auto closed by AWS IOT*

*myMQTTClient.configureEndpoint("a1nxkyamzlj3tu-ats.iot.us-east-1.amazonaws.com", 8883)*

*myMQTTClient.configureCredentials("/home/pi/Desktop/iot\_Certificates/AmazonRootCA1.pem", "/home/pi/Desktop/iot\_Certificates/iot-private.pem.key", "/home/pi/Desktop/iot\_Certificates/iot-certificate.pem.crt")*

*myMQTTClient.configureOfflinePublishQueueing(-1) # Infinite offline Publish queueing*

*myMQTTClient.configureDrainingFrequency(2) # Draining: 2 Hz*

*myMQTTClient.configureConnectDisconnectTimeout(10) # 10 sec*

*myMQTTClient.configureMQTTOperationTimeout(5) # 5 sec*

*print ('Initiating Realtime Data Transfer From Raspberry Pi...')*

*myMQTTClient.connect()*

*def distance():*

*#set GPIO Pins*

*GPIO\_TRIGGER = 23*

*GPIO\_ECHO = 24*

*#set GPIO direction (IN / OUT)*

*GPIO.setup(GPIO\_TRIGGER, GPIO.OUT)*

*GPIO.setup(GPIO\_ECHO, GPIO.IN)*

*# set Trigger to HIGH*

*GPIO.output(GPIO\_TRIGGER, True)*

*# set Trigger after 0.01ms to LOW*

*time.sleep(0.00001)*

*GPIO.output(GPIO\_TRIGGER, False)*

*StartTime = time.time()*

*StopTime = time.time()*

*# save StartTime*

*while GPIO.input(GPIO\_ECHO) == 0:*

*StartTime = time.time()*

*# save time of arrival*

*while GPIO.input(GPIO\_ECHO) == 1:*

*StopTime = time.time()*

*# time difference between start and arrival*

*TimeElapsed = StopTime - StartTime*

*# multiply with the sonic speed (34300 cm/s)*

*# and divide by 2, because there and back*

*distance = (TimeElapsed \* 34300) / 2*

*return distance*

*if \_\_name\_\_ == '\_\_main\_\_':*

*try:*

*while True:*

*dist = distance()*

*print ("Measured Distance = %.1f cm" % dist)*

*time.sleep(1)*

*myMQTTClient.publish(*

*topic="home/helloworld",*

*QoS=1,*

*payload='{"Sensor":"'+str(dist)+'"}')*

*# Reset by pressing CTRL + C*

*except KeyboardInterrupt:*

*print("Measurement stopped by User")*

*GPIO.cleanup()*

\* Review the results.

The below images displays the output after connecting the raspberry pi to the AWS cloud environment console.

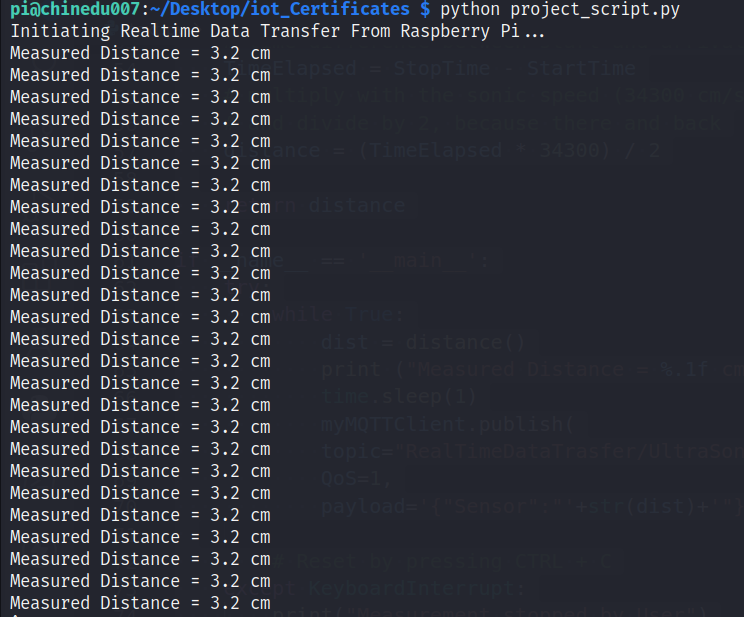


Figure 3.3: Output from Terminal.

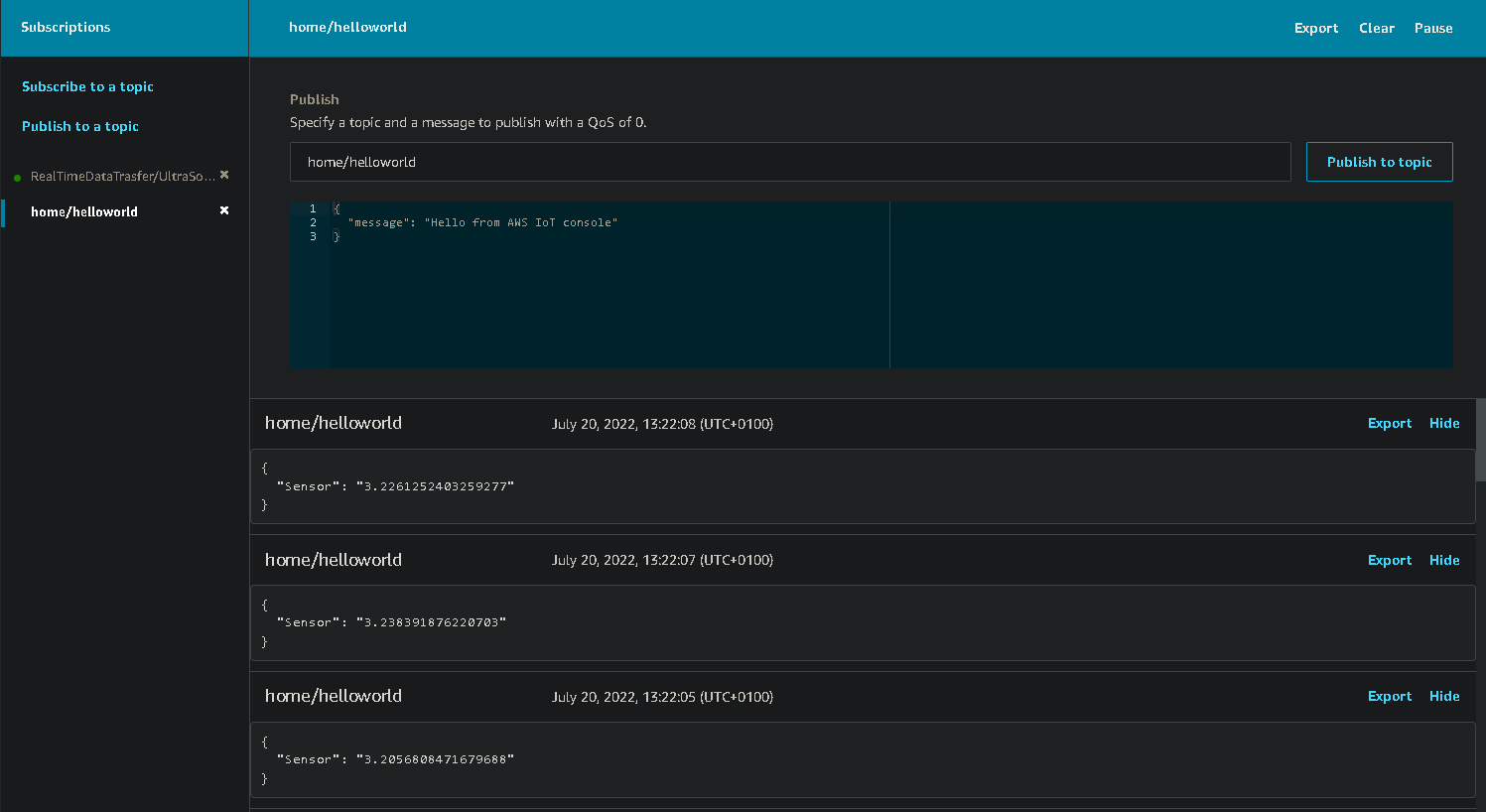


Figure 3.3.1: Output from AWS console.

**3.4 ATTACK GENERATION TOOLS AND TECHNOLOGY USED**

**3.4.1 Nmap**

Nmap, short for Network Mapper and also known as the Swiss Army Knife, is a free and open source tool used for vulnerability scanning, port scanning, and, of course, network mapping.

Nmap is one of the pre-installed tools on Kali Linux that is used for network exploration and auditing. It aids in determining both live systems and system breaches. Nmap on Linux simplifies network reconnaissance for hackers by identifying real-time information networks (Kaur & Kaur, 2017). In this study, this tool was used to launch a series of port scan attacks to find open ports that could be vulnerable to attacks. Nmap UDP Scan, Nmap TCP Scan, and Nmap Service Version Scan are some of the stealth scans used for this project.

**3.4.2 Hping3**

The hping3 tool is a Linux software tool that can display the responses of its targets by sending customised ICMP packets—hoping checks whether a host's network is alive by generating a sequence of numbers that originate from the targeted host's web. Packet-Hoping is used to cause traffic to the hosts' network by sending large amounts of TCP while simultaneously disconnecting the IP address sources. By doing so, hping3 makes the data appear to be from a specific user with a defined network (Horák et al., 2020).

**3.4.3 Slowloris**

Slowloris is a DDoS attack that uses partial HTTP requests to open connections between a single computer and a targeted Web server, then keeps those connections open for as long as possible, overwhelming and slowing down the target. This type of DDoS attack uses very little bandwidth and only affects the target web server, leaving other services and ports unaffected. Slowloris DDoS attacks can target a variety of Web server software, but have shown to be particularly effective against Apache 1.x and 2.x.

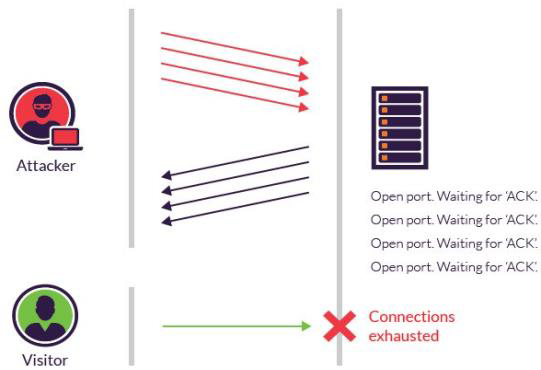


Figure 3.4: Slowloris Attack

**3.4.4 UFONet**

UFONet is a free software, peer-to-peer, and cryptographic disruptive toolkit that enables us to perform DoS and DDoS attacks on Layer 7 (APP/HTTP) by exploiting Open Redirect vectors on third-party websites to act as a botnet, and on Layer 3 (Network) by abusing the protocol. It also functions as an encrypted DarkNET, publishing and receiving content via a global client/server network.

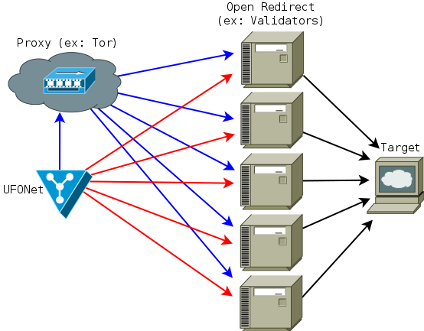


Figure 3.4.1: Ufonet Schema

**3.4.5 Wireshark**

Wireshark is a packet analyzer for networks. A network packet analyzer displays captured packet data as precisely as possible.

Consider a network packet analyzer to be a measuring device for examining what's going on inside a network cable, similar to how an electrician uses a voltmeter to examine what's going on inside an electric cable (but at a higher level, of course).

Wireshark is the most popular and widely used network protocol analyzer in the world. It provides a microscopic view of what's going on in your network and is the de facto standard for many commercial and non-profit enterprises, government agencies, and educational institutions.

## **3.5 Legal, Social, and Professional Considerations**

**3.5.1 Legal Obligations and Liability**

DDoS attacks are prohibited. Unauthorized DDoS attacks are punishable by up to ten years in prison and a $500,000 fine under the Federal Computer Fraud and Abuse Act. Conspiring to do so can result in a 5-year prison sentence and a fine of $250,000.

These serious consequences, however, apply to attacks launched without permission.

**3.5.2 Professional Issues**

Professionals conduct DDoS attack tests under controlled conditions with the client's knowledge and consent.

There also have certain core responsibilities in the context of IoT, as mandated by the GDPR and DPA. These responsibilities include things like Integrity, Authentication, Confidentiality, Security, Privacy, Legislative Knowledge, and so on. (O. Inieke, 2020)

**3.5.3 Social Issues**

The following are the social issues associated with IoT devices: over-reliance on technology, security of digital and physical possessions, human interference, social pressure, and loss of human interaction. (A. Habibipour et al., 2019)

**CHAPTER FOUR**

**EXPERIMENT, RESULTS and ANALYSIS.**

**4.1 Introduction**

In this chapter, we will attempt to replicate the attack surface by launching a DDoS ICMP flood attack on a Raspberry Pi system that hosts the IoT in the cloud. This includes using the Nmap tool to perform a TCP/SYN scan on the system's network to identify open ports that may be vulnerable to attacks before launching a stealth DDoS attack to assess the level of damage our attack has caused to the system's performance. We will examine the outcome of the attack and the extent to which it was successful.

There are numerous risks associated with conducting DoS and DDoS attacks without permission on AWS cloud infrastructure (IoT gateway). Our primary focus will be on replicating the attack at the network layer of the system (Raspberry Pi).

**4.2 Experiment:**

The experiment (attacker) employs a variety of methods to flood malicious packets to the targeted web server. The identification of signatures attack is significant; this allows users to find a way to detect DoS attacks. The method proposed two separate machines, one of which physically houses an attacker simulator. It can carry out several types of attacks on the target machine: one machine is used as an attacker to flood malicious packets to the server machine, which has a tool for monitoring and capturing all traffic in real-time.

The experiment consists of two phases (Port Scanning and DoS attack launch) and each of the phases involves several test carried out on the victim machine.

Systems Description

|  |  |
| --- | --- |
| Attacking Machine | Victim Machine |
| IP Address: 10.0.2.15 | IP Address: 192.168.0.10 |
| System OS: Kali Linux | System OS: Raspbian OS |

**4.2.1 Phase 1: Port Scanning**

The steps involved in achieving this phase includes:

1. Connecting the Raspberry Pi (victim machine), which serves as the server for the Ultrasonic Range Sensor IoT, to a power source and remotely logging into the system via the open SSH channel on port 22.

***System command:*** sudo ssh [pi@192.168.0.10](mailto:pi@192.168.0.10)

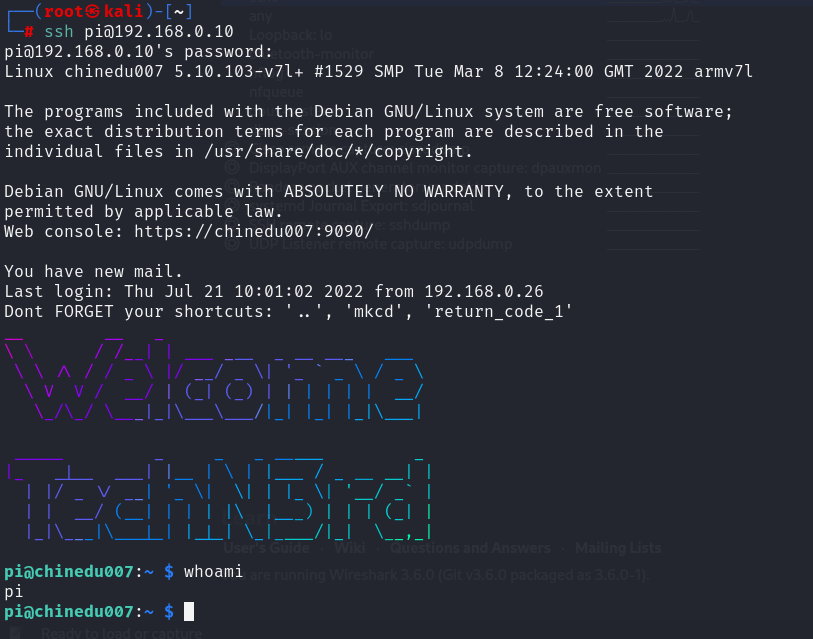


Figure 4.0: Screenshot of Remote SSH into the Raspberry Pi

***System command:*** ifconfig

The *ifconfig* command is used to display the network interfaces.

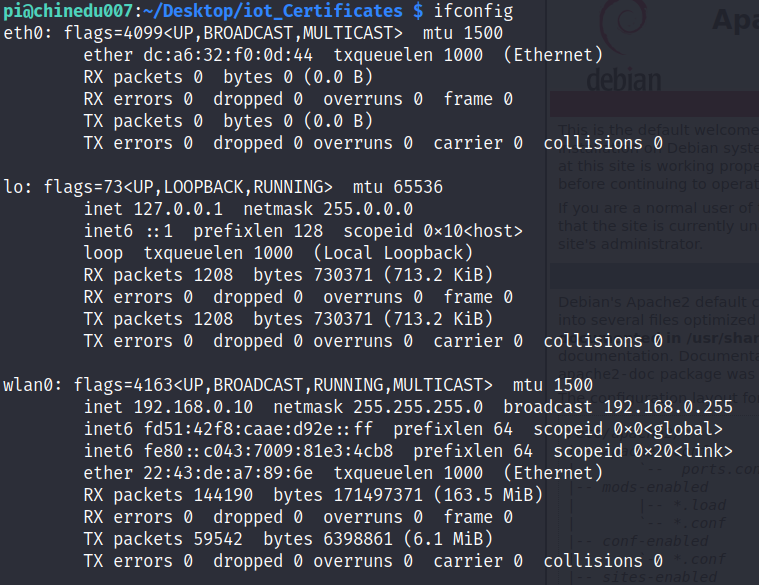


Figure 4.1: Screenshot of Ifconfig

1. Run an ICMP scan on the victim’s network to see if the network is alive.

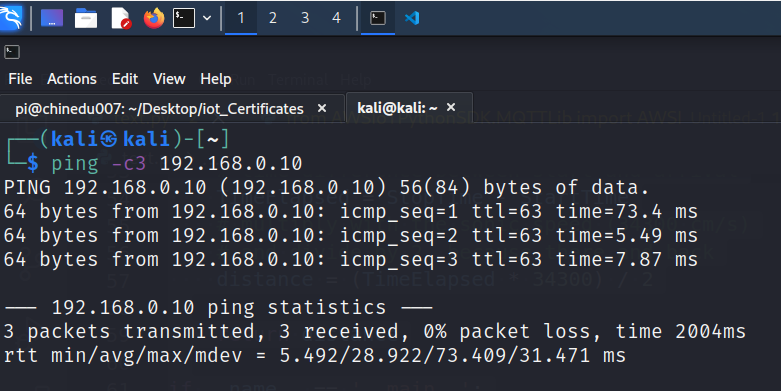


Figure 4.1.1: Screenshot of Ping (Port knocking) on Raspberry Pi from Attacker machine

According to the above results, the network is operational and ping requests are successful.

***System command:***  ping -c3 192.168.0.10

***Parameter Used:***

-c: This parameter stops the ICMP data packet after a specified number of count.

1. Using Nmap and UFONet, we will scan the target for open ports and firewall leaks, as well as perform a vulnerability assessment.

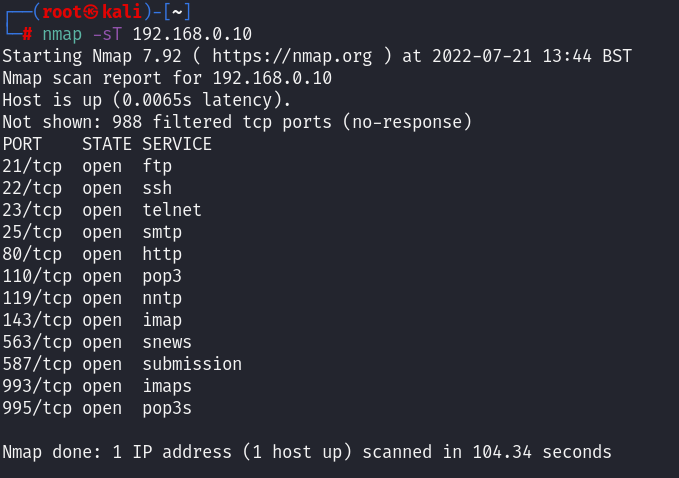


Figure 4.1.2: Screenshot of Nmap Port Scan

According to the above results from the Nmap scan, there are 12 open ports and 988 filtered or closed ports, and the scan was carried out in 104.34 seconds. We can see that port 80 (HTTP) is open and hosting a HTTP web server among the open ports. Because this port is open, we can try a DoS attack with the various tools we've discussed to see if we can slow down the server's speed.

***System command:***  sudo nmap -sT 192.168.0.10

Sudo nmap -sF -sV 192.168.0.10

***Parameter Used:***

-sT: TCP connect scan is the default TCP scan type when SYN scan is not available.

-sV: Probe open ports to determine service/version info.

-sF: TCP FIN scan.

The UFONet vulnerability assessment also reveals that the system is running Apache 2.4.38 (Raspbian) server software and lacks a Web Application Firewall (WAF) and an Internal Detection System (IDS). The scan also detected some Common Vulnerabilities and Exposures (CVE) to which the system is vulnerable.

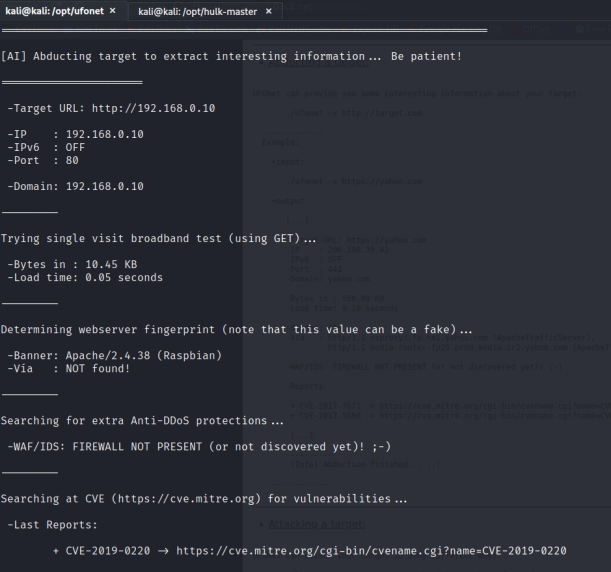


Figure 4.2: Screenshot of Ufonet Vuln Scan

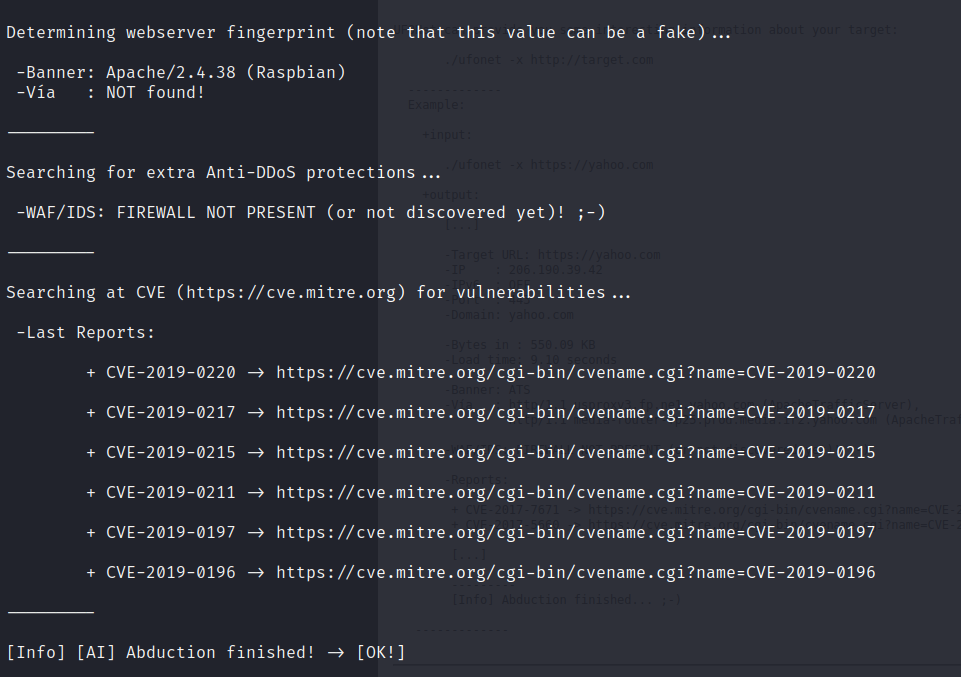


Figure 4.2.1: Screenshot of Ufonet Vuln Scan

***System command***: sudo ufonet -x http://192.168.0.10.

**4.2.2 Phase 2: DoS Attack Launch**

To accomplish this goal, a DoS attack is launched against the victim machine using various open-source DoS/DDoS attack tools.

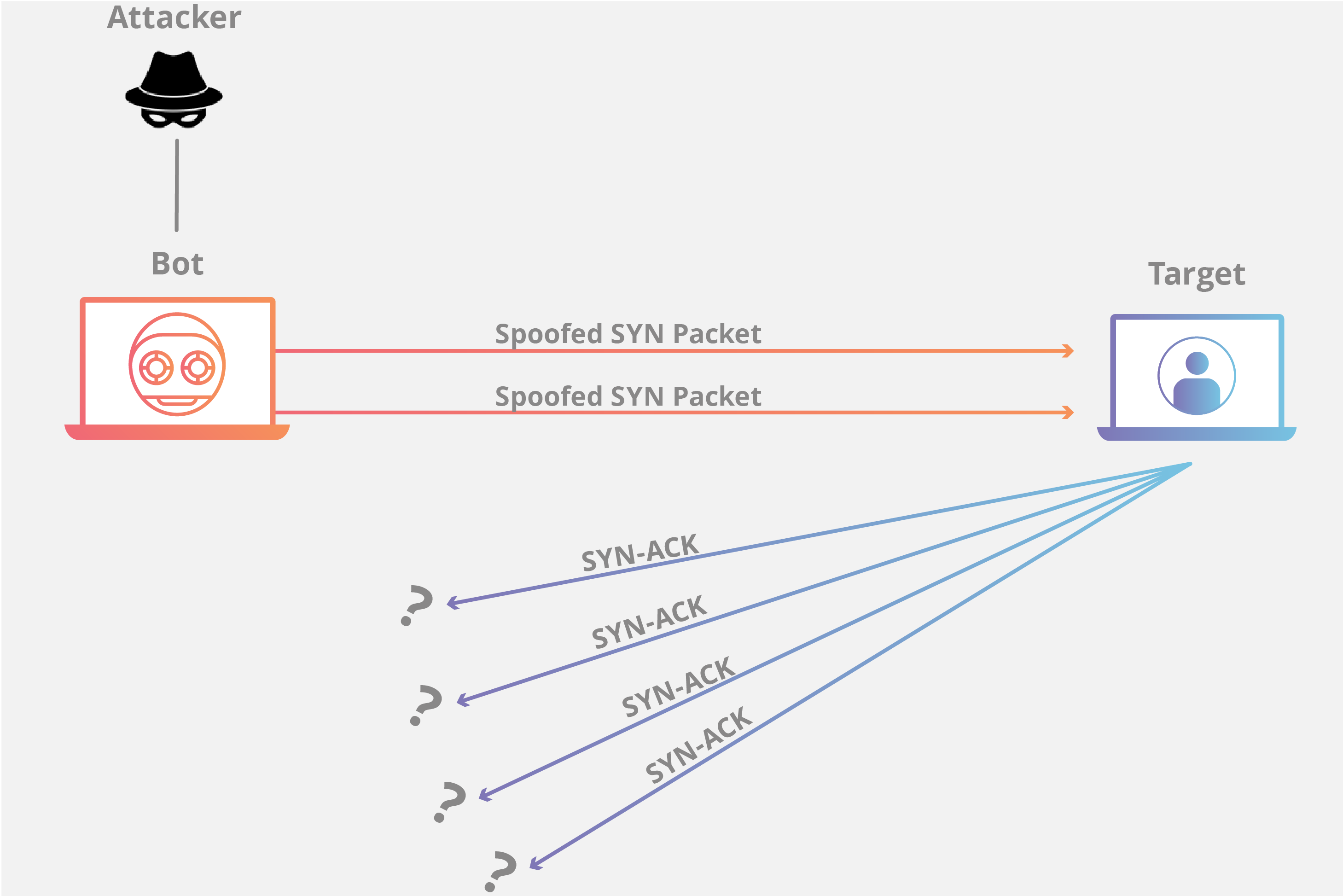


Figure 4.3: Screenshot of DoS attack

**4.2.2.1 DoS attack using Slowloris**

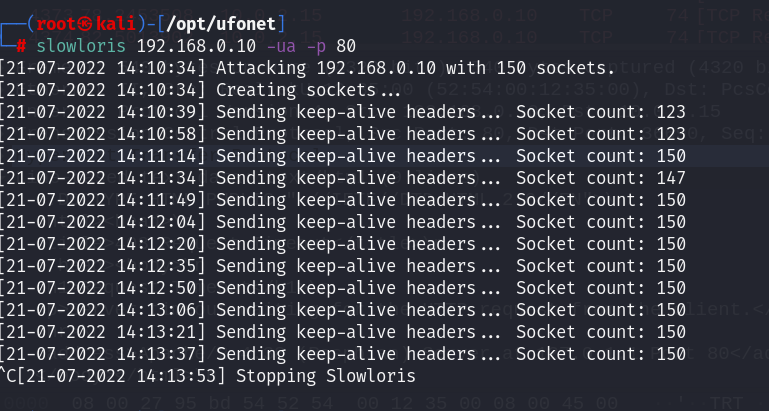


Figure 4.4: Screenshot of Slowloris attack

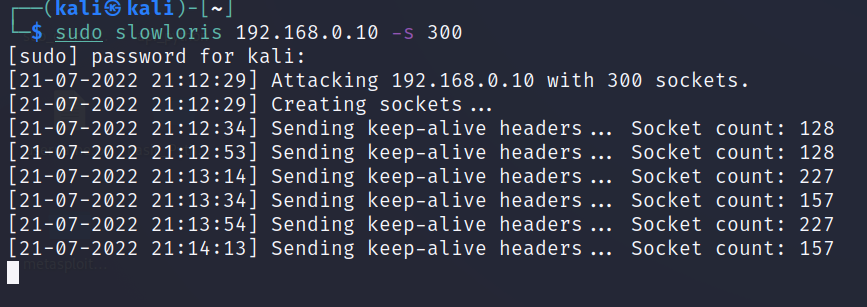


Figure 4.4.1: Screenshot of Slowloris attack

***System command:*** sudo slowloris 192.168.0.10 -ua -p 80

sudo slowloris 192.168.0.10 -s 300

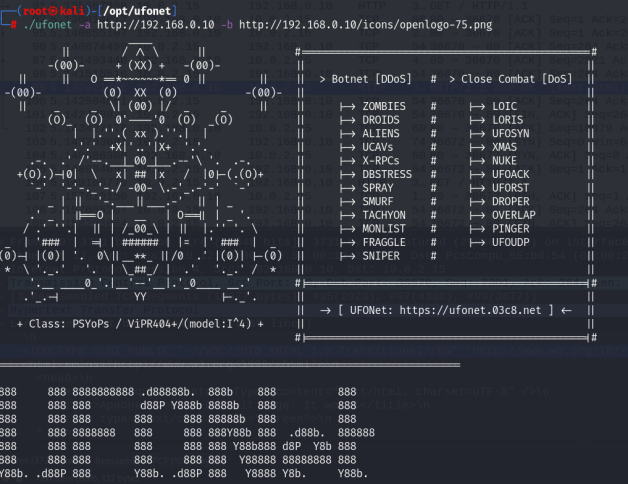
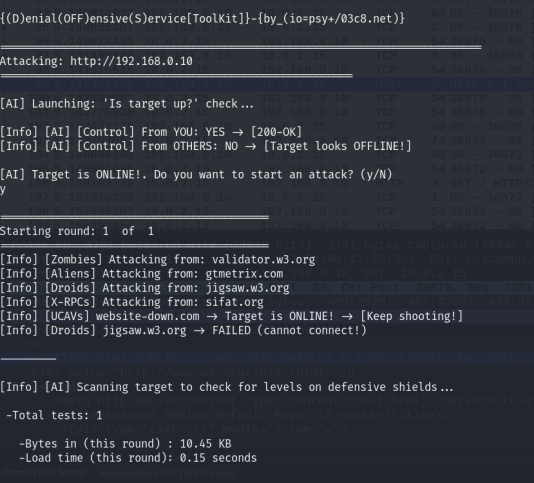
***Parameters Used:***

-ua: Randomizes the user-agents with each request.

-p: Specifies the port of the webserver to attack.

-s: Specifies the number of sockets to use in the test.

**4.2.2.2 DoS attack using UFONet**

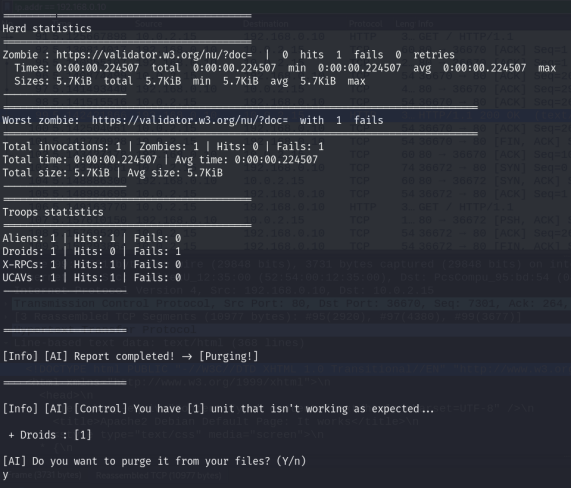


Figure 4.5: Screenshot of Ufonet attack

***System command:*** sudo ufonet -a <http://192.168.0.10> -b <http://192.168.0.10/icons>/openlogo-75.png

Parameters Used:

-a: Target; DDoS attack on a target.

-b: Place; set a place to attack (ex: ‘/path/image.jpg’). ["/icons](http://192.168.0.10/icons)/openlogo-75.png” was discovered after ufonet was used to fuzz the victim machine

**4.2.2.3 DoS attack using Hping3**

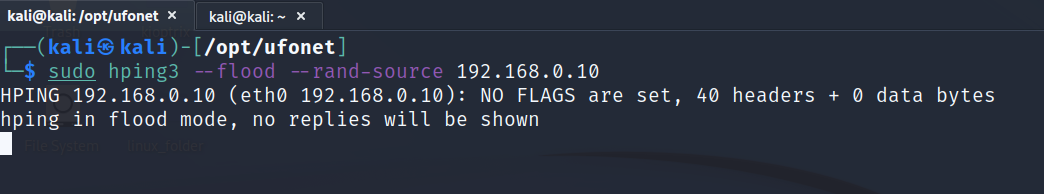


Figure 4.6: Screenshot of Hping3 attack

***System command:*** sudo hping3 --flood --rand-source 192.168.0.10

Parameters Used:

--flood: This parameter is used to send packets as fast as possible. No need to show replies.

--rand-source: This parameter tells hping3 to use the random source IP address mode.

**4.3 Result:**

The experiment's results were examined and discussed in this section.

The TCP SYN flood is one of the most damaging types of DoS attacks. When clients and servers need to communicate, they must first establish connectivity by performing a three-way handshake, "SYN-SYN-ACK and ACK." In this case, an attacker attempts to impersonate a trusted client, and servers continue to wait for acknowledgment until TCP timeout. These attacks were designed to deplete server resources such as firewalls and communication tools. Figure 4.7 depicts a TCP capture and analysis using Wireshark.

As explained in the following section, this method employs packet sniffing to capture ongoing packets, identify, and analyse the behaviour of attacks.

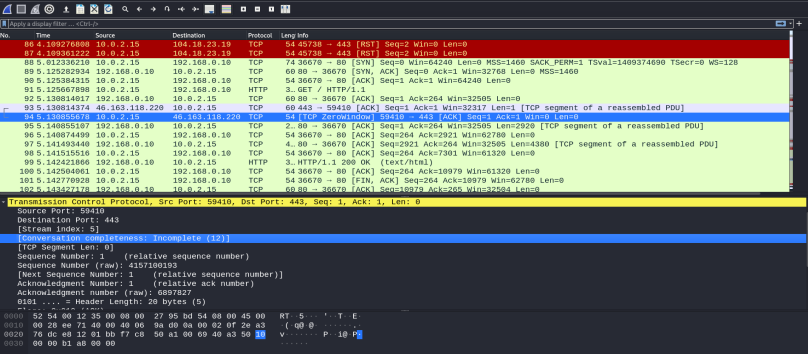


Figure 4.7: Analyzing TCP Flood attack using Wireshark.

Figure 4.7.1 describes the TCP stream packets sent to the target machine. You can see these details by right-clicking on the packet from the TCP connection you’re interested in, clicking on “Analyze” > "Follow TCP Stream." The screenshot shows that bad requests were made to the target server.

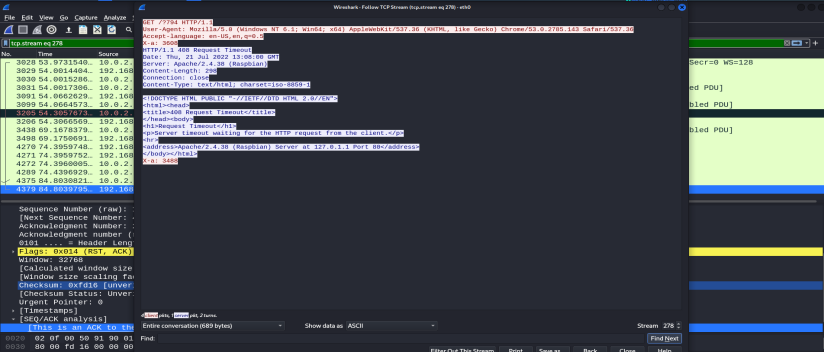


Figure 4.7.1: Screenshot of TCP Streams

The Packet Counter menu shows the total number of packets for each status code in Figure 4.7.2. The rate at which the packet is being sent is measured in milliseconds (ms). The total HTTP packets sent was 512, with HTTP Bad Request taking a chunk of the packets captured.

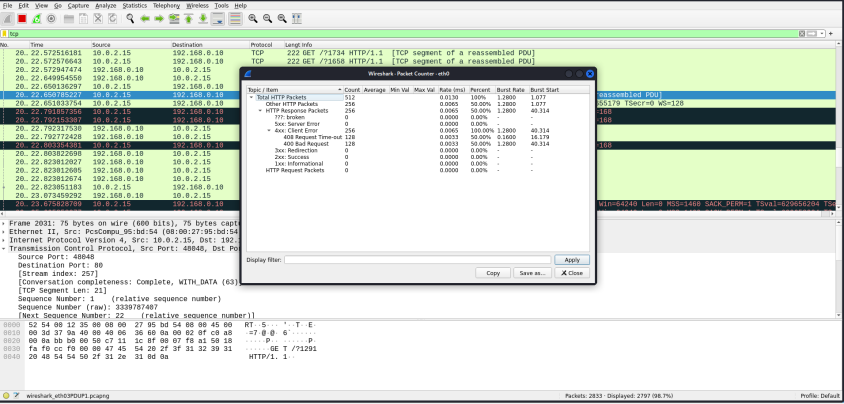


Figure 4.7.2: Screenshot of Packet Counter.

TCP flooding of (DDoS) attacks results in packets being sent to the victim server. By viewing the information details of malicious packets, you can see the packet sequence graphically by selecting them from the menu "Statistics," > Flow Graph. This tool allows you to follow the TCP connections and behaviour shown in Figure 4.7.2.

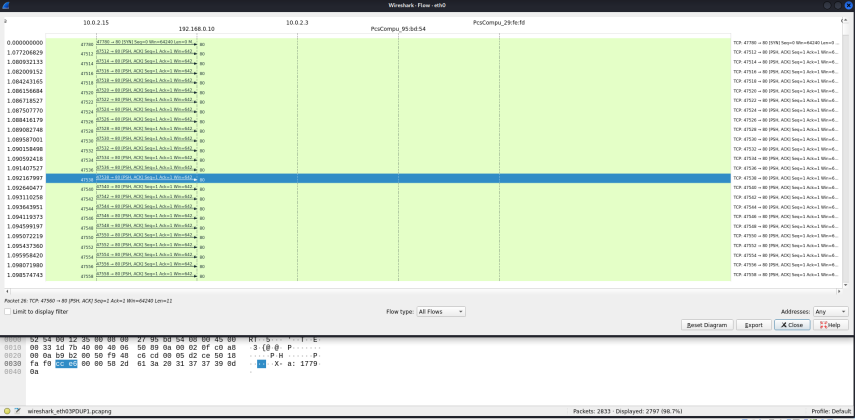


Figure 4.7.2: Screenshot of TCP Flow Graph

According to Figure 4.7.2, the time is in seconds (s), the source's IP address is 10.0.2.15 (attacker's machine), and the port number is random between 47558 and 47780 port. The destination's IP address is 192.168.0.10, and the port number is 80 p. (port). The source sends attack packets with an unspecified port number here. The client's IP address (10.0.2.15) establishes a TCP connection with the victim's IP address (192.168.0.10), which is referred to as a server. Through Wireshark traces, we were able to identify some suspicious downloads (PSH ACK and TCP DUP ACK) as belonging to abnormal packets. For example, the attacker can use PSH ACK to create an attack similar to TCP ACK attacks.

**4.4 Analysis:**

It was discovered during the attack execution that the traffic flow in loading the website on the victim's machine became slow over time, resulting in the system shutting down automatically. However, when examining whether the attack affected the flow of packets towards the ICMP ping, the results show that, while the attack affected the flow of traffic at the application layer, it did not affect the flow of packets at the network layer.

Figure 4.8 depicts the impact of the DoS attack on the victim's web server. The website took a long time to load.



Figure 4.8: Screenshot of Effect of DoS attack on Webserver

**CHAPTER FIVE**

**DISCUSSION and CONCLUSION.**

**5.1 Discussion**

People today place a premium on networking security in the technological world, especially when it comes to the Internet of Things (IoT). Essentially, the network enables users to easily communicate and access resources. Because all users rely on the internet to serve as a global information source, its availability is critical. The main goal of DoS attacks is to make services unavailable to users. When an attacker compromises essential services such as emails, websites, and other online interactions, users are initially denied access when a machine or the entire network connection is completely compromised.

Attackers can increase the level of DoS by carrying out these attacks in a distributed fashion, which is known as Distributed Denial of Service (DDoS). A DDoS attack involves a large number of compromised machines launching coordinated attacks against a single victim. An attacker conducts a different type of attack in which multiple UDP and TCP packets of varying sizes are sent to the victim's machine at the same time. As a result, the tool successfully monitors and captures an attacker's network packets. UDP and TCP flood attacks exhaust server resources much faster because they consume all network bandwidth on the server's network link by denying access to legitimate users.

General security practices for the IoT include changing default usernames and passwords, using complex passwords to reduce the risk of brute force and rainbow attacks, and so on. Auditing, defining user privileges, whitelisting, using Intrusion Detection Systems (IDS) and Intrusion Prevention Systems (IPS) to analyze network abnormalities, tuning true false alarms to reduce the risk of intrusions, and raising user awareness of social engineering are all part of the process.

**5.2 Conclusion:**

Unfortunately, a one-size-fits-all approach is ineffective for deterring and preventing DDoS attacks, which remain the most common cybercrime threat. In addition to legal and legislative safeguards, technology plays an important role in preventing such threats. This can be accomplished by optimising protection measures such as prevention, detection, correction, and reaction. Nonetheless, cloud computing is regarded as the most recent computing paradigm, providing a wide range of flexible and consistent services. It is also a rapidly evolving technology that is widely recognized as the computing model throughout the world due to features such as large storage space, rapid deployment and distribution, cost efficiency, and the ability to access the system from anywhere and at any time.

The growing use of cloud services by government agencies poses a new threat to e-government and e-governance applications. The harsh reality is that in order to reap the benefits of this technology, it must be visible on unprotected internet networks. With this reality, the cloud is vulnerable to data confidentiality threats, and unauthorised attacks would constantly seek to exploit the cloud's services. Furthermore, the availability of botnets and virtualization in cloud systems poses a threat to the efficacy of DDoS attacks. The consequences of government services and resources not being available in the cloud are disastrous, and this can result in a partial or even total failure to deliver the required service.

This study was able to investigate how DDoS attacks affect cloud-hosted IoT systems. This project examines the effects of a DDoS attack on a web-server that hosts IoT. Due to the risk of being arrested if we attempted to perform a DoS attack on an AWS gateway endpoint, we channelled our attack to the network layer, as the systems are hosted on my private home network. *We discovered that a DDoS attack has a significant impact on the entire internet community, including cloud computing and other distributed systems, after analysing and surveying several DDoS attack*s.

As cloud computing evolves, DDoS attacks become more sophisticated, to the point where they can overwhelm a cloud provider. It is confirmed that DDoS will not go away with high availability, resilient control infrastructure, and low cost. As DDoS attacks become more common in all emerging technologies, we can expect an increase in vulnerabilities and security measures in the future.

On the other hand, adopting such cutting-edge technology as quantum computing, which is one of a handful of next-generation computing power solutions, could be used to solve cyber security issues. Such as searching for specific patterns in a big data repository for intrusion detection and more sophisticated forms of parallel computing that can prevent or mitigate potentially any sophisticated type of DDoS attack.

**5.3 DDoS Protection Techniques in the Cloud.**

1. Reduce Attack Surface:

One of the first techniques for mitigating DDoS attacks is to reduce the surface area that can be attacked, limiting attackers' options and allowing you to build protections in a single location. We want to make sure that we don't expose our application or resources to ports, protocols, or applications that aren't expecting any communication from them. As a result, the potential points of attack are reduced, allowing us to focus our mitigation efforts. You can do this in some cases by placing your computation resources behind Content Distribution Networks (CDNs) or Load Balancers and restricting direct Internet traffic to specific parts of your infrastructure, such as your database servers.

1. Plan for Scale:

The two most important considerations for mitigating large-scale volumetric DDoS attacks are bandwidth (or transit) capacity and server capacity to absorb and mitigate attacks.

\* Transit Capacity: When designing your applications, ensure that your hosting provider provides ample redundant Internet connectivity that allows you to handle high traffic volumes. Because the ultimate goal of DDoS attacks is to disrupt the availability of your resources/applications, you should locate them not only near your end users but also near large Internet exchanges, allowing your users easy access to your application even during high volumes of traffic. Web applications can also go a step further by utilising Content Distribution Networks (CDNs) and smart DNS resolution services, which provide an additional layer of network infrastructure for serving content and resolving DNS queries from locations that are frequently closer to your end users.

\* Server Capacity: Most DDoS attacks are volumetric, consuming a large amount of resources; therefore, it is critical that you can quickly scale up or down on your computation resources. This can be accomplished by running on larger computation resources or those with features such as more extensive network interfaces or enhanced networking that support higher volumes. Additionally, load balancers are commonly used to continuously monitor and shift loads between resources to avoid overloading any one resource.

1. Know what is Normal and Abnormal Traffic:

When we detect high levels of traffic hitting a host, our baseline is to accept only as much traffic as our host can handle without affecting availability. This is known as rate limiting. More advanced protection techniques can go a step further and intelligently accept only legitimate traffic by analyzing individual packets. To do so, you must first understand the characteristics of good traffic that the target typically receives and then compare each packet to this baseline.

1. Deploy Firewalls for Sophisticated Application attacks:

Using a Web Application Firewall (WAF) to protect against attacks such as SQL injection or cross-site request forgery that attempt to exploit a vulnerability in your application is a good practise. Furthermore, due to the unique nature of these attacks, you should be able to easily create customised mitigations against illegitimate requests that may disguise as good traffic or originate from bad IPs, unexpected geographies, and so on. It may also be useful in mitigating attacks because they have experienced support to study traffic patterns and create customised protections.

**REFERENCES**

Alaba, F.A., Othman, M., Hashem, I.A.T. and Alotaibi, F., 2017. Internet of Things security: A survey. *Journal of Network and Computer Applications*, *88*, pp.10-28.

Ali, J., 2018. *Development of an extended input-output model for evaluating the impacts of information technology disruptions on interdependent economic sectors* (Doctoral dissertation, The George Washington University).

Celosia, G., 2020. *Privacy challenges in wireless communications of the Internet of Things* (Doctoral dissertation, Université de Lyon).

1. Lohachab and B. Karambir, “Critical Analysis of DDoS—An Emerging Security Threat over IoT Networks,” *J. Commun. Inf. Networks*, vol. 3, no. 3, pp. 57–78, 2018, doi: 10.1007/s41650-018-0022-5.
2. Roohi, M. Adeel, and M. Ali Shah, “DDoS in IoT: A Roadmap Towards Security & Countermeasures,” in *DDoS in IoT: A Roadmap Towards Security & Countermeasures*, 2019, no. September, pp. 5–7.
3. Deep, X. Zheng, A. Jolfaei, D. Yu, P. Ostovari, and A. K. Bashir, “A survey of security and privacy issues in the Internet of Things from the layered context,” *arXiv*, no. March, 2019.
4. Vishwakarma and A. K. Jain, “A survey of DDoS attacking techniques and defence mechanisms in the IoT network,” *Telecommun. Syst.*, vol. 73, no. 1, pp. 3–25, 2020, doi: 10.1007/s11235-019-00599-z.
5. Dineva and T. Atanasova, “Security in IoT systems,” *Int. Multidiscip. Sci. GeoConference Surv. Geol. Min. Ecol. Manag. SGEM*, vol. 19, no. 2.1, pp. 569–577, 2019, doi: 10.5593/sgem2019/2.1/s07.075.
6. U.Farooq, M. Waseem, A. Khairi, and S. Mazhar, “A Critical Analysis on the Security Concerns of Internet of Things (IoT),” *Int. J. Comput. Appl.*, vol. 111, no. 7, pp. 1–6, 2015, doi: 10.5120/19547-1280.

1. Ning and H. Liu, “Cyber-Physical-Social Based Security Architecture for Future Internet of Things,” *Adv. Internet Things*, vol. 02, no. 01, pp. 1–7, 2012, doi: 10.4236/ait.2012.21001.
2. Kumari, M. K. Khan, and M. Atiquzzaman, “User authentication schemes for wireless sensor networks: A review,” *Ad Hoc Networks*, vol. 27, no. November, pp. 159–194, 2015, doi: 10.1016/j.adhoc.2014.11.018.

J. Y. Khan, “Introduction to IoT,” *Internet of Things (IoT)*, no. January 2019, pp. 1–24, 2019, doi: 10.1201/9780429399084-1.

H. Nemati, *Information Security and Ethics: Concepts, Methodologies, Tools, and Applications*, New York: IGI Global, 2008, Ch.1.1.

1. Mell and T. Grance. “The NIST Definition of Cloud Computing”. *National Institute of Standards and Technology (NIST) Special Publication*. Gaithersburg, MD. https://csrc.nist.gov/publications/detail/sp/800-145/final, 2011.
2. Y. Oktoberry, “The Role of The Law in Combating DDoS Attacks against e- Government A Comparative Analysis of The Substantive detection against network mimicking DDoS attacks”. *Communications Letters, IEEE*, 13(9): 717 719, 2009.

Y. Mehmood, F. Ahmad, I. Yaqoob, A. Adnane, M. Imran, and S. Guizani, “Internet-of-Things-Based Smart Cities: Recent Advances and Challenges,” IEEE Communications Magazine, vol. 55, no. 9,

pp. 16–24, 2017.

A. Bijalwan, M. Wazid, E. S. Pilli, and R. C. Joshi, “Forensics of Random-UDP Flooding Attacks,” *J. Networks*, vol. 10, no. 5, (2015).

K. Ahmad, S. Vivekananda, and U. Pradesh, “Classification of Internet Security Attacks Classification of Internet Security Attacks,” *Researchgate*, no. October, pp. 1–4, (2015).

El-Sofany, H. and Abou El-Seoud, S., 2019. A Novel Model for Securing Mobile-based Systems against DDoS Attacks in Cloud Computing Environment.

Fujino, N., Ogawa, K. and Minowa, M., 2016. Wireless network technologies to support the age of IoT. *Fujitsu Sci. Tech. J*, *52*(4), pp.68-76.

Kaur, C., 2020. Cloud computing and the internet of things (IoT). *International Journal of Scientific Research in Science, Engineering and Technology*, *7*(1), pp.19-22.

Kimani, K., Oduol, V. and Langat, K., 2019. Cyber security challenges for IoT-based smart grid networks. *International Journal of Critical Infrastructure Protection*, *25*, pp.36-49.

Munirathinam, S., 2020. Industry 4.0: Industrial Internet of things (IIOT). In *Advances in Computers* (Vol. 117, No. 1, pp. 129-164). Elsevier.

Nagy, J., Oláh, J., Erdei, E., Máté, D. and Popp, J., 2018. The role and impact of Industry 4.0 and the internet of things on the business strategy of the value chain—the case of Hungary. *Sustainability*, *10*(10), p.3491.

Singh, R. and Sharma, T.P., 2019. Present Status of Distributed Denial of service (DDoS) attacks in the internet world. *International Journal of Mathematical, Engineering and Management Sciences*, *4*(4), p.1008.

Somani, G., Gaur, M.S., Sanghi, D., Conti, M., and Buyya, R., 2017. DDoS attacks in cloud computing: Issues, taxonomy, and future directions. *Computer Communications*, *107*, pp.30-48.

Vignau, B., Khoury, R., Hallé, S. and Hamou-Lhadj, A., 2021. The evolution of IoT Malwares, from 2008 to 2019: Survey, taxonomy, process simulator and perspectives. *Journal of Systems Architecture*, *116*, p.102143.

Wang, L., Ranjan, R., Chen, J. and Benatallah, B. eds., 2017. *Cloud computing: methodology, systems, and applications*. CRC Press.

Wanjun, L.I.U., 2016. The Visualization Analysis of Cloud Computing Research Based on Mapping Knowledge. *Management Science and Engineering*, *10*(4), pp.41-46.

Amazon Web Services, Inc. 2022. What is a DDOS Attack & How to Protect Your Site Against One? [online] Available at: <https://aws.amazon.com/shield/ddos-attack-protection/> [Accessed 6 May 2022].

D. Peraković, M. Periša and I. Cvitić, “Analysis of the IoT Impact on Volume of DDoS Attacks”, PosTel 2015, Belgrade, December, 2015

T. Khalil, “IoT Security against DDoS Attacks Using Machine Learning Algorithms”, International Journal of Scientific and Research Publications, vol. 7, June, 2017

D. Mukhopadhyay, B. Oh, S. Shim and Y. Kim, “A Study on Recent Approaches in Handling DDoS Attacks”, International Journal of Computer Science Trends and Technology, vol. 3, 2015

Administrator (2018). *Raspberry Pi Ultrasonic Sensor HC-SR04 Interface Tutorial*. [online] Electronics Hub. Available at: https://www.electronicshub.org/raspberry-pi-ultrasonic-sensor-interface-tutorial/ [Accessed 22 Jul. 2022].

docs.aws.amazon.com. (n.d.). *Connect a Raspberry Pi or another device - AWS IoT Core*. [online] Available at: https://docs.aws.amazon.com/iot/latest/developerguide/connecting-to-existing-device.html.

docs.aws.amazon.com. (n.d.). *Tutorial: Connecting a device to AWS IoT Core by using the AWS IoT Device SDK - AWS IoT Core*. [online] Available at: https://docs.aws.amazon.com/iot/latest/developerguide/sdk-tutorials.html#sdk-tutorials-experiment [Accessed 22 Jul. 2022].

Ferranti, M. (2018). *What is Nmap? Why you need this network mapper*. [online] Network World. Available at: https://www.networkworld.com/article/3296740/what-is-nmap-why-you-need-this-network-mapper.html.

ferrisbuller (2022). *16 Best DDOS Attack Tools in 2022*. [online] Wallarm. Available at: https://lab.wallarm.com/16-best-ddos-attack-tools-in-2022/ [Accessed 22 Jul. 2022].

GeeksforGeeks. (2021). *Challenges in Internet of things (IoT)*. [online] Available at: https://www.geeksforgeeks.org/challenges-in-internet-of-things-iot/.

Miyachi, C. (2018). What is ‘Cloud’? It is time to update the NIST definition?. *IEEE Cloud Computing*, 5(3), pp.6–11. doi:10.1109/mcc.2018.032591611.

Page, V. (2019). *What Is Amazon Web Services and Why Is It so Successful?* [online] Investopedia. Available at: https://www.investopedia.com/articles/investing/011316/what-amazon-web-services-and-why-it-so-successful.asp.

Team, T. (2021). *Architecture of IoT*. [online] TechVidvan. Available at: https://techvidvan.com/tutorials/architecture-of-iot/ [Accessed 22 Jul. 2022].

Techvidvan.com. (2021). [online] Available at: https://techvidvan.com/tutorials/wp-content/uploads/sites/2/2021/10/6lowpan-architecture.webp [Accessed 22 Jul. 2022].

Wireshark Foundation (2016). *Wireshark*. [online] Wireshark.org. Available at: https://www.wireshark.org/.

www.codemag.com. (n.d.). *Introduction to IoT Using the Raspberry Pi*. [online] Available at: https://www.codemag.com/article/1607071/Introduction-to-IoT-Using-the-Raspberry-Pi#:~:text=Raspberry%20Pi%20is%20a%20low [Accessed 22 Jul. 2022].

1. [online] Available at: <https://www.researchgate.net/publication/339577222\_A\_Low-Cost\_Distributed\_Denial-of-Service\_Attack\_Architecture/figures?lo=1&utm\_source=google&utm\_medium=organic> [Accessed 6 May 2022].