DETECTION AND PREVENTION SCHEMES FOR DDOS, ARP SPOOFING, AND IP FRAGMENTATION ATTACKS IN SMART FACTORY

CHAI TZE UEI

MASTER OF SCIENCE (COMPUTER SCIENCE)

FACULTY OF INFORMATION AND
COMMUNICATION TECHNOLOGY(FICT)
UNIVERSITI TUNKU ABDUL RAHMAN
SEPTEMBER 2023

DETECTION AND PREVENTION SCHEMES FOR DDOS, ARP SPOOFING, AND IP FRAGMENTATION ATTACKS IN SMART FACTORY

By

CHAI TZE UEI

A thesis submitted to the Department of Computer and Communication Technology,
Faculty of Information and Communication Technology,
Universiti Tunku Abdul Rahman,
in partial fulfillment of the requirements for the degree of
Master of Science (Computer Science)
September 2023

ABSTRACT

DETECTION AND PREVENTION SCHEMES FOR DDOS, ARP SPOOFING, AND IP FRAGMENTATION ATTACKS IN SMART FACTORY

Chai Tze Uei

Industry Revolution 4.0 allows Internet of Things (IoT) resource constrained devices to be integrated into the technologies and systems to develop intelligent solutions that leverage the value of data and deliver insight. The network configuration can be complex due to the dynamic IoT environments, such as numerous diverse devices that interact to deliver an autonomous function. In this situation, the environments can produce a significant amount of data and expose vulnerabilities in the communication protocols. Once an attacker breaks into the network, the whole network infrastructure can be broken down.

Therefore, this research selects three potential attacks with an evaluation of the protections, namely 1) Distributed Denial of Service (DDoS), 2) Address Resolution Protocol (ARP) spoofing, and 3) Internet Protocol (IP) Fragmentation attacks. In the DDoS protection, the F1-score (a.k.a. F-score), accuracy, precision, and recall of the four-feature Random Forest with Principal Component Analysis (RFPCA) model are 95.65%, 97%, 97.06%, and 94.29% respectively. In the ARP spoofing, a batch processing method adopts the entropy calculated in the 20s of time window with sensitivity to network abnormalities detection of various ARP spoofing scenarios involving victims' traffic. The detected attacker's Media Access Control (MAC) address is inserted in the block list to filter malicious traffic. The proposed protection in the Internet Protocol (IP) fragmentation attack is to implement one-time code (OTC) and timestamp fields in the packet header. The simulation shows that the method can detect 160 fake fragments from attackers in 2040 fragments.

APPROVAL SHEET

This thesis entitled "**DETECTION AND PREVENTION SCHEMES FOR DDOS, ARP SPOOFING, AND IP FRAGMENTATION ATTACKS IN SMART FACTORY"** was prepared by CHAI TZE UEI and submitted as partial fulfillment of the requirements for the degree of Master of Science (Computer Science) at Universiti Tunku Abdul Rahman.

Approved by:

Goh.

(Assoc. Prof. Dr. Goh Hock Guan)
Date:…27/9/2023……..
Supervisor
Department of Computer and Communication Technology
Faculty of Information and Technology
Universiti Tunku Abdul Rahman

(Prof. Dr. Liew Soung-Yue)
Date:…27/9/2023……..
Co-supervisor
Department of Computer and Communication Technology
Faculty of Information and Technology
Universiti Tunku Abdul Rahman

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

UNIVERSITI TUNKU ABDUL RAHMAN

Date: 27-09-2023

SUBMISSION OF THESIS

It is hereby certified that _*CHAI TZE UEI*_ (ID No: *20ACM00941*) has completed this final year thesis entitled "*DETECTION AND PREVENTION SCHEMES FOR DDOS, ARP SPOOFING, AND IP FRAGMENTATION ATTACKS IN SMART FACTORY*" under the supervision of Dr. Goh Hock Guan (Supervisor) from the Department of Computer and Communication Technology, Faculty of Information and Communication Technology, and Dr. Liew Suong-Yue (Co-Supervisor) from the Department of Computer and Communication Technology, Faculty of Information and Communication Technology.

I understand that University will upload softcopy of my thesis in pdf format into UTAR Institutional Repository, which may be made accessible to UTAR community and public.

Yours truly,

(*CHAI TZE UEI*)

DECLARATION

I hereby declare that the thesis is based on my original work except for quotations and citations which have been duly acknowledged. I also declare that it has not been previously or concurrently submitted for any other degree at UTAR or other institutions.

Name ____Chai Tze Uei_____________

Date _______27-09-2023_____________

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVATION

5G	5th generation
6LoWPAN	IPv6 over Low Power Wireless Personal Area Network
AI	Artificial Intelligence
AP	Access point
API	Application Programming Interface
ARP	Address Resolution Protocol
BDA	Big Data Analytics
BLE	Bluetooth Low Energy
C&C	Command and Control
CPS	Cyber-Physical System
CSP	Cloud Service Provider
DDoS	Distributed Denial of Service
DF	Don’t Fragment
DNS	Domain Name Systems
DoS	Denial of Service
DSCP	Differentiated Service Code Point
ECN	Explicit Congestion Notification
ERP	Enterprise Resource Planning
ETL	Extract-Transform-Load
HART	Highway Addressable Remote Transducer
HMI	Human machine interface
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
I/O	Input/Output
ICMP	Internet Control Message Protocol
ICS	Industrial Control System
IEEE	Institute of Electrical and Electronics Engineers
IIoT	Industrial Internet of Things
IoT	Internet of Things
IP	Internet Protocol
IP-ID	IP Identifier
IP-MAC	Internet Protocol- Media Access Control
IPv4	Internet Protocol version 4
IPv6	Internet Protocol version 6
IT	Information Technology
KNN	K-Nearest Neighbors
LAN	Local Area Network

LoRaWAN	Long Range Wide Area Network
LPWAN	Low Power Wireless Area Networks
MAC	Media Access Control
MEC	Multi-Access Edge Computing
MES	Manufacturing Execution Support
MF	More Fragment
MITM	Man-In-The-Middle
MQTT	Message Queuing Telemetry Transport
MTU	Maximum Transmission Unit
NIDS	Network intrusion detection systems
OPC UA	Open Platform Communication Unified Architecture
OS	Operating System
OSI	Open Systems Interconnection
OT	Operating Technology
OTA	Over-the-air
OTC	One-time code
OVH	On Vous Heberge
OWASP	Open Web Application Security Project
PC	Principal Component
PCA	Principal Component Analysis
PLC	Programmable Logic Controller
PMTUD	Path MTU Discovery
RAM	Random Access Memory
RF	Random Forest
RFID	Radio Frequency Identification
RFPCA	Random Forest with Principal Component Analysis
RTOS	Real-Time Operating System
SME	Small and Medium-sized enterprises
SOA	Service-Oriented Architecture
SOAP	Simple Object Access Protocol
SQL	Structured Query Language
SSH	Secure Socket Shell
SVM	Support Vector Machine
TCP SYN	Transmission Control Protocol Synchronized
TCP/IP	Transmission Control Protocol/Internet Protocol
TLS	Transport Layer security
TOS	Type of service
TSN	Time-sensitive networking
UDP	User Datagram Protocol
USB	Universal Serial Bus
VOIP	Voice over IP
WirelessHART	Wireless Highway Addressable Remote Transducer

WSN	Wireless Sensor Network
WWW / W3	World Wide Web
XSS	Cross-site scripting

CHAPTER 1

INTRODUCTION

1.1 Motivation

The Industry 4.0 revolution is leading towards a digitisation of the factory into a cyberspace. The efficiency and effectiveness of intelligent systems, machines, and algorithms are essential to support production operations running 24/7. As organisations continue investing in technology, the number of devices connected through the network increases and accelerates the convergence of Information technology (IT) and Operating Technology (OT) [1]. The security triads which enforce the basic requirements in the IT and OT environments are different [2]. In the context of Operational Technology (OT), a combination of hardware and software is employed to observe the environment and utilise sensor data to make necessary adjustments in response to changes. The Internet of Things (IoT) considered a resource constraint with limited computing power widely adopted in the environment, provides data to the Cloud to support analytics and prediction [3]. IoT needs firmware [4] to perform a specific function to meet the real-time processing deadline and fit into the manufacturing use case scenarios with limited capability in terms of Random Access Memory (RAM) and processing power [5]. The current machines use the legacy system to run the operation with minimal downtime are lack security considerations [6][7]. The system is connected locally, linked to the Cloud through the Internet, and exposed to various potential threats.

IoT is increasing while functional areas are widely adopted [8]. Research shows that organisations still need to focus on cybersecurity, and investments still need to be made to protect the functional area. The two main objectives for the industry to move into the Smart Factory are business objectives related to earning and security concerns which express a need for more detail and confidence in cybersecurity protection.

In a Wireless Sensor Network (WSN), the sensor nodes are distributed and interact with each other to collect, process, and deliver the sensing details of the physical environment to the destination node. Data are aggregated at the node to reduce network congestion, where Internet access is an option through the gateway. For IoT, data are processed and transformed into information that supports decentralised decision making [9]. Internet is mandatory in this case. IoT and Industrial Internet of Things (IIoT) are similar, where IIoT is considered a subset of IoT. The IIoT is used in industrial processes, such as manufacturing, production, and supply chain, which monitor the processes. This device is required to handle the operation in critical infrastructure with complex instruments focused on reliability more than data security [10]. Typical IoT use cases are consumer or office.

Adversaries no longer depend on individual actions to trigger the attack. Wide ranges of vulnerabilities in IoT [11] can be utilised to penetrate the network, inject the smart devices with malware to act as a botnet and launch the attack through remote commands with an Internet connection [12]. The software patch intending to recover vulnerable devices is faced with the complexity in the patching process and trade off. In the firmware update, it is not only to consider the security need but also availability, as some patches may require a reboot, leading to a production outage. In this situation, the technician may delay or skip the patching, leaving the vulnerable devices to remain as the target [13]. All the devices connected and extracting information for further processing are exposed and vulnerable [14]. For example, a smartphone or handheld device is connected to IoT to retrieve production status. IoT usually connects to just a few applications through the Internet. In addition, mobility increases the possibility of the security breach spreading through the networks. Also, a technician or workers lack knowledge of the devices connected in the environment simultaneously, and the security vulnerability of the IoT allows grant access remotely or physically as simple as using a Universal Serial Bus (USB) plug-in [15]. The vulnerability allows attackers to break into the local network, intercept communication between legitimate nodes, such as the application server, and download the malware into the devices [16].

Incidents show the vulnerability of critical infrastructure in various industries, with manufacturing as one of the significant targets of attackers [17]. The attacker's interest is more than the data, from revealing sensitive information to preparing for a severe attack, such as causing physical damage and shutting down the operations [18]. The IoT, the extension of the Industrial Control System (ICS), further exposes internal network vulnerability [19]. As in earlier, those critical infrastructures operated in isolation. The highly interconnected IoT devices lead to more complex and reliant on the network. Security breach from the IoT leads to the penetration of the network and disabling of the devices' functions, such as smart lock [20], with failure of firmware update causing the door to lock [21]. The thermostat, which has been hacked and fails to provide correct readings, increases the temperature in the environment [22]. The lessons have shown security in automation systems has become one of the weakest links [23]. Skilful attackers actively develop capabilities and deploy sophisticated attacks [24]. Therefore, the network must protect itself from attackers and recover at the earliest, if not immediately.

The study of security in Industry 4.0 gives the awareness of the potential environmental threats, security issues, and challenges led by emerging technology and highly interconnected network and system. The assets, such as data, networks, and systems, must adhere to the guidelines and standards to provide sufficient protection.

1.2 Problem Statements and Research Questions

The fourth industrial revolution heavily relies on technology to operate. As a result, there has been a surge in the deployment of smart devices, known as the IoT, in the interconnected network of smart manufacturing. This integration of systems has enabled both horizontal and vertical integration on a large scale. IoT networks are highly interconnected in a heterogeneous environment and generate data in different formats and sometimes with missing values. The devices use Transmission Control Protocol/Internet Protocol (TCP/IP) and the custom design protocol in the connection and communication. There are vulnerabilities exhibited in the layers of the TCP/IP model, such as the access control in the network layer. IoT lacks resources and computing power for the execution of complex algorithms to implement protection such as virus protection. Lack of advanced protection can lead to security issues such as the attack on the devices connected to the Internet, where IoT is a source of data collection, which grows exponentially and consequently becomes the source of information. The most significant concern in smart manufacturing is caused by a lack of data or false data and availability in the operation.

Many of the complex IoT are already in operation and lack security measures. Distributed Denial of Service (DDoS) is much more dangerous for large scale appliances in safety-critical systems, and the timing of the attack can lead to disastrous events. IoT is easily infected by malware. The infection can quickly spread to a large number and create zombies controlled by the Command and Control (C&C) server. These IoT are unlike conventional Internet devices in which the attack can be resolved by turning on and off or restarting the devices. The highly intensive attack requires many infected or compromised devices that can turn into a botnet and are synchronised to launch the attack simultaneously from many networks, which is difficult to defend against. In the attack, data aggregate in the gateway can create tremendous data that floods and brings down critical infrastructure.

Address Resolution Protocol (ARP) is a protocol that adopts a simple form by broadcasting an ARP message to acquire mapping for the dynamic Internet Protocol (IP) address to the physical address between the communication hosts or machines. The physical address is a Media Access Control (MAC) address. The ARP is a TCP/IP protocol that works between layers 2 and 3 of the Open

System Interconnection (OSI) model. The MAC address exists in layer 2, a data link layer and an IP address in layer 3, a network layer. This protocol has been proven efficient to work on IP networks and is widely utilised in IoT systems. However, it has come with security risks and being the target of attack due to the vulnerability of the ARP protocol, which is a lack of a proper authentication process. When a malicious node is present in the local network, it can easily acquire the address mapping of the local target host and intercept the communication. It becomes a security issue in the IoT, where the plain text in the transmission between nodes is due to a lack of security implementation in the first place. This situation can lead to exposure to sensitive details such as passwords. ARP spoofing can facilitate other malicious attacks, such as Man-In-The-Middle (MITM) attack, session hijacking or denial-of-service attack. ARP follows the principle of adopting the last ARP message to update the ARP cache. Thus, an attack can continue to send a forged ARP packet to the target and hide its activity by sending a high volume of the message.

In most cases, the IoT handles only a small amount of the data, such as sending or sharing data and receiving instructions. The process allows IoT to communicate efficiently. However, there is a situation where the IoT needs to receive more data size messages, such as firmware patches. In typical transmission, the data travels from different networks with different Maximum Transmission Unit (MTU) sizes and causes fragmentation, especially when a large amount of data is involved. IP fragmentation can happen in the Smart Factory and IoT. The packet header identifier is an essential detail in a fragment that the attacker can guess. Meanwhile, avoiding fragmentation vulnerabilities by simply preventing fragmentation can cost more transmission sessions. The TCP protocol does not prevent an attack from adversaries that send a spoofed Internet Control Message Protocol (ICMP), and the source can be tricked into fragmenting TCP segments. When the packet is fragmented, and header identifiers exist in many transmissions, the attacker can carefully craft a fragment to attack, such as causing misassociation and leading to unsuccessful firmware updates or installing malicious code.

Cybersecurity threats come from internal and external sources, and the attacks can be stealthy. These attacks are a common yet severe problem. In Cyber-Physical Systems (CPSs), the intelligent systems control the machines through commands that work autonomously and link the physical components to cyberspace [25]. When the smart manufacturing system is attacked, production stops, and infrastructure breaks down. Thus, high detection and reliable protection schemes are required to protect the network and raise a timely warning when the attack occurs to prevent massive damage to the critical infrastructure and organization lost.

Research questions:

1. Based on the selected attack scenarios, how can DDoS, ARP spoofing, and IP fragmentation attacks be identified in Smart Factory?
2. When the attacks occur, can the protections be implemented at the point of the network and filter the attackers from further destroying the operation in the Smart Factory?

3. How to evaluate and verify the effectiveness of the protection schemes based on the defined attack scenarios?

1.3 Objectives

The objectives of the research are :

1. To present the detection methods adopted in the selected attack models.
2. To implement the protection and evaluate the network before and after the recovery of the network after the protection is executed.
3. To propose the protection scheme for the identified attacks based on the result of detection performance.

In order to achieve the research objective, the network simulation method is crucial to study the attacks in the Smart Factory, each present in the network or infrastructure in the environment. OMNeT++ software simulation tool is a discrete event simulation that supports a hierarchical model with a graphical editor with GUI-based execution. Together with the INET framework, the simulator provides protocols, agents and models to work with communicating networks, passing the message and modelling their behaviours and interactions in normal conditions and when the attack occurs in Smart Factory. These mechanisms and functions suit the objective of the simulation to model the problem stated in all three attacks in the Smart Factory. The simulation includes modelling the topology of each attack and network activity, generating the network traffic data based on each situation and problems defined and used for data collection and analysis.

1.4 Main Contribution

The contributions from the research are:

1. To show the effect of different detections based on the models and the attack scenarios.
2. To compare and evaluate the protection measures in the network traffic and show the effect of the recovery.
3. To demonstrate attacks that can be protected by choosing the appropriate protection mechanisms and methods.

1.5 Organisation of the Thesis

The organisation of the thesis begins with the introduction in Chapter 1. This chapter explains the problem solved in work. Chapter 2 presents the literature review in the scope of Industry 4.0 evolution, emerging technologies, and issues and challenges in the Smart Factory that attacker launches attacks. In Chapters 3 to 5, each chapter covers individual attacks and presents the details of works with the protection associated with each attack. The sections in these chapters include the detection, verification plan, protection, result, and conclusion remark. Those attacks include DDoS in Chapter 3, ARP spoofing in Chapter 4, and IP fragmentation attack in Chapter 5. Lastly, Chapter 6 concludes the research work.

CHAPTER 2

LITERATURE REVIEW

2.1 Security in Industry Revolution 4.0

The first industrial revolution started through mechanisation at the end of the eighteenth century. In this century, the industry mainly used the power of water and steam in production. The second industrial revolution occurred during the late nineteenth century with the invention and use of electricity, gas, and oil as power sources. The third industrial revolution was caused by the invention of computers. The industry used IT, where the two significant inventions, the Programmable Logic Controllers (PLCs), and robots, ran the production, leading to the era of high-level automation. The fourth industrial revolution computerises production and develops the world's virtual reality [27].

Typical factories consist of IT, which handles the office's computer system that communicates in an IP-based network, and OT, which handles the operation of the machines in the plants with various proprietary protocols. The two systems were segregated and protected by their network practice and procedure in previous years. In Industry 4.0, the two systems are increasingly merging [28], and IT technologies threats can bring to the OT environment [29].

Implementing IT systems focuses on meeting integrity, confidentiality, and availability (CIA). OT follows the priority of availability, integrity, and confidentiality (AIC) [30]. The components run 24/7. However, there are differences between the IT and OT systems. In an IT network, the architecture supports a single component's failure and ensures availability across different sections with technology to speed up the recovery process. IT network adopts standard communication protocols on the Internet, such as Hypertext Transfer Protocol (HTTP) and Secure Socket Shell (SSH). In OT, the system and software controlling the physical operations, that single point of device failure may cause the operation to pause or stop. These devices communicate with protocols different from IT, such as EtherNet/IP and Profinet. Reliability is vital that the machinery is designed with safety and reliability as a significant consideration. In the application of OT, Time-sensitive networking (TSN) is deployed in the Local Area Network (LAN) [31] to allow horizontal communication from machine to machine and controller to controller that places high performance of network requirements in terms of latency and certainty.

In the OT, finding those machines from a different manufacturer in the machine-to-machine direct communication, such as PLC, to a robotic arm is common. Those components need to support a longer operation lifetime with communication handled in their proprietary protocols, developed by the hardware manufacturer to achieve competitive advantages and solve specific problems in the industry application [32]. The adoption of multiple proprietary protocols that use by the devices shows the lack of interoperability in a situation where multiple vendors' devices are used in operation across the network.

IIoT refers to industrial sensors and instruments connected through a network controlled by an industry computer system in manufacturing. The leading wireless communication protocol in an industrial environment, the Wireless Highway Addressable Remote Transducer (WirelessHART), is capable of monitoring, receiving commands from field devices and acting on the actuators [33]. The WirelessHART is compatible with existing Highway Addressable Remote Transducer (HART) devices and connects to the process control system through Modbus TCP/IP. The field devices sending data transmitted through the gateway supported Open Platform Communication Unified Architecture (OPC UA), a standard with platform independence and interoperability as a goal in the communication. They make those data available to the higher level system through the OPC UA server. In terms of the configuration, by default, all WirelessHART devices use a Join Key to connect to the network and exchange control packets with the network gateway. In this case, using a unique password is not recommended since this password is highly used in all broadcast communications. The devices connected through a gateway to the network become vulnerable when the penetration happens to the ICS network. The situation leads to a severe security breach that exposes the entire network to outsiders once the devices, such as routers or gateway, are compromised.

When moving into Industry 4.0, the connected devices will have access to the Internet. Production robots and actuators receive the instruction and command remotely. An attacker can launch the attack without physically existing in the location. Although the production runs in a closed-loop environment, a single breach in the network or IoT can lead to a security issue and cause massive damage in production [34]. It is no longer possible or sufficient for security personnel to run the security assessment and tackle security breaches.

As for the existing ICS that remains supported by the legacy system, cyber security is sure to be a threat. In such cases, when considering more secure communication, existing customers running on the Modbus network leave them with several considerations in cost-effective measures [35]. Firstly, the Modbus lacks functionality that common support for master-slave architecture. Also, there may be devices in the network that currently in operation for a very long time and need to be updated to support the security in Modbus TCP. They can also adopt OPC UA, which is more secure and provides functionalities that prepare them for Industry 4.0. This protocol allows the support of a services-oriented network. For example, a computer can run on the application or cross-platform Simple Object Access Protocol (SOAP) with Hypertext Transfer Protocol Secure (HTTPS).

With the emergence of technology, the industry adopts resource sharing to improve efficiency. The current intelligent devices in the industry are built with connectivity capability to support various protocols, such as web services running on SOAP and remote access. Nevertheless, due to the lack of an up-to-date Operating System (OS) running the services, the vital assets and critical databases that hold valuable and essential data are easily infiltrated and penetrated by the attacker.

With the intelligent sensors and IoT integrated into the system, these innovative products let manufacturers run the organisation more effectively and efficiently while at the same time maintaining the existing ICS that has been used comes with security risks [36]. First, smart devices have excellent capability and connectivity, allowing data sharing across networks. Without security concerns and measures, these devices monitor the operation of various machines if malfunction and defects are only being identified when cause physical damage, loss of communication, product misconfiguration, or failed devices are lost. Secondly, in the ICS, attention is placed on the machine, sensors, and devices. However, a lack of consideration and attention to the network and infrastructure gives attackers a vast opportunity and interest to target the networks or connected network links to the system. Consequently, the critical infrastructure under attack will lead to considerable losses to the organisation.

Moving into Industry 4.0, the ICS no longer stands for isolated network and system. CPS makes decentralised decisions in the virtual world where data are used in learning algorithms to support intelligent systems to run autonomously. A lack of knowledge or skill to handle the tasks and human error can expose a vulnerability in the IoT network environment [1]. This leave some of the organisation such as Small and Medium-sized enterprises (SME) which lack of resource to implement advanced defence system are vulnerable.

The machine equipped with various sensors, controlled by the instruction sent from the server, is monitoring the condition and sending the data to the Cloud for analysis and evaluation. Cloud storage, CPS and IoT form a system linking the physical component and machine to cyberspace. It collects data from the sensors and actuators used by the intelligent system to adapt to the physical situation and improve resource efficiency [37] with a progressive method for maintenance. Predictive and preventive maintenance are two core components that support the production operation of products running 24/7. Big Data Analytics (BDA) and Cloud predictive and preventive maintenance are designed and implemented in the Smart Factory to reduce the number of failure incidents and improve the reliability of the asset. Before introducing the technology, the factory operator must handle the maintenance task by manually overseeing the activity through a series of tasks, such as calculating the correlation information related to defective products based on the problem defined in the process. While in automated manufacturing, overhead is increased to analyse vast amounts of data collected [38]. Predictive maintenance is time-intensive to implement correctly. On the other hand, it can reduce the maximum downtime, while preventive maintenance is relatively easy to implement and better than reactive maintenance. However, this comes with the risk of possibly damaging the asset. Therefore, Smart Factory adopts promising tools such as machine learning methods that require effective data processing, such as accurate and valuable data that go through the Extract-Transform-Load (ETL) data transmission process.

Automation in Industry 4.0 occurs in the machine and operation in handling the software and firmware that enable proper sensors and IoT functions. In Industry 4.0, firmware updates become an essential aspect of security. The firmware update aims to patch vulnerabilities in the components, bug fixes, and even add features to meet the functional requirements. In the past, the firmware update was handled manually by skilful workers. Large scale IoT deployment with many devices is impractical to handle manually. Those devices can utilise the connected network and automatically install the updated firmware version into the IoT. An automatic update process can avoid time-consuming, costly, and lack of proper procedure and supervision in the updating process. However, OT, the patch often delivers much slower than the IT environment, which often drags more than 60 days. On the World Wide Web (WWW), information can be easily acquired. If the old version of firmware with published vulnerability is not patched in time, this can potentially become the target of attackers. Manually updating can be adopted in malfunctioning devices requiring quick fixes to restore operation.

There are constraints in the low-end devices and PLCs, such as limited processing capabilities and the need to ensure a long operation time. Manufacturers are not considered fundamental protection mechanisms such as software patches and updates through Over-the-air (OTA), and lack of encryption or authentication of the device used in the industrial process. Once broken by attackers, the devices become vulnerable.

In addition to unexpected failure, many control systems and operations cannot be down without impacting production and safety. In those cases, the products produced are more critical than the information being relayed. IT practices to reboot a component would violate the adherence to requirements to meet the ICS's high availability, reliability, and maintainability. There needs to be more log data where there are necessary to program each PLC to save the history of a process. The lack of sufficient log history impacts the troubleshooting and investigation's effectiveness. Another challenge is the fragmentation in the collaborating diverse technologies without systematic recipes for security standards and guidelines, which leads to the sector's practical systems and services security diversities. For example, Message Queuing Telemetry Transport (MQTT), being one of the options in IoT communication, which relies on the developer to implement a Transport Layer Security (TLS) mechanism for security purposes, may be ignored. The production is designed to be a fast reconfiguration and adaptability, producing a customised product and instant data sharing and processing that derive insight and facilitate decentralised decision making in a self-organised manner. The extensive quantity of IoT, a resource constraint with unsubstantial defence systems, uses diversified communication protocols that weaken the ability to protect themselves. The DDoS attack targets critical infrastructure and servers like CPS or Domain Name Systems (DNS) servers. It interrupts the continuity of communication of devices connected to the network and impacts the operation for hours. In this case, the physical component cannot communicate with the server and complete the command for the production process. An ARP spoofing can occur as simply as a USB plug-in method. When ARP spoofing occurs, the IoT interact with an intruder, which can cause unstable or intermittent access to the Internet or steal sensitive data as communication is conducted in plain text. The attacker can maliciously send a fraudulent message to a host and database or a false command to interact with the actuator that misleads with an abnormal operation or action. These can lead to the production of defective products. There are challenges to maintaining an up-to-date physical part of IoT. Some IoT solutions require a complex setup for firmware patches or face difficulty in maintaining a network connection to perform OTA updates. Patches are written for the existing IoT devices, but the patching process can be complex and is ignored or skipped by the user, leaving the vulnerability open for the attacker. IP fragmentation occurs when a large packet is transmitted across the network and finally reassembled into the original datagram at the destination. The attacker sends a fake fragment that cannot be reassembled at the destination. In a transmission such as a firmware update, this not only impacts the update activities and delays the operation but can cause the failure of firmware update, which serves the purpose of a security patch, leaving the IoT vulnerable.

2.2 Cyber Security Technology in the Smart Factory

2.2.1 Cloud Computing

Cloud is the server that is accessed remotely over the Internet. Cloud computing offers storage and processing power that can support the vast demands of various services where the information is accessed remotely in virtual spaces. The high performance programs and services that cannot run on the local computer can now relocate to the Cloud for processing [39]. Cloud and other enabling technologies, such as the IoT and Artificial Intelligence (AI), make the Smart Factory practices fully comprehensive. As for the Cloud to support a wide range of services and functionalities, the resource must group and share across the network.

In a Smart Factory, the Cloud receives data from the IoT through communication, such as MQTT and uses it to train for an enterprise-wide predictive convergence model. While the Cloud has tremendous storage capacity, it is often not cost-effective to handle all IoT data generated in daily activities to support daily operations. Those data require preprocessing, and only the meaningful data are valuable to serve the Cloud applications and services, such as reporting. Cloud needs to select the data processing method effectively [40] and execute complex data analytics algorithms to derive insight. In the implementation, some applications running on the Smart Factory, such as Digital Twin and Enterprise Resource Planning (ERP), utilise huge computing power and storage capability to support application requirements [41]. The data provide seamless factory operations integration and minimise infrastructure administration with a "pay as per go" pricing model. Integrating the IoT into the ERP helps to gain insightful decision making and triggers reactions such as re-order, replenishment and out of stock inventories.

Ontology modelling is used to improve data flow across automation solutions and utilised by applications across the Smart Factory, which different stakeholders may develop. The Service-Oriented Architecture (SOA) is implemented with ontology to model the Smart Factory, automating the data acquisitions and unification that support OT and IT applications like human-machine interaction sessions with any IoT devices and reconfiguration on edge devices. The data are collected from the data stream using a tool such as Apache Kafka, which ensures the data are stored at the buffer and transmitted to the destination whenever the connection is available. When the transmission is complete, these data will only be processed in the Cloud. In other words, the processing logic decouples from the data collection [42] [43].

Cloud processing power and storage capacities are scalable. They can handle the increased number of IoT deployments in the Smart Factory and react quickly to changing requirements through vertical or horizontal scaling. Vertical scaling is achieved by adding powerful computing units and resources to existing servers, while horizontal scaling provides additional servers for requirements. Utilising the huge capacity in the Cloud allows the convergence of models to represent the manufacturing process, optimisation, and continuous improvement of the models, allowing intelligent algorithms to deliver accurate prediction and decision making. The executable models can be downloaded and offloaded processing to the Fog or Edge computing nodes, improving the response time and ensuring operation that is time critical is handled close to the location of the device, such as the production area or LAN. Cloud-based microservices Digital Twin provides integration of Smart Factory functionalities with enhancements inserted at a minimal impact on the entire system. The end user can easily access these functionalities and services. Cloud provides better integration and efficient processes at a lower cost.

Data are stored in distributed locations, while functions are assigned to the resources close to the data for processing. Data transition and service become difficult to move between Clouds, especially for migration [44]. As a result, the dependency of a customer on a single Cloud Service Provider (CSP) becomes a lock-in problem. Application is evolving to integrate with the enterprise's data and infrastructure or from different service providers. Therefore, the Cloud must enhance the ability of different platforms to communicate effectively over secure protocols. The data are the backbone of the Smart Factory operation and various applications and services. Cloud interoperability requires shared processes, Application Programming Interfaces (APIs), and data models over multiple Cloud environments in reliable, performance, and secure manners [45]. The initiative for the CSP is essential to cope with the system's complexities while maintaining and offering the services to the customer by developing a universal API to support the virtualised applications.

BDA uses the data to predict and prevent attacks in the Smart Factory that support customer requests. A multi-tenant Cloud is a software instance running on the same system architect and hardware supporting multiple customers. Although this strategy allows better utilisation of the Cloud resources with the ease of setting up the platform and low cost, it has drawbacks such as limited management and customisation. It is easier to serve as a target for attackers due to multiple Access points (APs) to expose system vulnerabilities [46]. Among the security threats are Cross-site scripting (XSS), Structured Query Language (SQL) injection attacks, and security misconfiguration caused by avoiding updating and changing the default password. On the other hand, a single-tenant Cloud runs on dedicated infrastructure and provides enhanced security and reliability of performance for an application at a higher cost.

2.2.2 Fog Computing

Fog computing is a subdivision of Cloud computing and decentralised computing structure that uses edge devices to perform computation locally while maintaining connectivity with the Cloud [47]. Fog inherits functionalities and offloads the tasks from the Cloud. With virtualisation, services and applications are distributed closer to where data are produced [48]. Fog computing is formed from a group of Fog nodes which can be a router, network switch, local server, and the gateway connected to the edge network and devices. This layer is referring the LAN inside the factory with a database and server that offer limited computing, storage, and networking resources. Compared to the Cloud, Fog computing provides better coverage of the application in the area with real-time or close to real-time communication and handles critical infrastructure and system requests when there are many connected devices to the network [49]. In addition, Fog computing helps to process the data instantly or in batch processing based on the request requirements. This ability enhances the process of interest data and sends it to the Cloud, which can ease bandwidth. Also, the CPS application can support intelligent algorithms and machines simultaneously while offering fault tolerance and reconfiguring the system. Machines and devices such as PLC can communicate the instruction in close to real-time. When implementing IoT architectures, energy consumption should be considered as a battery powers the nodes. Examples of applications are energy or power consumption and scheduling management. Other applications and services under support are inventory management and intelligent maintenance management.

Fog computing can provide functionalities locally that ensure critical functions are still operated in the local network when the Internet connection is unstable. The process in Fog computing is where all Fog nodes form the network, which facilitates the resource sharing of manufacturing equipment and tools. As the local network maintains an Internet connection to the Cloud, Fog computing is exposed to various security threats and vulnerabilities. An Internet connection allows attackers to take vulnerabilities in the system, networks, or local devices and penetrate the network remotely. For example, the compromised node exists in the local network. The attacker can spoof the gateway, sending invalid data or instructions to the Manufacturing Execution Support (MES), ERP, or Digital Twin systems.

With computing and shared resource, a hypervisor can pose a threat to memory overload and causes vulnerabilities in managing resources. Fog platform over web services prone to attackers. For Example, SQL code injection, insecure API, or XSS to target other applications.

Dynamic provision of the computing resources is necessary to adapt to the changes in service load where unexpected service requests surge in complex IoT environments that support multiple applications and requests [50]. Correctly anticipating the request load and adequately handling the source in the virtualised environment can help to cater to threats that target resource depletion in the networks.

2.2.3 Edge Computing

Edge computing is located close to physical areas such as shop floor, production, or manufacturing. These intelligent edge devices or IoT integrate with CPSs simultaneously, running the services and operations closest to the data sources, and users enable processing at greater speeds. An IoT platform that manages the connectivity of devices to store, extract and analyse large volumes of data through edge analytics in real-time enables a level of automation of processing the transaction [51]. Some edge devices are robot arms, IoT, and sensors. These edge devices are bridged and connected through network devices in the wired and wireless connection such as Industrial Ethernet, sensor network, WiFi, and 5th generation (5G), which are vulnerable to attacks.

The deployment of the IoT at the edge and the combination of actuators and sensors form an edge system that executes real-time tasks based on the sensitive and time-critical information existing in different data types [52]. With limited computation capability, they can perform restricted critical tasks such as monitoring the status of the devices or sensors and sending the update to the server to verify the condition and further action should it be required. Edge computing transforms the way to leverage data and improve operation efficiency. At the same time, those are non-time critical relevant data sent to the Cloud, which allows monitoring of the operation across the factory.

IoT gateway handles the protocols and data format conversion, enabling IT and OT convergence by sharing data between OT equipment and IT equipment. A standard protocol such as OPC UA and MQTT is adopted, allowing the edge devices and IoT to securely communicate with various services and applications running as Edge computing [53]. These protocols can run in interoperability and focus on security and sustainability to integrate the information across the network.

An example of the analysis that utilises real-time data collection from the IoT and sensors is verifying the devices' health status to ensure the equipment is in proper condition. Unlike conventional method that utilises correlation analysis, an intelligent algorithm improves the decision based on the generated model. This technique enhances accuracy and adaptability as the model can be updated based on the physical conditions and execution context. Hardware resources such as industrial Personal Computers, PLCs, gateways, and iHubs can handle the additional requirement to collect sensor data and send them to the control devices to run on the machine learning for application. As for proactive prognosis, an edge device such as an industrial Personal Computer is added close to the production. Through Multi-Access Edge Computing (MEC), the resource can be optimised at the edge while at the same time handling the IoT application and workload assignment effectively and securely [54]. The edge enhances the responsiveness of the communication between edge devices that allow decision making in an autonomous and self-organised manner.

The edge gateway can handle a wide range of network connectivity protocols supporting scalability and monitoring activities in the Smart Factory. In [55], six important functionalities of edge IoT include IoT application, Edge rule engine involving notification and callout, connectivity to the Cloud, edge analytics, edge data normalisation and edge data storage. The edge device that acts as a control should be able to run in autonomous operation and adapt to the production. Data are enhanced and transformed into a shareable form. Although the standard of OPC UA is required to be met, legacy systems and old devices still exist and cannot support these protocols. Some of these protocols used by the devices initially come with insecure communication without proper authentication. Traditionally, Modbus protocols send messages without security implementations and thrust the command entirely through the master-slave one-way communication. This protocol allows the compromised mobile and Personal Computer to issue an invalid command through the Human machine interface (HMI) to the ICS. The local network can be exposed to vulnerabilities in an attack. On the other hand, because Edge computing provides services without sending data to the Cloud, it can help to resolve data privacy issues.

2.2.4 Internet of Things (IoT)

The IoT is a resource constraint device with specific characteristics. These devices are powered by wired cable or battery for months to ten years. It is powered by limited computing capabilities to support data sharing and decision making in the functional areas. The application areas range from production to warehouse logistics. The connectivity of IoT compared to a computer system is different. IoT runs on OS, which is limited to essential task scheduling functionality such as Real-Time Operating System (RTOS) [5]. These conditions allow the IoT to optimise the performance in communication by utilising different sleep times, wake up, and functionalities, which guarantees the quality of packet delivery rate and saves battery consumption. The Institute of Electrical and Electronics Engineers (IEEE) 802.15.4 is a prominent standard defining the requirements for IoT implementation [56]. Each IoT functions differently in the application area, and it can connect to different amounts of devices based on the scenario and application. The data formats vary from device to device and are categorised as event-based and control data sending in a specific time interval. Integrating the IoT in the Smart Factory, such as in the product and scanning through Radio Frequency Identification (RFID), allows tracking of the movement and inventories. In IIoT, it refers to sensors, actuators, and robotics. The IoT requires firmware, a code in a non-volatile part of the device, to connect to the environment through the hardware interface. This device has several components, such as the kernel, bootloader, storage, and memory [57].

This firmware requires an update for bug fixes and enhancement that requires time allocation without impacting the operation of the function area. IoT OS manages the hardware and the applications that run on the devices by coordinating and allocating resources to execute the series of tasks efficiently [58]. IoT OS depends on the hardware board, memory, software installed, real-time computing, and implementation environment that deals with storage, networking connectivity, interoperability, and security. The advanced OS can support the platform to run AI. The IoT application requires the developed OS that runs on the devices to meet standards such as energy efficiency and consumption, real-time computing, network connectivity, security, heterogeneous devices support, and intelligent IoT.

The OS is optimised to support different hardware platforms in IoT to perform functional tasks. Recently, most OS has integrated with the IP networking stack by supporting network protocols such as Internet Protocol version 4 (IPv4) and Internet Protocol version 6 (IPv6). The network protocols are essential to allow the connectivity of IoT solutions. It is common for IoT OS to integrate the complete IP stack to support network communication. Also, most OT systems that connect with IoT are switched to support the IP [5]. However, the consideration of an increased number of IoT leads to the adoption of IPv6. The advantage of IP support is scalability, which has existed for many years and allows the IoT solution to share and support the application through the Internet. The advantages include application protocols for file transfer, mobility, and access to the website.

An additional consideration is the overhead of IP. As a result, adaptation through the gateway to map IP and non-IP networks is essential in the heterogeneity network. Deployment of IoT is on a large scale, and the interconnectivity of the devices exposes the vulnerability of IoT, especially since Internet connections are available [59].

The Table shows various connectivity standards used in IoT. The IoT handles the IEEE 802.11 WiFi network and Zigbee with standard IEEE 802.15. The difference between 802.11 is more power consumption with a range of coverage of around a few km. The encryption associated with the technology is available. However, its implementation relies on the IoT capacity. Due to various IoT, some do not come with encryption.

Table 2.1: Technology for IoT.

						LPWAN	
Characteristic	802.11-ah	Zigbee	BLE	RFID	6LoWP-AN	Sigfox	LoRa-WAN
Frequency	900MHz	2.4MHz	2.4MHz	various	868-915 MHz	868-902MHz	867-928MHz
Power consumption	Low-medium	Low	Low	Ultra-Low	Low	Low	Low
Range	Medium <1 km	Short	Short	Short	Short	Long <40km	Long 20km

Despite the above advantages, there exist challenges in the IoT solution. Based on the Open Web Application Security Project (OWASP) has highlighted the vulnerabilities of the IoT [60].

The top 10 IoT vulnerabilities stated in the OWASP are:

1. Weak, Guessable, or hardcoded passwords

   IoT passwords are vulnerable and easily break through brute force. The use of weak passwords that can be easily remembered is common. Some devices' passwords are hardcoded by the manufacturer and must be configured for password changes. This practice eases the technician for remote maintenance and debugging.

2. Insecure network services

   Insecure network services are often used by malicious malware before the attack, such as forming the botnet and receiving instructions from the C&C server. This unused open port and services should be restricted or turned off.

3. Insecure ecosystem interfaces

   IoT can connect a wide range of devices and application servers around the environment through network interfaces. The insecure interfaces include weak API and device connectivity authentication, such as Bluetooth Low Energy (BLE) pairing and lack of encryption.

4. Lack of secure update mechanism

   The firmware of the IoT is vulnerable and consists of essential details such as configuration. Security in updating or patching process is vital to ensure the intended firmware is installed with the correct version. If the update fails, the device should be protected from anti-rollback to ensure that only the vendor-authorised firmware is supported and prevent those deemed out of date or insecure versions of firmware from exploiting.

5. Use of insecure or outdated components

6. Insufficient privacy protection

   Privacy protection highlights the importance of data privacy and the storage that needs to be encrypted. This practice prevents the disclosure of data without the user's permission.

7. Insecure data transfer and storage

   Data needed to be protected in storage and during transmission. The network nodes should be authenticated, and data transmission should be encrypted so that no others can read the content except the intended recipient.

8. Lack of device management

   IoT is an asset that requires proper monitoring and management in various aspects, such as update management and handling of compromised devices. The device management includes verifying the connection and configuration for onboard devices, understanding the operational status and diagnosis, and maintaining devices.

9. Insecure default settings

   Default settings such as password and device configuration can introduce security vulnerabilities. Preparing alternatives for issues like allowing reconfigures of passwords and settings before being placed in the operation fields can prevent an attack.

10. Lack of physical hardening

    Take preventive measures such as disabling debug port and secure boot to validate firmware, allowing only the correct program to execute in the devices.

There are existing IoT exposed security issues, such as the thermostat or the camera that connects through the Cloud, which can handle the transaction and store necessary files which can be utilised and located afterwards, such as IP cam recording.

IoT is vulnerable when exposed to unauthorised access. There are various methods to extract essential details from the firmware, such as configuration and relevant credentials to launch a target at the victim. The Google thermostat shows that the boot process has been interrupted to load the malicious code and take control of the devices [21]. [20] highlights that the firmware's sensitive information can be easily identified and extracted since open-source tools are available. The paper [4] discusses the steps in firmware analysis, comparing many diverse embedded devices in the IoT environment and proposes improvements in the firmware analysis based on the challenges. The source code in smartphone and web applications can be extracted through reverse engineering. This technique can identify some hardcoded passwords and launch the attack. On the other hand, the IP of the interacted devices can be extracted from the Shodan search engine.

Malware is a source code installed on the target devices to achieve a range of malicious activities. Mirai was the first to detect a severe occurrence during the Distributed Denial of Service (DDoS) that caused massive damage. It has the potential to penetrate the network and intercept the traffic for a sufficient duration of time to study the sensitive details. These devices show the communication in plain text and redirect the traffic from the server, allowing the direct data flow from the IP camera to the mobile devices.

The IoT vulnerability depends not only on the devices themselves but also on the ones connected, such as smartphones and web application vulnerabilities. [21] shows that the telnet server's passwords are removed, leading to unprotected access by unauthorised access and control of the field devices. The paper also highlights the vulnerabilities of the firmware once injected through malicious activities with the source code capable of revealing the weaknesses of the devices.

In the update process, some procedures are to check on the firmware version or patch through the insecure transmission. Malpractices lead to security issues.

Another vulnerability from the web camera shows a lack of binary protection, weak server-side control, and improper use of network services [22].

2.3 The Security Issues and Challenges

2.3.1 General Issues and Challenges

The revolution of Industry 4.0 comes with various issues and challenges during its implementation. The IoT offers efficient and intelligent capabilities to perform autonomous operations in production. Sensors and actuators are the edge devices that provide or serve as the data source, sharing the data to facilitate decision-making. These devices have small computing power and firmware to communicate in the environment using the IEEE 802.15.4 standard, which best fits the resource constraint network.

The complex and highly interconnected environment requires an in-depth strategy to defend ICS in critical infrastructure [61]. Different manufacturers produce these sensors with their communication protocols without sufficient consideration of security measures and open vulnerability once implemented in the network [62][63]. In complicated cases, the IoT is manufactured by stakeholders, and another develops the firmware. Legacy devices can sometimes not support firmware updates due to limited Input/Output (I/O) capabilities [64]. In the higher capability of IoT, many stakeholders are involved in the process, including suppliers of the many parts that are put together to form intelligent devices. The sensors are implemented in a wide range and scale for the production and operation to execute normally. In [65], it highlights breaches that exist in the IoT with countermeasures. Nevertheless, this product is often easily implemented in the environment without following the proper guidelines [66].

The PLC, which runs on the Modbus protocol, conducts the one-way communication in the master-slave manner where the instruction is not authenticated and verified with the identity [67]. The existing legacy system needs additional consideration to handle or cope with the specifications and requirements, such as communication protocols with encryption. Previously, systems such as ICSs handled primary operations that received commands from sensors and actuators without the impact of the vulnerability of the Internet [68]. Also, similar to the IT system, these machines are susceptible to ARP spoofing attacks. In the OT environment, device logic is loosely related to the data, which lacks information and creates difficulty for an attacker to learn the network details. The ICS faces security challenges while moving into Industry 4.0, where the IoT network handles communication with the Internet connection [69]. Existing sensors and IoT that initially lack security measures are already in the market, and some are implemented in the industry. These products lack the infrastructure to support and implement the protective mechanism. Even some intelligent devices with some processing power cannot perform more advanced security such as encryption and authentication. It is essential to apply advanced cybersecurity, which offers considerable security advantages in the industry and can increase productivity [70].

There are security threats in IT networks, and the incidents are relatively frequent. Because the network links to the Internet, communication applies across different geographical areas. Most systems run with similar software and OS, which give attackers more information and potential methods to manipulate the system behaviours. Conversely, the OT is more destructive if an attacker has penetrated the environment and launched an attack on the critical infrastructure. Through the digitalisation of Industry 4.0, digital systems connect to the ICS that links the physical operation to cyberspace, such as CPSs. As a result, the vulnerability in one environment can be easily propagated to others. Security patches, which fix the vulnerabilities of the existing running system, are frequent in the IT environment and are automatically conducted through the Internet while patching in OT is slow and probably handled manually. Unauthorised physical access places all network devices under security threat.

Operators in the factory can issue the command from the HMI running multiple applications connected through the IT infrastructures [71]. Even if the system is not connected to the Internet, this device can expose security issues when the workers do not correctly handle the devices.

2.3.2 General Attacks Type

1. Denial of Service (DoS)

   A Denial-of-Service attacks the network of the system from operation. It is a malicious attempt to oversaturate the machine's capacity by flooding it with excess traffic or sending falsified packets that break the standard operating procedure.

2. Sybil attack

   Sybil attack occurs when the network is manipulated, and an attacker takes over the control through many fake identities and subverts the service's reputation system. This attack takes the opportunity of the peer-to-peer network by producing multiple identities and sabotaging equal resource usage of IoT [72].

3. Spoofing

   Spoofing is an attack in which the attacker misleads the node's address through a falsified packet sent by the attacker to the target network or machine. In an IP network, the local machine is identified through the local address, which is MAC and IP address. Through spoofing, an attack gains legitimate access into the network, which can prepare for a further attack, such as listening to the network, manipulating the target machine message, or simply dropping the packet received when it acts as the gateway.

4. MITM attack

   In the MITM attack, the attacker acts as an intermediate node and intercepts communication between the sender and receiver. This attack is conducted by passively listening to the traffic, intercepting the connection, terminating, and setting up a new connection, making both parties believe they are directly communicating.

5. Eavesdropping

   Eavesdropping is a passive attack where the attacker secretly listens to the network communication of the involved nodes to acquire interesting information.

6. False data injection

   False data injection attacks data integrity and operation where the physical system's state, sensor data, or control command is modified and deviates from the original. This attack targeted the lack of tamper-resistance hardware and compromised the devices. Example such as [73] shows the false data injection in CPS by Gaussian noise to replace innovation with the optimal attack.

7. Replay attack

   In a replay attack, an unauthorised node captures valid data and fraudulently resends it to the receiver, acting as if the data come from the original sender.

8. Zero-day attack

   Zero-day means the vulnerability in the software and system that remains unknown by the developer or vendor. Zero-day attack targets an unknown vulnerability in the program by exploiting and leading to abnormal functions in the target machine.

9. Covert-channel attack

   Covert channel attack evades by transferring information through a compromised device over a legitimate communication channel. Three types of covert channel attacks are i) storage-based covert, ii) timing-based covert, and iii) behaviour-based covert [74].

2.3.3. Simulation Attack

The vulnerability in the protocol poses a security threat to the network. Many IoT exist and can quickly launch a DDoS attack. The attacker looks for IoT vulnerabilities and turns them into a bot to receive a synchronised command, which launches DDoS and commits a security attack. There is often unencrypted IoT communication in the network, which the attacker targets and launches the ARP spoofing and IP fragmentation attacks.

The attacker looks for vulnerabilities in the network to steal sensitive details and bring down the infrastructure and important systems. Unauthorised physical access allows malicious code to be installed in the device through USB. IoT is Internet-oriented, with many applications in the Smart Factory. To further enhance and prepare the attack, an attacker can target vulnerable protocols, mainly ARP and IP fragmentation, as those are IP-based and regularly invoke in the specific timing that an attacker can utilise. For example, ARP resolves IP addresses to physical addresses in the local network and fragmentation of large packets in the updating or patching processes. When the attacker launches the attack, they regularly choose the timing when the system is vulnerable.

2.3.3.1 DDoS Attack

A DDoS attack targets to bring down the infrastructure or make the server unable to respond to a legitimate request by flooding the server with excessive data. This attack utilises many compromised devices caused by malware like Mirai that turns the infected devices into a botnet and synchronises it to launch the attack. Due to various challenges, there needs to be more mitigation implemented. Examples of DDoS attacks are volume-based, protocol, and application layer attacks.

Volume-based attacks - The volume-based attacks include User Datagram Protocol (UDP) flooding and ICMP flooding attacks. These attacks aim to cause bandwidth depletion at the target node.

Protocol attacks - Protocol attacks include HTTP Flood and Transmission Control Protocol Synchronized (TCP SYN). These attacks consume resources like a load balancer, firewall, or communication devices at the target node or servers.

Application attacks - Examples of application attacks are GET/POST HTTP attacks. In the application layer attack, a hacker attempts to consume the same resources from the application server with the same requests repeatedly, such as Apache, Windows, or Linux-based server. This attack is brutal to stop a legitimate request within a specific application.

The Mirai malware was first found on August 2016, and since then, several large scale incidents dealing with the malware have occurred. For example, an attack on the French web host (OVH) has been identified. The size of the DDoS attack on OVH's server claims to be 1.5 Tbsp [75]**.** The trend shows that the DDoS attack volume has increased over the years.

The DDoS attack could harm the system and become unresponsive to legitimate requests. Subsequently, the attack can sometimes adapt to the environment and manipulate the machines' behaviour, allowing attackers to steal classified information. DDoS is a network-based attack that targets Internet-based services and networks such as routers, servers such as DNS, web servers, and infrastructures. Moving into the revolution of Industry 4.0, this attack expands the target to various machines in the fields and operation areas.

Unlike the standard DoS attack, which directly attacks servers or devices, the DDoS distributed the source of the attack to form a botnet before giving a command to launch the attack to the target devices that provide services and connectivity. This source distribution gives enormous challenges for implementing security measures and protection. A single breach in a highly connected environment allows the infected devices to spread the malware across the networks. Industrial Control System (ICS) are vulnerable to this attack. Once attacked, the system resource is exhausted and interrupts the services until the server is manually shut down or turned off. The interruption impacts the operation severely, which often requires real-time communication and response to keep the operation running smoothly.

Other than that, the attack spreads the source of the attack across different networks, and the source is often unable to trace back. Those attackers that employ this method can even enhance and modify the open-source code and make the malware stealthy, eventually leading to a stealthy DDoS. In the Smart Factory, an advanced attacker can use the tools, hiding the initial attack from a small bandwidth connection or a slight increase of the traffic from the edge network, which, when accumulated at some point, leads to gigabytes of traffics.

2.3.3.2 ARP Spoofing

ARP spoofing attacks the device due to a lack of authentication implemented in the protocol. Existing research has shown various methods to protect the local network that adopts the IPv4 protocol standard, such as a static ARP table.

ARP is a LAN communication protocol that operates in the network layer in the Open Systems Interconnection (OSI) model. The ARP protocol resolves a host's MAC address given its IP address through a broadcasted ARP request packet on the network [8]. ARP is a stateless protocol. The network host will cache ARP replies automatically, and there is no predefined authentication to identify the sender. As a result, ARP spoofing can happen.

ARP stands for Address Resolution Protocol and is used in the LAN nodes to resolve address mapping in the data link layer before communication. ARP spoofing is a well-known attack employed by an attacker to interrupt communication in the local network nodes, possibly granting Internet access through the gateway to the server.

In the ARP process, a node looks for the ARP cache or table, which consists of the Internet Protocol-Media Access Control (IP-MAC) mapping. Initially, the destination IP address is known. The host needs the MAC to address the destination and send the packet to the local network. Therefore, an ARP request is broadcast to the local network, waiting for the intended recipient to reply with its IP-MAC. The ARP protocols' weakness is that there is no proper verification of the sender's identity. It is a stateless protocol with weak authentication. As a result, the host receives the ARP reply result and inserts the record in the ARP cache. Moreover, the host only takes the last reply and updates the table to worsen the situation. The attacker can send multiple ARP replies to overwrite the legitimate ARP packet record. ARP is not a security protocol and can be attacked from various scenarios, making the attack difficult to detect.

2.3.3.3 IP Fragmentation Attack

A fragmentation attack is an attack that targets the breakdown of an IP packet. In order to successfully reassemble fragments at the destination, all the fragments should have similar Identification, the correct offset of each fragment that follows the order, receive all the fragments with the last fragment indicating no more fragments follow this one, and the correct length of the data in the fragment.

There are various methods and types of IP fragmentation attacks, as described in the paper [76]:

Overlapping Fragment attack - The attack occurs when more than one fragment has offset, which indicates overlapping each other in the same packet. In this case, one data can be overwritten by another fragment.

Resource exhaustion attack - This attack targets the reassembly process caused by missing or incomplete fragments received at a destination node. The memory filled up with those incomplete fragments and become overwhelmed. Ultimately, reassembly of the packet becomes impossible.

Predictable fragment Identification - The fragment contains an ‘Identification’ field representing the fragments belonging to the same packet. The generation of this value depends on the implementation of the OS, which can be predictable. A forged fragment with similar Identification can interrupt the destination reassembly process to fail.

Evasion of Network intrusion detection systems (NIDS) - This attack attempts to mislead a NIDS and causes the target victim to reassemble the fragments sent by an attacker.

IP fragmentation attack (misassociation) targets the protocol's vulnerability by correctly guessing the IP Identifier (IP-ID) or packet header identifier and interrupting the reassembly process [77]. OS controls IP-ID based on its specific implementation [78].

2.4 Protection Methods

2.4.1 Protection for DDoS

A DDoS can bring down the infrastructure or make the server unable to respond by flooding the destination server with excessive data. This attack achieves the objective by infecting many devices with Mirai malware, which turns them into a botnet and synchronises it to launch the attack. Detection that adopts an intelligent algorithm requires training the model from a dataset, often insufficient and unavailable in the IoT network. In addition, the model's quality relates to the selected features or data attributes that can impact the predictive power of a classifier. In [79], the KDD dataset with extracted features trains the model using a Support Vector Machine (SVM) and achieves higher detection results than the Decision Tree. However, the classifier is unsuitable for large datasets and when the dataset has more noise. In [80], K-Nearest Neighbors (KNN) is a lazy learning algorithm that classified detection successfully in various applications, including the DDoS with UDP data. Like many other classifiers, the imbalanced class distribution significantly impacts KNN, especially when there is more outliers in the data point. In [81] shows that machine learning algorithms, such as Random Forest (RF) and SVM, are good at detecting the DDoS at the local attack, which supports multiple protocols such as TCP, UDP, and ICMP. The model was trained with extracted features. Although the detection rate is high, the above classifiers deliver different results and performances in DDoS detection. In particular, the RF adopts randomisation in splitting the nodes, which can generate noisy trees and impact the accuracy of a new sample.

2.4.2 Protection for ARP Spoofing

ARP spoofing attack the devices due lack of authentication of the protocol. Existing research has shown various methods to protect the local network that adopts the IPv4 protocol standard. A network segment is one of the mitigating approaches. However, this does not prevent the gateway from being attacked. Referring to paper [82], static ARP entries increase the administrative overhead, which burdens the dynamic network, which has many devices connected through a wireless connection. In [83], the method adopts packet filtering to filter and block attacker traffic. It is considered lightweight and reduces power consumption. The detection is required to set the parameter value of the window packet count and detect a vast similar ARP reply packet from the same source compared to the ARP request packet. This method detects the most IP addresses and makes hiding the attacker's real MAC through a massive volume of packets difficult. The method shown in [84] inspects each ARP packet. This method does not require storing the ARP packet's IP-MAC for validation. It detects the variant of spoofed ARP through a rule-based method by checking and comparing data link and ARP header data in the ARP packet. However, this method is time-consuming and cannot detect spoofed ARP packets other than the gratuitous variant. An attacker can adopt various methods to launch ARP spoofing, either ARP request, ARP reply, or both.

A flow-based time series entropy detection for abnormal traffic is shown in the paper [85][86], with the entropy (H) during ARP spoofing falling in 1.3 and regular traffic without attack above 2.0. A threshold can be set for ARP spoofing detection. ARP spoofing is detected using the entropy method. Generally, the effect of time series detection depends on the entropy calculation (H) shown in [87]. Entropy detection does not capture the identity of the potential attacker's address, which can be used to filter or block the network.

Equation (1) shows the entropy formula to calculate H_{field} of each window slot.

$$H(X) = -\sum_{i=1}^{n} p\,(x_i)\;log_b p(x_i) \qquad (1)$$

An abnormality is detected based on the entropy, and further action to filter and block the attack traffic is implemented. In this case, the suspicious source MAC address identified in the ARP packet can represent an attacker.

2.4.3 Protection for IP Fragmentation Attack

IP fragmentation attack (misassociation) targets the protocol’s vulnerability in which the IP-ID can be guessed and spotted, and the attack can interrupt the reassembly process [77]. This attack becomes even more effective and efficiently conducted when considerable traffic is found unencrypted. The IP-

ID is based on OS's specific implementation. Existing research proposes mitigation, such as adopting TCP to avoid fragmentation or even avoid fragmentation in the first place [88]. Based on the reference from the KDD dataset, the most relevant feature for detecting a teardrop is the ‘Wrong fragment’ field [89]. However, no declaration is found on how the wrong fragment fields are extracted [90].

Similarly, paper [91] stated that Path MTU Discovery (PMTUD) is vulnerable and can be exploited by an attacker. Low power consumption in the IoT network is needed for reliable communication and minimises retransmission's impact [92]. In case of fragmentation, it can reduce the delay latency in IoT communication links, increase the delivery rate of packets and ensure effective communication. Protection of the IP-ID is necessary to prevent the reassembly process interruption.

CHAPTER 3

DISTRIBUTED DENIAL OF SERVICE

3.1 Simulation Model

In an IoT DDoS attack, the environment is exposed to a malware infection such as Mirai that can schedule and synchronise an attack. The interconnected networks and heterogeneity of the IoT environment lead to a quick spread of the malware. The number of infected devices increases and forms a botnet capable of triggering a DDoS attack. The IoT devices behave as usual when no attack command is received. A local network comprises vast numbers of IoT, which can be connected through a computer or gateway. Therefore, even though each IoT component sends a small amount of the data packet in a consistent time, the data aggregate at the gateway becomes large. The situation is escalated when a node that acts as an Internet gateway involved in the attack has a larger bandwidth to flood the server.

The normality of IoT devices that send the data consistently can hide the earlier stages' attack phenomena. The outbound, periodic traffic dominates IoT communication consisting of TCP and UDP data. IoT devices often have limited power and resources for processing and storing data. Therefore, the infected devices are placed in sleep mode when not in the attack session, which could save energy and power for the time to launch the attacks.

Figure 3.1 shows the model of the UDP flood which is a type of DDoS triggered by the IoT devices from networks. The data aggregate at the nodes and send to the target server. The model is designed and implemented using OMNeT++. Due to the framework's design, the server has unlimited processing power and can handle all the incoming packets. The model implements DDoS with the incoming packet reaching the server side router, which serves as an entry point, and has a limited queue buffer to the server. The compromised IoT devices receive command with the time to attack, and the devices begin to trigger outbound UDP data synchronously and at a high rate. All the nodes aggregate and forward the data to flood the server. As time goes by, the queue on the server-side router increases. The attack continues and the buffer queue exceeds capacity. Flooding occurs in this situation, and the router must drop the remaining packets. Due to the incomplete packet received, the server cannot proceed with the correct processing of the application. When the server cannot receive complete packets, the flooding process exhausts the resources, which leads to unresponsiveness to the senders' requests. This situation may continue until the attack stops or the protection is implemented to recover the network.

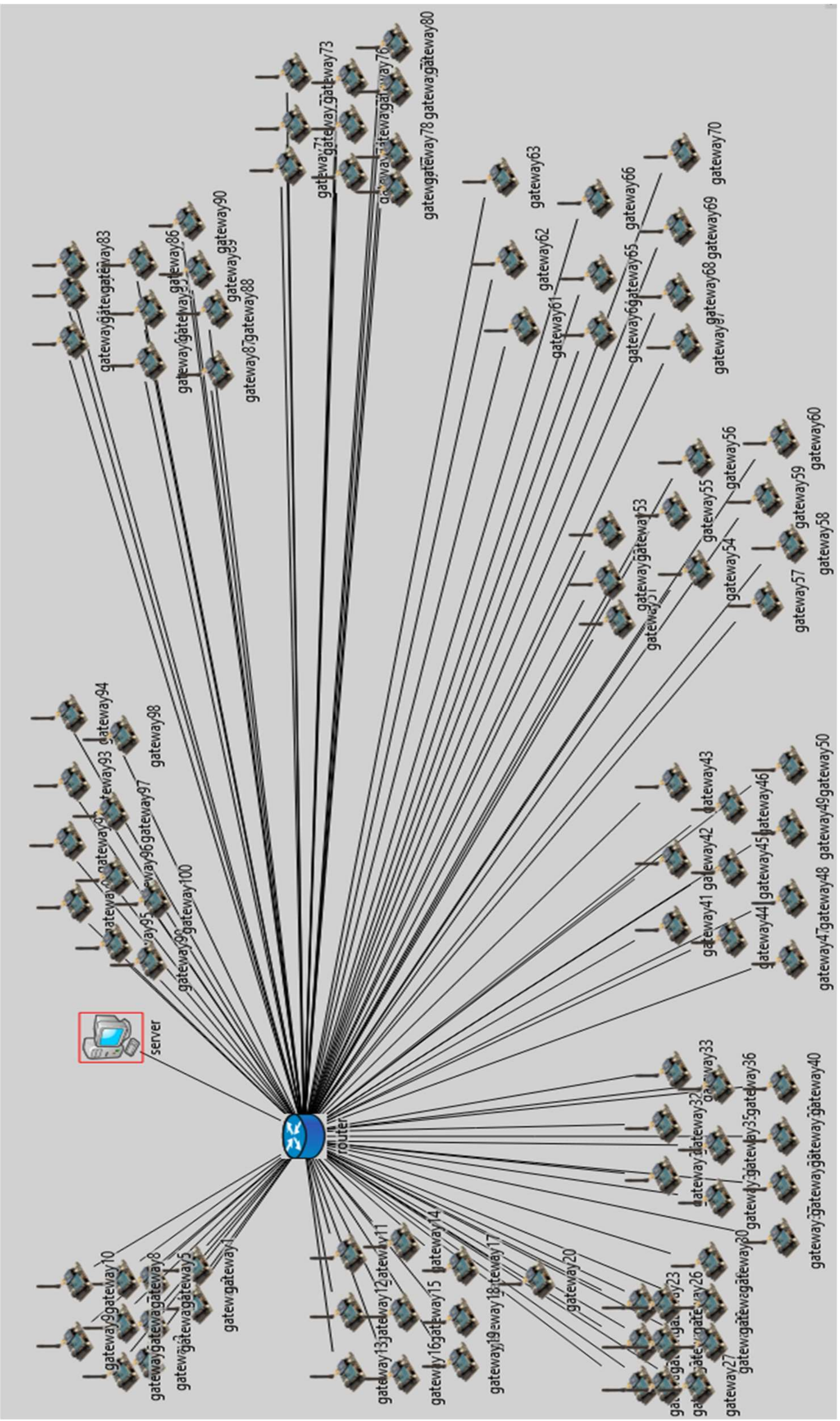

Figure 3.1: DDoS model.

3.2 Detection

Figure 3.2 compares the regular traffic simulated in 65 nodes and an attack simulated in 100 nodes, where 65 are regular nodes, and 35 are attack nodes. The server receives all incoming packets in the simulation with only regular traffic. This simulation shows that the bandwidth can cope with regular traffic. In the simulation, with 100 traffic nodes, the data sent to the server increased and reached 40000 packets. Figure 3.3 shows that the attack traffic flood the server in 0.5s-0.6s, which lead to an increase in the queue at the router. The router drops the incoming packets as they reach the queue limit, shown with total incoming packets higher than that outgoing from the server side router or received by the server. The high traffic volume during the attack session led to a drop of many packets quickly driven by the UDP traffic. Thus, the DDoS occurs during the attack session in 0.5s-0.6s.

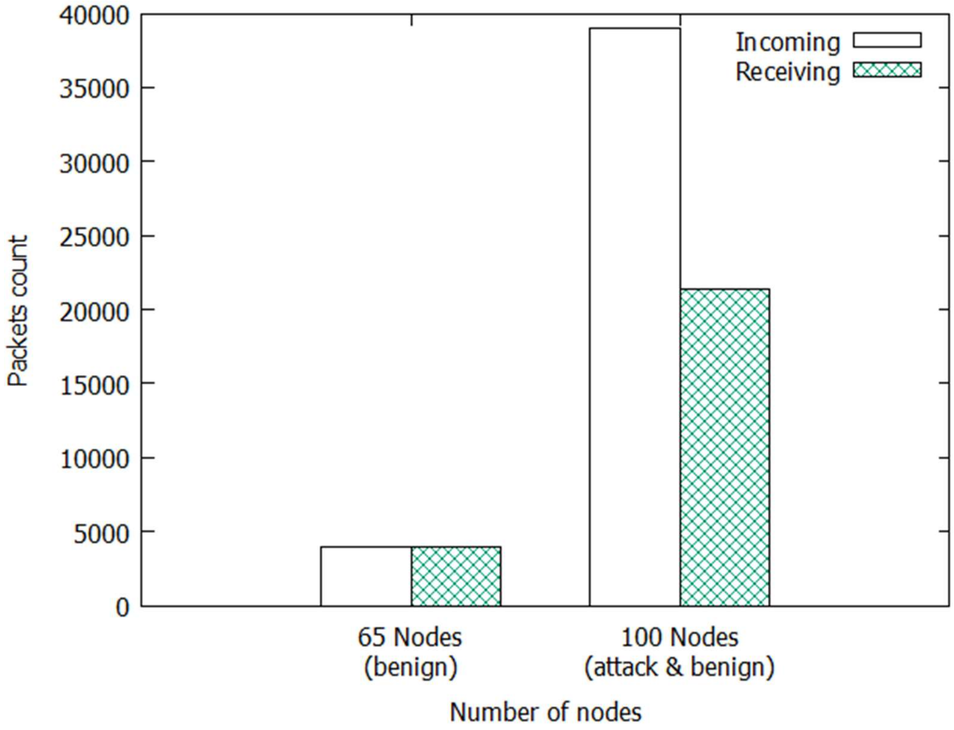

Figure 3.2: Two data flow scenarios from the server-side router to the server in 100s of simulation, one comprising 65 nodes and the other with 100 nodes.

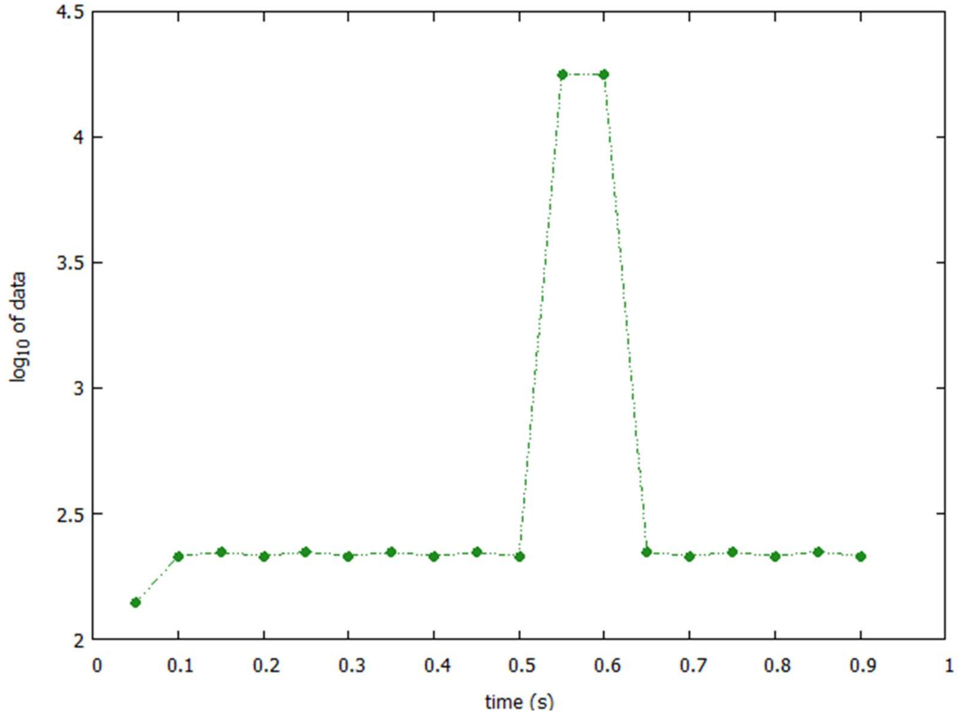

Figure 3.3: DDoS traffic with 100 nodes presents in log$_{10}$.

3.3 Verification Plan

The DDoS model with incoming data at the router is calculated by Shannon entropy, as shown in Figure 3.4. The entropy-based method relies on the packet's destination port distribution to show the network behaviour in the calculation. The verification of the model uses entropy to categorise the regular traffic, which shows a random distribution of the packet sent by each network node with a higher entropy value. In the attack, traffic often hides in the network and contributes to a high volume of flows or packets, which show a highly similar packet received by the server. A lower entropy value reflects this phenomenon. The regular traffic consists of UDP and TCP, which have different sending times. The sending interval for TCP traffic is 0.3s with different starting times, while UDP traffic is more frequent and occurs in 0.01s-0.02s. Equation (1) shows the entropy value for the measurements of network activities. The (H) value is based on every 0.05s in the verification. For example, there are 43 unique destination ports in 0.05s-0.1s, each contributed 5 messages. Thus, the occurrences of each destination port are as follows:

$$p(x_i) = (\frac{5}{215}) \log_2(1/(\frac{5}{215})$$

For 43 destination ports, the n equals 43, and the H(X) entropy value is 5.42626. During the attack session, the entropy drops to a lower value in 0.5s-0.6s below 5.25, caused by a high volume of UDP traffic flooding into the server. As for recovery, which is in 0.6s-0.9s, the entropy is 5.4-5.6. More unique addresses are found in the time slot, leading to a higher value. Thus, the changes in entropy values are used to indicate unusual events and reflect the network activities.

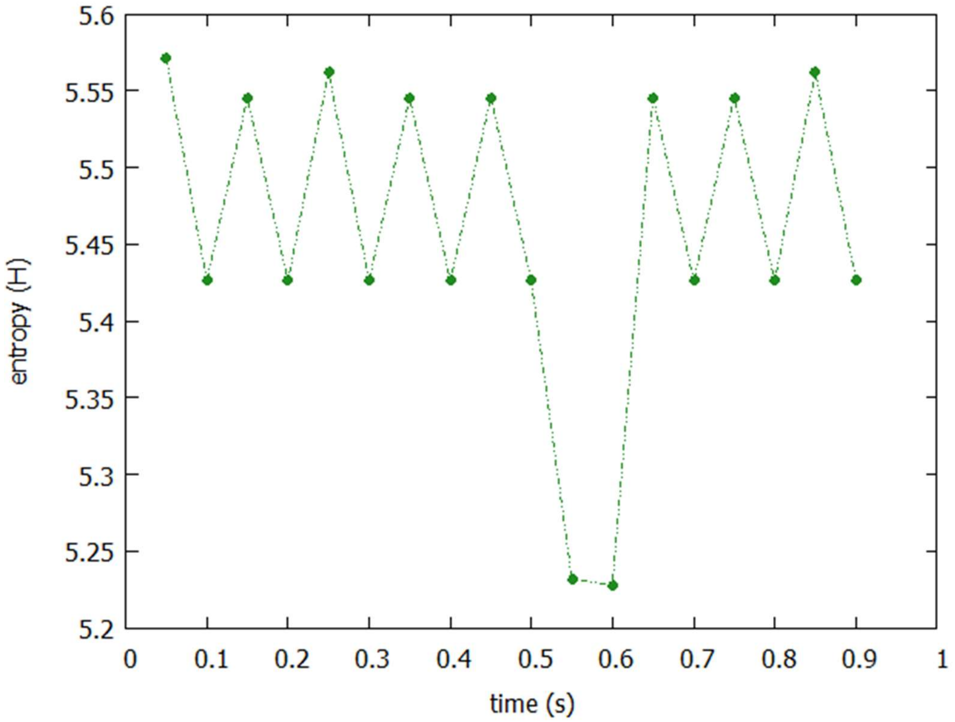

Figure 3.4: DDoS verification model using entropy.

3.4 Protection

RF is a machine learning algorithm that uses a bagging technique that relies on many decision trees and merges them for an accurate prediction. Each of the trees is called a forest, uncorrelated to each other. It can randomly select attributes to split the nodes in the decision tree without pruning. The bootstrap method uses a resampling technique with a replacement that estimates a population and copes with insufficient data to train the model. The average result of all the forests represents the final prediction, and with the maximum vote techniques, there is a high chance of getting the correct answers, as shown in Figure 3.5.

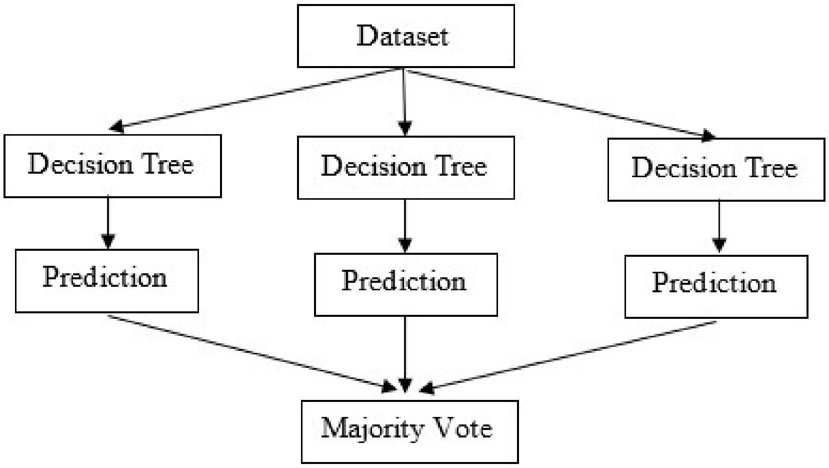

Figure 3.5: RF algorithm.

Principal Component Analysis (PCA) is the technique used in dimensionality reduction. This method converts the original data into uncorrelated PCs. The PCs are arranged from highest variance to lowest variance or in descending order. The result for PCA is to transform the dataset into PCs by maintaining the highest possible variances.

The eigenvector of matrix A and the λ which is scalar eigenvalue for matrix A (square matrix) as in Equation (2):

$$AU=U\lambda \tag{2}$$

Equation (3) is the root of the characteristic equation, which is solved to obtain the value of λ:

$$\det(A - \lambda I)=0, \tag{3}$$

As for the PCA calculation, the PCs are arranged from the largest eigenvalue to the lowest. Thus, the first selected Principal Component (PC) contributes the highest variance of the data.

To evaluate the performance of each classifier, some evaluation indicators were introduced. Some predefined definitions are as below.

TP = True positive, TN = True Negative, FP = False Positive, FN = False Negative

The formulas adopted in the evaluation of classifiers are shown below.

$$\text{Precision} = TP / (TP + FP), \tag{4}$$

$$\text{Recall}=TP / (TP+FN), \tag{5}$$

$$\text{Accuracy} = (TP + TN) / (TP + TN + FP + FN), \tag{6}$$

$$\text{F1-score} = 2 \times (\text{Precision} \times \text{Recall}) / (\text{Precision} + \text{Recall}), \tag{7}$$

3.5 Result

Table 3.1: Dataset attributes.

No.	Field name	Field name (dataset)	Data type	Description
1	Protocol	Protocol	Numerical	Protocol id represent TCP/UDP
2	Source port	srcport	Numerical	Source port from sender (incoming)
3	Destination port	destport	Numerical	Destination port (incoming)
4	Byte	byte	Numerical	Data bytes (incoming)

Table 3.2: Papers refer in DDoS.

Method	Authors	Reference
SVM	K.M. Sudar, M. Beulah, P. Deepalakshmi, P. Nagaraj and P. Chinnasamy	[79]
KNN	S. Dong, and M. Sarem	[80]
RF	J. Pei, Y. Chen, and W. Ji	[81]

Table 3.3: DDoS simulation parameter.

Parameter	Value
No. of Legitimate Nodes (Sender/Gateway)	65
No. of Attacker (Gateway)	35
No. of Router (Server side)	1
No. of Server	1
Attacker target	Server side
Simulation time	1s
Attack duration	0.5s – 0.6s
Sending Interval (Attack)	0.0001s
Sending Interval (Regular)	UDP=0.01s – 0.02s; TCP=0.3s

The dataset comprises 35% attack traffic and 65% benign traffic generated from 100 nodes. Source port, destination port, protocol, and byte are four features shown in Table 3.1. All of them are from numerical data type. Table 3.2 lists the comparing classifiers. The simulation configuration applied in the dataset is set based on Table 3.3. Based on the correlation matrix in Figure 3.6, all the features used for the classifier to train the model do not have a high correlation. The correlation between the protocol and byte is 0.326, the highest among independent attributes. Maintaining a low correlation among attributes and removing those highly correlated, higher than 0.8, can prevent the curse of dimensionality when the features apply in a model's training.

In the Scikit-learn library, the model_selection module imports the train_test_split function, and the preprocessing module imports the StandardScaler function. After the software configuration, the dataset is split into 75 percent to a training set and 25 percent to a testing set. After splitting, each set goes through preprocessing by applying a standard scaler to standardize the data value. This process can avoid the different scales in each attribute, which can impact the learning of the model with a bias. The convergence model impacts the performance in a DDoS classification. The protection relies on a high recall rate in the classification model to accurately identify the attacker.

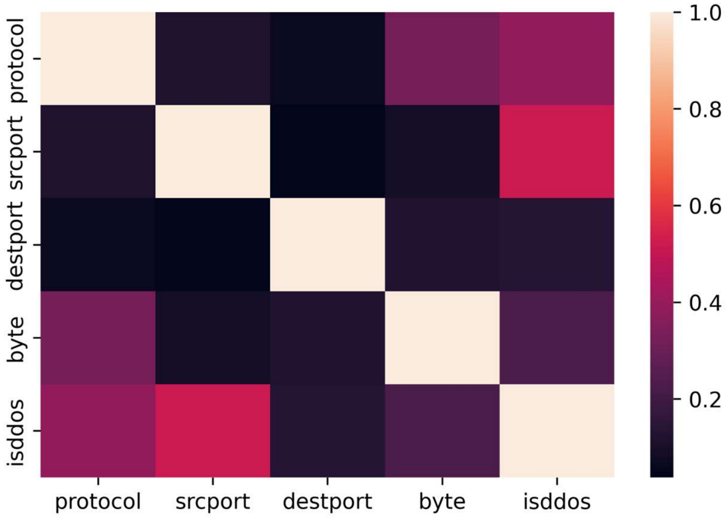

Figure 3.6: Correlation matrix (Pearson) of the dataset attributes.

Table 3.4: Classifiers' configuration.

Classifier	Library (*import*)	Configuration
Random Forest with Principal Component Analysis (RFPCA)	sklearn.decomposition (PCA)	n_components = {number of features. E.g. 2,3,4}
	sklearn.ensemble (RandomForestClassifier)	n_estimators = 10, criterion = 'entropy'.
RF	sklearn.ensemble (RandomForestClassifier)	n_estimators = 10, criterion = 'entropy'.
KNN	sklearn.neighbors (KNeighborsClassifier)	n_neighbors = 5, metric = 'minkowski', p = 2.
SVM	sklearn.svm (SCV)	kernel = 'rbf'.

In the protection, Table 3.4 shows the configuration of each classifier that runs on the dataset to generate the convergence model adopted in the later classification problem to identify the standard and benign traffic. Python software implements the configuration as stated in each classifier with the imports of the library stated. In PCA, the attribute n_components represents a number of features used in the model. In RF, the attribute n_estimators is set to 10, representing the number of trees in the forest. In KNN, the attribute n_neighbours is set to 5, representing the number of neighbours and the Minkowski metric with the power parameter is set to 2. Lastly, the SVM classifier is configured with a Radial Basis Function kernel.

Table 3.5: Comparison result of classifiers.

		RFPCA	**RF**	**KNN**	**SVM**
2-Feature	F1	91.43	83.33	69.7	74.67
	Accuracy	94	88	80	81
	Precision	91.43	81.08	74.2	70
	Recall	91.43	85.71	65.71	80
3-Feature	F1	91.18	89.55	81.01	79.02
	Accuracy	94	93	85	83
	Precision	93.94	93.75	72.72	69.57
	Recall	88.57	85.71	91.43	91.43
4-Feature	F1	95.65	91.18	82.67	80.52
	Accuracy	97	94	87	85
	Precision	97.06	93.94	77.5	73.81
	Recall	94.29	88.57	88.57	88.57

Precision and recall calculations are shown in Equation (4) and Equation (5), respectively. Based on Table 3.5, the proposed method of RFPCA has the highest accuracy as in Equation (6). The prediction of RF, KNN, and SVM by two features, three features, and four features are improving in terms of the F1-score, as in Equation (7). Prediction in the four-feature model of RFPCA achieved a 95.65% F1-score and 97% for accuracy. Figure 3.7 shows the prediction of each classifier based on a different number of features. RF, KNN, and SVM started with a lower F1-score in the two features model, which adopted protocol and source port to train a model, and the score increased in the three features training process of a model, which included the destination port.

The same models in Figure 3.7 are applied to the DDoS classification problem presented in the simulation. The entropy and Log_{10} results highlight the different detection rates by the classifiers in the network analyses. In general, more features in training lead to higher accuracy and take more computation time. The traffic analyses are shown in Figure 3.8 in a two-feature model, Figure 3.10 in a three-feature model, and Figure 3.12 in a four-feature model. Before the attack, entropy was maintained between 5.43-5.57 in 0s-0.5s. A huge entropy dropped, showing the high volume of attack concentrated in 0.5s-0.6s. In Figure 3.8, entropy for KNN achieved 3.81 at 0.6s, with a 65.71% recall value, the highest entropy value compared to the other classifiers. Entropy for RFPCA achieved 2.55 at 0.6s with the same recall value, and precision achieved 91.43%, becoming the best model in this category. In Figure 3.10, entropy for SVM dropped to 2.3 at 0.6s, with a high recall value of 91.43% and achieved 69.57% precision. This shows the high concentration of attack traffic in 0.5s-0.6s. In

Figure 3.12, the entropy for RFPCA dropped to 2.43 at 0.6s and maintained at 5.39-5.53 after 0.6s, closest to normal. Traffic analyses adopt Log_{10} with the two-feature model shown in Figure 3.9, the three-feature model shown in Figure 3.11, and the four-feature model shown in Figure 3.13. In Figures 3.9, 3.11, and 3.13, traffic maintained between 2.15-2.35 in 0s-0.5s. In Figure 3.9, the $\log_{10}$ value for RFPCA dropped to 3.23 at 0.6s and maintained between 2.3-2.32 at 0.6s-0.9s. In Figure 3.11, SVM dropped to 3.21 at 0.6s and maintained between 2.16-2.18. KNN dropped to 3.23 and maintained between 2.19-2.21 after 0.6s. In Figure 3.13, the $\log_{10}$ value for RFPCA dropped to 3.08 at 0.6s. It maintained between 2.32-2.34 in 0.6s-0.9s. The $\log_{10}$ graphs show the effectiveness of the RFPCA to block and filter out the attack traffic in 0.5s-0.6s and quick recovery of the network traffic back to normal stage.

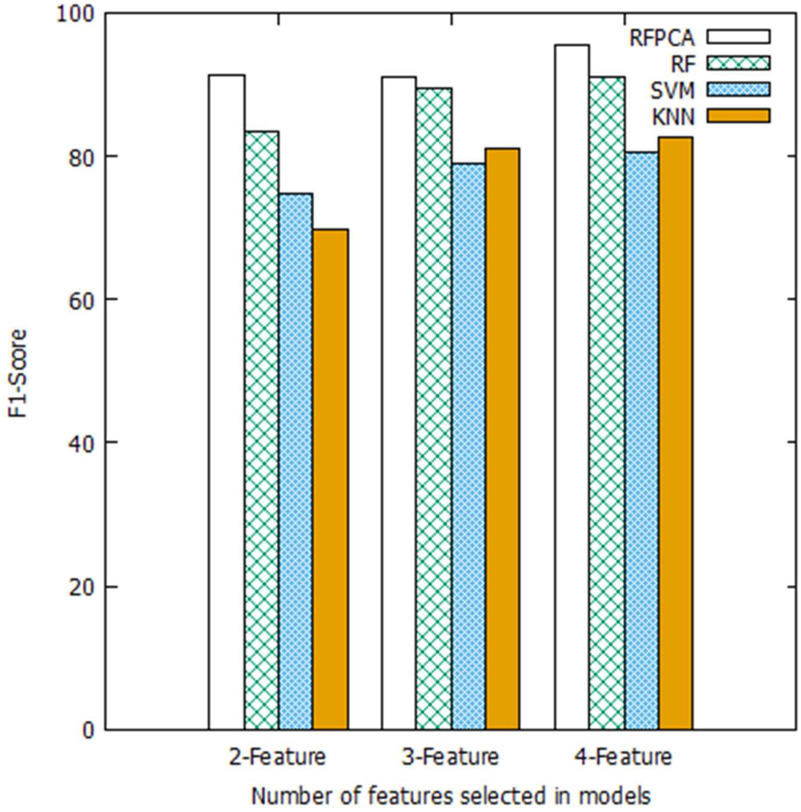

Figure 3.7: F1-score comparison of classifiers.

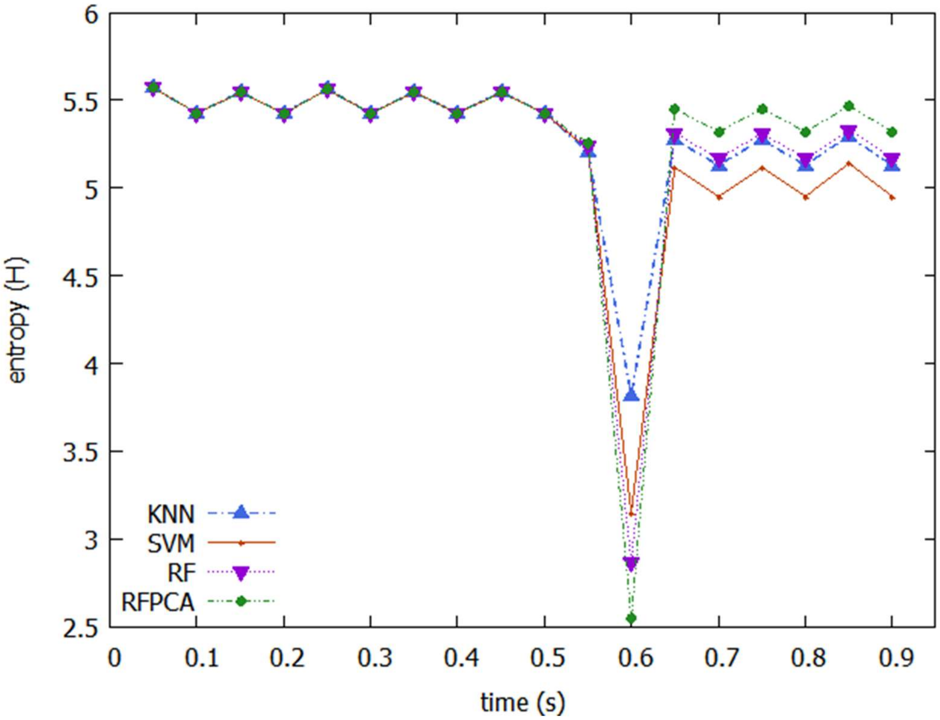

Figure 3.8: Traffic analysis of two-feature models calculates in entropy.

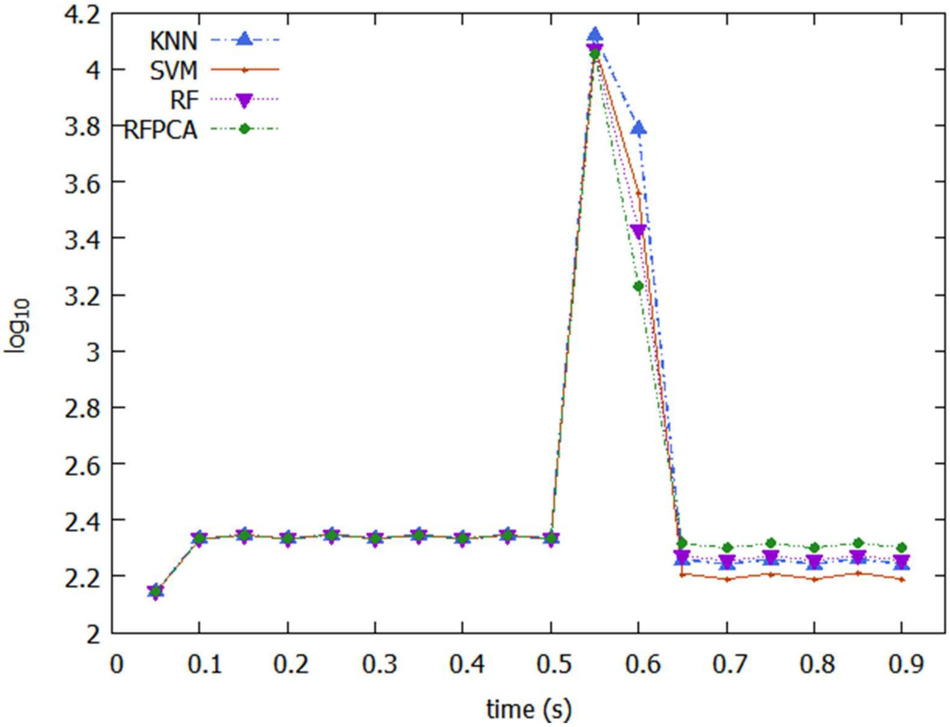

Figure 3.9: Traffic analysis of two-feature models calculates in Log_{10}.

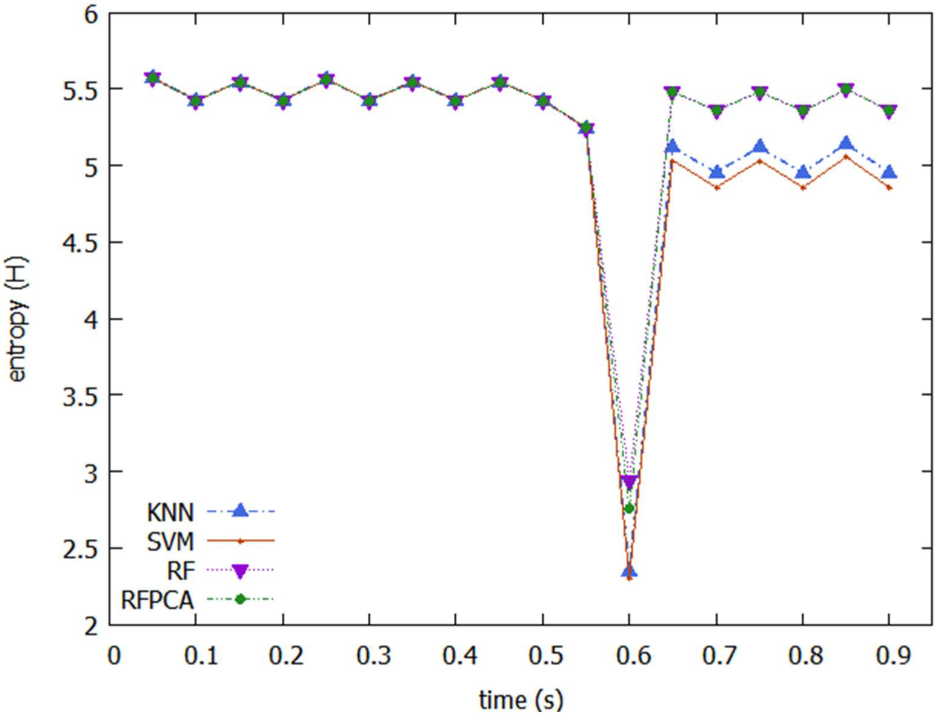

Figure 3.10: Traffic analysis of three-feature models calculates in entropy.

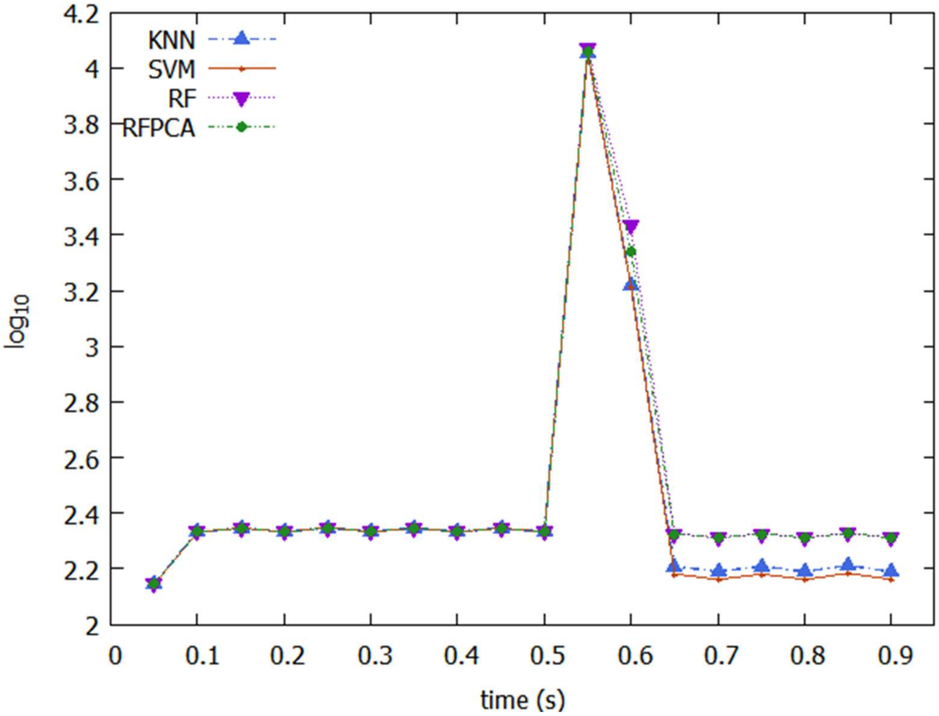

Figure 3.11: Traffic analysis of three-feature models calculates in Log_{10}.

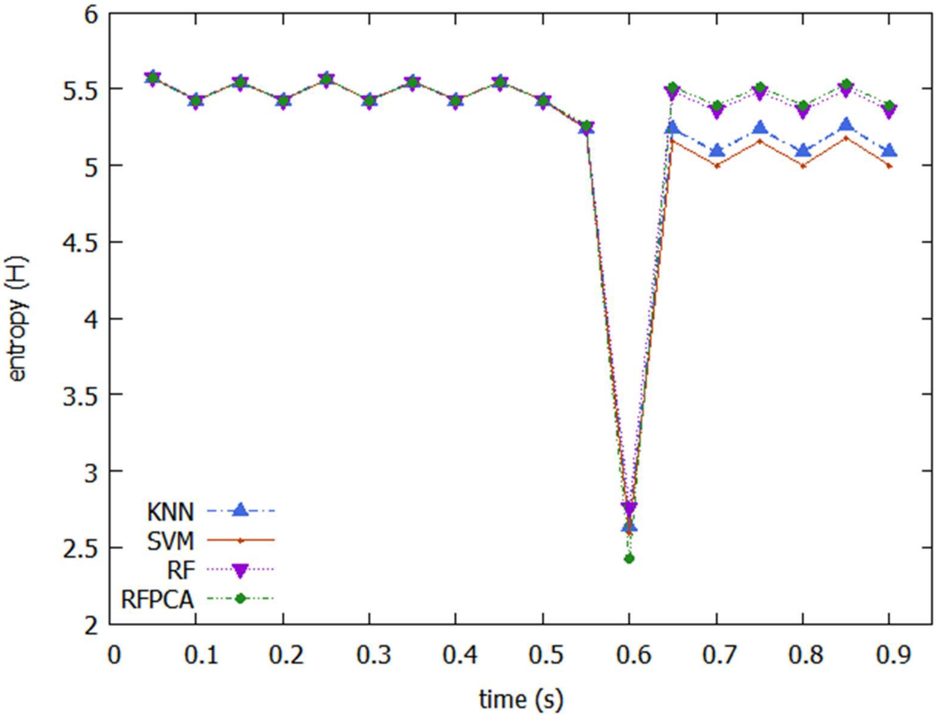

Figure 3.12: Traffic analysis of four-feature models calculates in entropy.

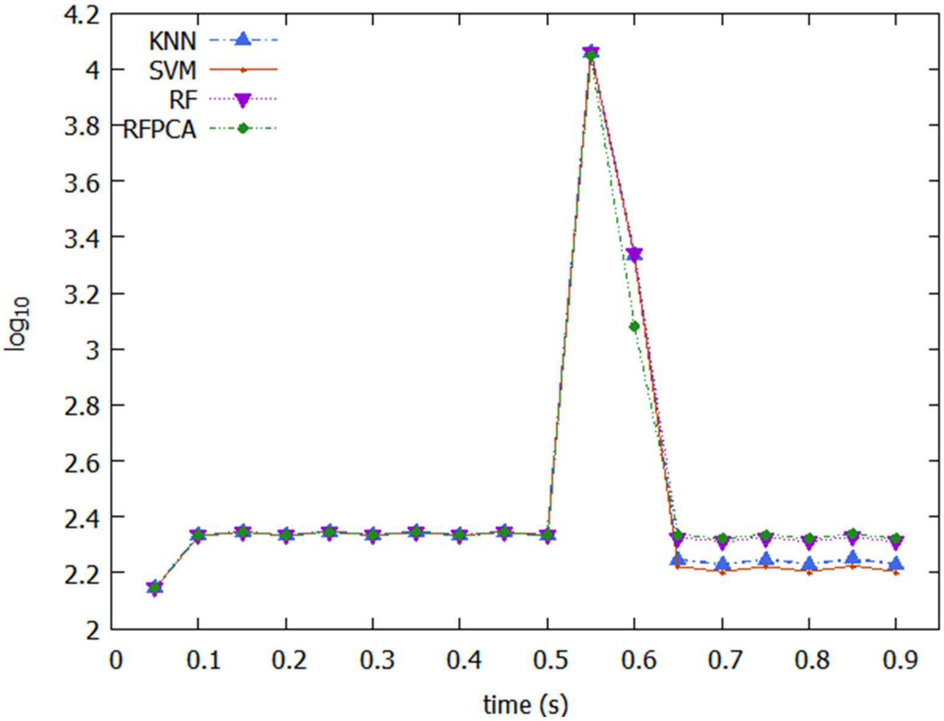

Figure 3.13: Traffic analysis of four-feature models calculates in Log_{10}.

3.6 Concluding Remarks

In a DDoS, the attacker looks for the vulnerability in an IoT, quickly infects many devices with malware, and forms a botnet capable of launching an attack targeting critical infrastructure. In this model, a massive number of attacking nodes consisting of 35 out of 100 nodes are modelled in the simulation, sending vast amounts of data continuously during the attack session in 0.5s-0.6s. This attack caused the target server to be unable to respond to a legitimate request and dropped the packets at the server entry point, which is represented as a router in the model. In the volumetric attack, this often requires enormous resources to handle the attack and recover the server with a high recall rate. The protection depends on the effectiveness and efficiency of correctly classifying the traffic and blocking the DDoS attack. Thus, the convergence of a model is related to the classifiers' performance. The selected classifiers use a dataset of 100 samples split into 75 per cent for the training set and 25 percent for the testing set. The trained model is evaluated with F1-score, accuracy, recall, and precision, showing the impact of the performance to detect the attack and recover the network. The four-feature model of RFPCA had an overall prediction performance with a recall of 94.29%, detecting most of the attack traffic and precision of 97.06%, showing good recovery. RF has also improved for the four-feature model compared to the two-feature and three-feature models. It had a similar recall of 88.57% with KNN and SVM in four-feature models. RF shows good classification performances with minimal computation. When a DDoS has been detected, the classifiers with a run on the packet data flood to the server to filter and block all the matches attack traffic patterns based on the model trained in the dataset.

CHAPTER 4

ADDRESS RESOLUTION PROTOCOL SPOOFING

4.1 Simulation Model

ARP spoofing occurs when the attacker attempts to participate in the communication as a legitimate node, as shown in Figure 4.1. The model connects local nodes through a switch or AP to the router. The link from the local nodes to the switch or AP can be wired or wireless. A node in the simulation can be a mobile device or IoT, and the switch can also represent a wireless AP. ARP spoofing can further attacks, such as DoS, DDoS, and MITM attacks. ARP spoofing starts by gaining the trust of the local network. The attacker screens the local network by sending bulk ARP packets with a range of destination IPs. This process allows the attacker to gain knowledge of the network topology and all the valid IP-MAC addresses currently active. This process continues until the attacker gains sufficient network knowledge and is ready to launch the ARP spoofing attack. In this case, the attacker targets are the router and Node03.

ARP protocol is a principle to acquire a physical address, a MAC address where the sender only knows the IP address of the receiving host. Therefore, in an ARP spoofing, the attacker sends the target a forged MAC address and IP address. The destination host updates its ARP table. It follows the steps for the attacker to commit an ARP spoofing attack. 1) The attacker initiates the attack based on the attack mode. This can either issue an ARP packet to destination hosts or reply to the host ARP request. In either case, the ARP packet consists of forged IP-MAC mapping. 2) The destination host receives the ARP packet from the attacker and updates its ARP table without verification of the attacker's identity. 3) At this point, ARP spoofing is complete. The attacker’s MAC is linked with the compromised nodes, and traffic sent to these nodes will head to the attacker instead. The attacker listens to the traffic between the two compromised nodes. ARP cache has the timeout set in each host, representing the validity duration of each address mapping. In this model, if the record is expired, the host must update the status before the following communication occurs. Thus, the attacker must send an ARP packet again to acquire the mapping, and an ARP request is broadcast into the network. In typical cases, once the attacker takes over the traffic of compromised nodes, this is followed by sending the spoofed ARP packet to the targets continuously or intermittently throughout the attack session. This action enables the attacker to maintain the connection with the target and learn the sufficient details of participated nodes without being interrupted by any other causes, such as legitimate nodes’ ARP packets or ARP cache expiration. The attacker forwarded all the messages to the destination node and acted as a forwarding. Also, the attacker activity shows an increase in ARP packets sent to the local network. The reasons for the previously mentioned situation are 1) screening activity consists of a wide range of possible IP addresses that can be utilised in the local network, and 2) In the attack session, an attacker sends redundant ARP packets to intercept the communication. Any ARP packet from a legitimate node will not successfully insert at the target host, as multiple packets or the last packet from the attacker will overwrite the address mapping. All participants who have passed through the ARP process can communicate directly with each other. The router can handle communication outside the network. The model applies to all three modes of attack.

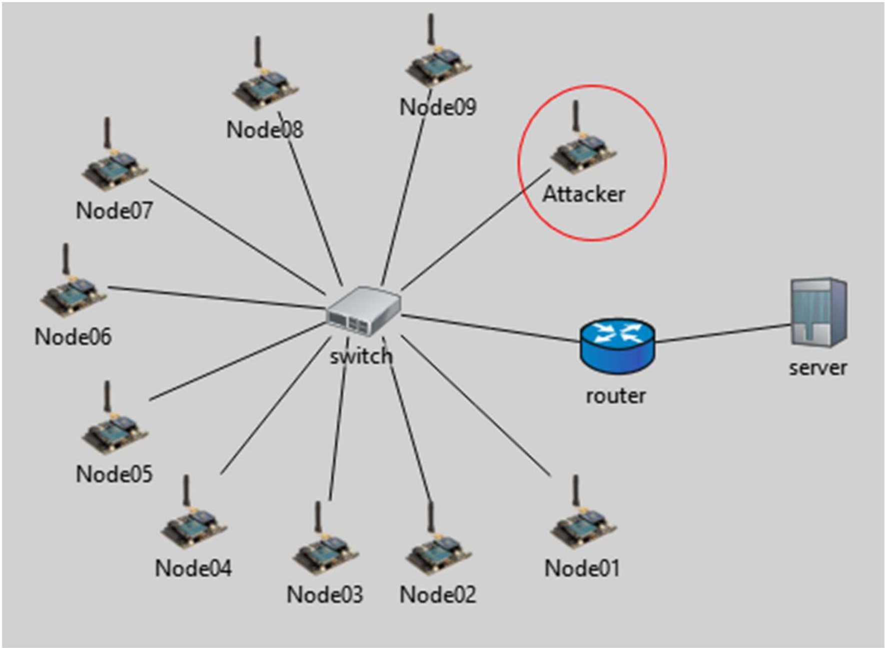

Figure 4.1: ARP spoofing model.

4.2 Detection

The attacker begins by scanning the network with its real IP-MAC to learn the network and acquire active IP addresses. This process involves many ARP packets sent into the local network. Then, the attack session occurs in 0.2s-0.7s after sufficient preparation by the attacker. The original IP for the router is 10.0.0.17, and the Node03 IP is 10.0.0.20. The attacker issues spoofed ARP, which contains forged IP-MAC mapping of the attacker's address to the intended target. When Node03 receives an ARP packet, it updates with the attacker's address instead of the router. Similarly, the router updates the MAC of Node03 to the attacker's address. Table 4.1 shows the changes of address in the ARP entries. The attacker acts as a forwarding node. Once the ARP spoofing occurs, the data will go through the attacker and route to the destination. Figures 4.2-4.4 show that the high volumes of ARP packets occur in the screening activities in 0.0s-20s, followed by the attack. ARP packets are sent intermittently to maintain a connection with the targets in the 20s-70s. Network traffic of ARP packets in the 20s-70s shows that a MAC address occurs in multiple IP addresses. A MAC address should not link to multiple IP addresses in a local network, which occurs in a relatively high volume of ARP traffic. Thus, ARP spoofing occurs and is detected.

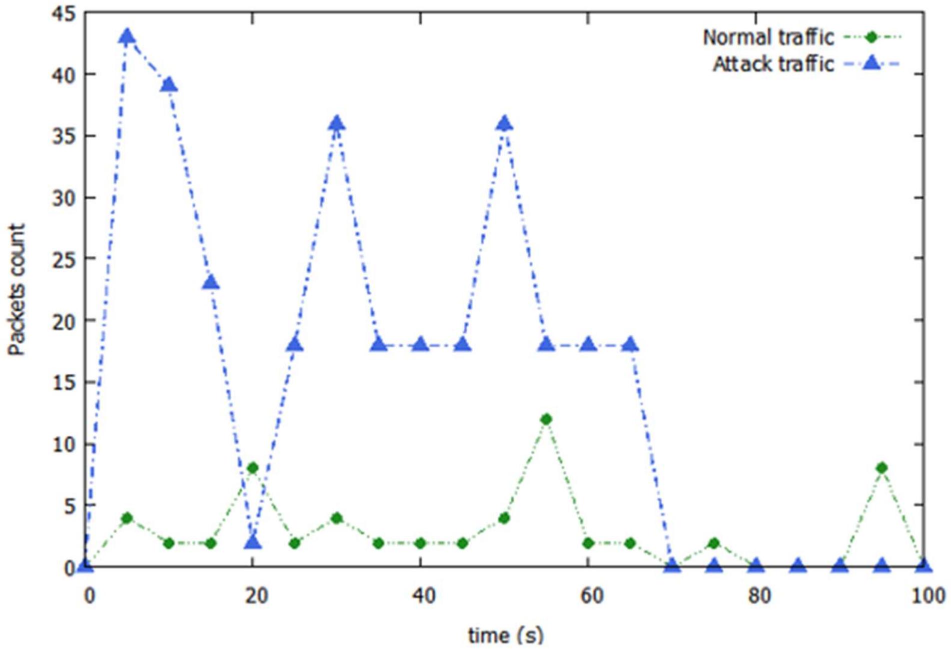

Figure 4.2: Mode-1 traffic for ARP packets comparison.

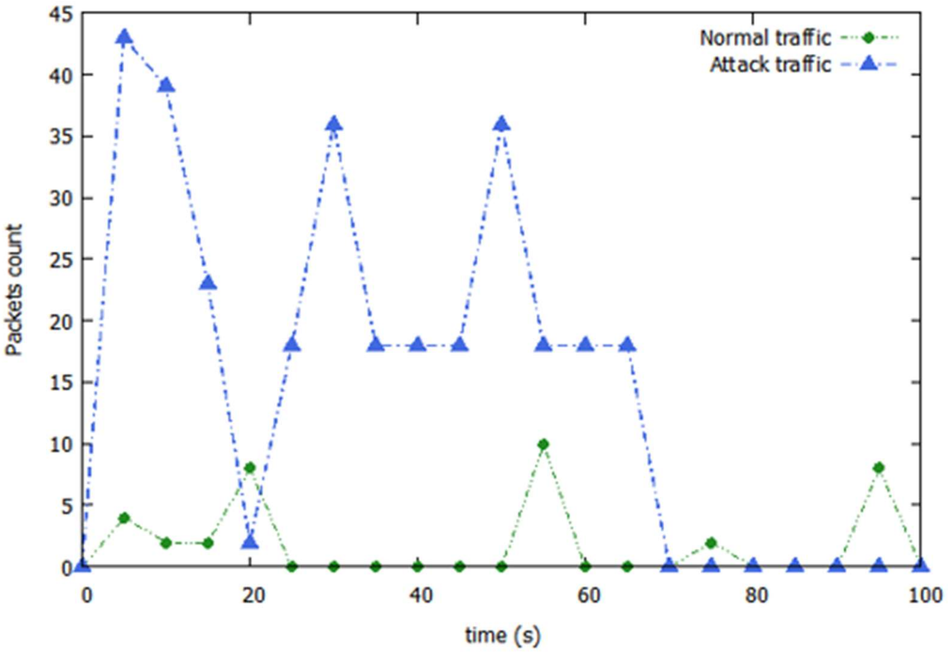

Figure 4.3: Mode-2 traffic for ARP packets comparison.

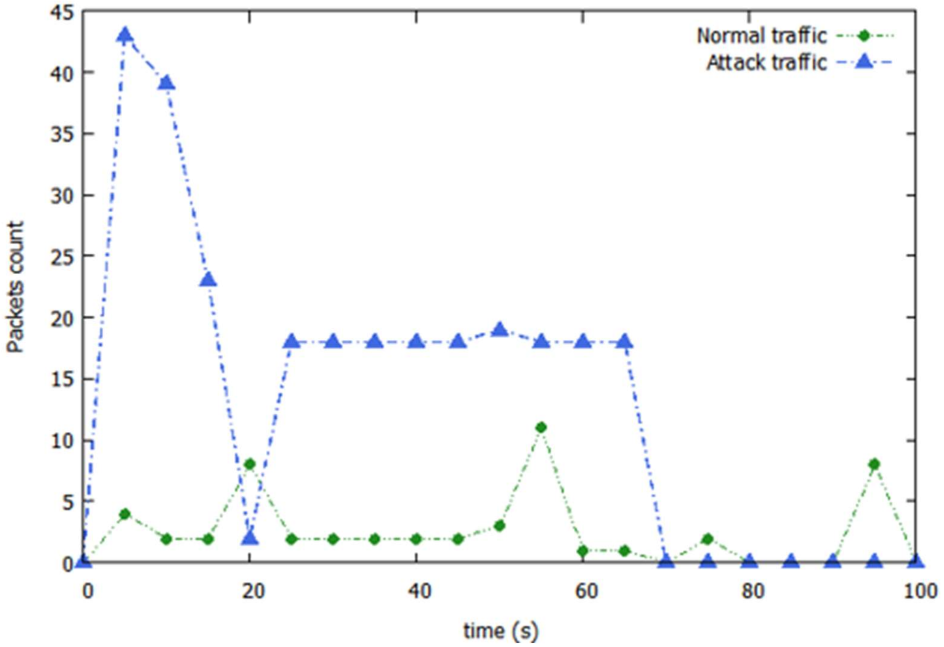

Figure 4.4: Mode-3 traffic for ARP packets comparison.

Table 4.1: ARP cache entries.

	Before	After
Node03	0A-AA-00-00-00-11	0A-AA-00-00-00-0F
Router	0A-AA-00-00-00-0C	0A-AA-00-00-00-0F

4.3 Verification Plan

Table 4.2: ARP spoofing attack mode.

Attack mode	Description
Mode-1	The attacker sends spoofed ARP requests to the router and Node03.
Mode-2	The attacker sends spoofed ARP replies, mimicking a gratuitous ARP variant to the router and Node03.
Mode-3	The attacker replies to a legitimate ARP request with spoofed ARP messages to Node03 and then sends spoofed ARP requests to the router.

*Note : Attacker sends the spoofed ARP packets continuously during the attack session to maintain the connection between target nodes.

The traffic data are collected at the point of AP and used to verify the model of ARP spoofing with the entropy method. The simulation presents three attack modes, which are shown in Table 4.2. The model describes the ARP process that occurs in 0s-100s. Besides the local communication from normal nodes, the attacker performs screening activity by sending the bulk of ARP packets to the network before the attack session. For the Mode-3 attack, the attacker continues with ARP requests to the targets from the 20s-45s. Node03 initiates another legitimate ARP request when the timeout of the ARP entries. The entropy shows a lower value when the attack occurs. On the other hand, the benign traffic is distributed. In Modes 1 and 3, the entropy values are higher than in Mode-2, where target nodes are involved in the ARP process during attack sessions in the 20s-70s.

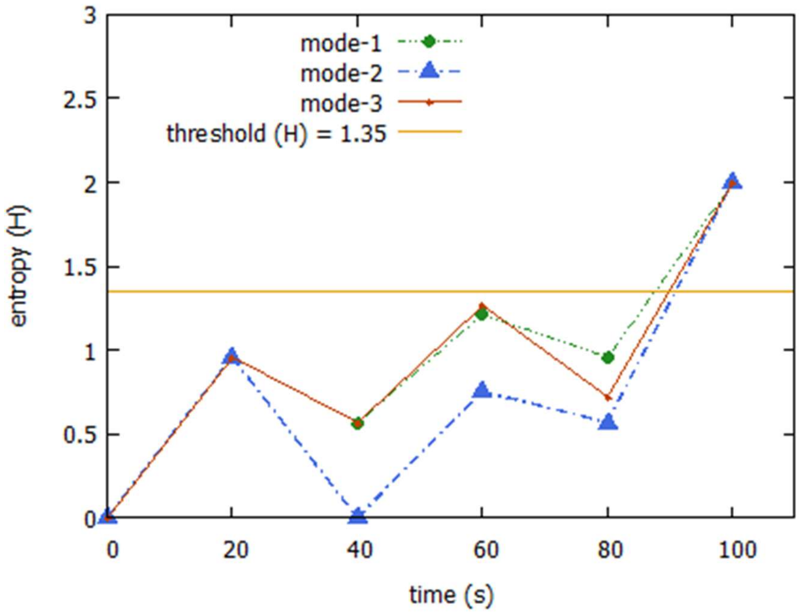

Figure 4.5: ARP spoofing verification model.

4.4 Protection

Algorithm 1 uses entropy (H), as shown in Equation (1), to protect the local network. The entropy in the ARP spoofing mainly falls to 1.3 [85]. The entropy in an ARP spoofing can achieve a high value but not too high compared to the normal ARP process. The threshold shows the gaps between normal and abnormal in ARP traffic, which triggers detection and protection. In this case, a threshold of 1.35 is set.

Let S be a set of all ARP packets collected at the time window. The x_i is the number of packets sent from a source. To calculate $p(x_i)$, x_i is divided by a denominator of the total number of ARP packets at the time window or timeslot i. The b is the base of a log which is set to 2.

The ARP traffic is collected at each time window and compared with the predefined threshold to detect anomalies in the network traffic flow. The entropy calculates with the sender's MAC address, shows the underlying probability distribution of the ARP packet, and describes the randomness of the flows in the observation, in this case, at the switch or an AP. The MAC address is selected to calculate the entropy of each time window. This feature shows the robustness to identify the attacker's behaviours correctly. In the usual case, local nodes communicate regularly, involving active IP addresses. Suppose the attacker begins to screen and learn the topology of the local network by looking for all active IP addresses in the local network. In that case, this can involve all possible IP addresses. The attacker generates significant ARP packets during the screening of the local network or launch of the attack. In such a case, the high volume of attack traffic leads to a drop in the entropy value.

Algorithm 1 : proposed ARP spoofing detect and protect based on entropy

```
log_monitor_ip_mac: dictionary to keep IP-MAC mapping
Input : Time Window slot (W1. . . Wn) = S, curEntropyAttr1 as sender MAC
Output :
Set Entropy threshold ← 1.35
Set log_monitor_ip_mac ← []
for all Wi ∈ S do
   // get entropy of the ARP traffic in current window slot
   curEntropyAttr1 ← COMPUTE entropy of sender MAC
   if (APi(MAC) in capture MAC address)
   then
      filterAttackerTraffic()
   endif
   // raise alarm
   if (curEntropyAttr1 < Entropy threshold)
   then
      ARP List ← UNIQUE IP, MAC for high volume of ARP packet’s in Wi
      // get the Sender IP, MAC in ARP
      MAC ← MAC in ARP List
      IP ← IP in ARP List
      length ← COUNT ARP List
      if (length > 1) // duplicate address found
      then
         detect spoofing ← True
      else
         match ← matchMacToIp (IP, MAC,
log_monitor_ip_mac)
      if(!match) then
         detect spoofing ← True
      end if
   end if
   if (detect spoofing = True)
   then
      // keep the mac address in a list
      UPDATE MAC in capture mac address
   else
      // keep the IP-MAC in dictionary for further
validation
      UPDATE IP, MAC in log_monitor_ip_mac
   end if
end if
```

4.5 Result

Table 4.3: Comparison of detection and protection.

Method	Description
Propose method	ARP packets are grouped by time window. Entropy triggers the protection algorithm to check for any IP-MAC mapping mismatch or duplicate IP showing the attacker's address.
Method-1 [83]	Each window consists of a maximum of 260 packets. In the separate window, if less than 1/3 of the ARP requests belong to the ARP replies triggered in bulk, then the source address in the ARP reply is the attacker's address.
Method-2 [84]	Spoofed ARP, a variant of gratuitous ARP, is detected if the destination address specifies the destination node.

Table 4.4: ARP spoofing simulation parameters.

Parameter	Value
No. of Attacker	1
No. of Router	1
No. of Server	1
Attacker target	2 nodes (Router and Node03)
Simulation time	100s
Attack duration	20s–70s
ARP cache timeout	30s

The setting in the proposed method is entropy threshold H < 1.35 and 20 seconds per slot of a time window. The simulation generates a csv file. The file contains ARP packet details and frame details with the protocol. The entropy drop is caused by excessive ARP packets from a single source during the attack.

In Figures 4.6, 4.8, and 4.10, entropy dropped below the threshold in 0s-20s. In this session, the attacker screens the local network, which is the general case where the attacker attempts to identify the active IP in the local network. The screening process continues until the reach of the attack session. The entropy value dropped due to the significant amount of ARP packets from the attacker in the 20s-80s. In comparing the detection and protection, the entropy shows the effectiveness of detecting abnormality in the traffic. The entropy values were relatively high in the 20s-70s, as shown in Figures 4.6 and 4.10, when the victims were involved in the ARP process. During the attack session, when cache timeout occurred in the 40s-60s, local nodes initiated the ARP process, which led to an increase in entropy that achieved 1.22 in Figures 4.6 and 1.27 in Figure 4.10. Based on the proposed method reaction to this abnormal traffic, Algorithm 1 reads and processes traffic in each time window and inserts rules to filter the ARP packet based on the attacker's MAC. The entropy value equals or exceeds the threshold value for benign traffic, and the ARP packet distribution is random.

In Figure 4.7, the proposed method detected ARP spoofing and blocked the malicious traffic in the 40s. There was no action from Method-1 and Method-2, and total traffic remained the same.

In Figure 4.9, Method-2 detected the attack at the 20s. This is a variant of gratuitous ARP to forge the target's IP-MAC mapping. The proposed method detected ARP spoofing in the 40s. Method-1 blocked the attacker at the 46s.

In Figure 4.11, the attacker MAC can only be identified after ARP cache timeout and the Node03 initiated the ARP request at the 45s. Method-1 blocked the attacker at 51s. The proposed method blocked the attacker in the 60s. There were no responses or measures taken for Method-2 in this case.

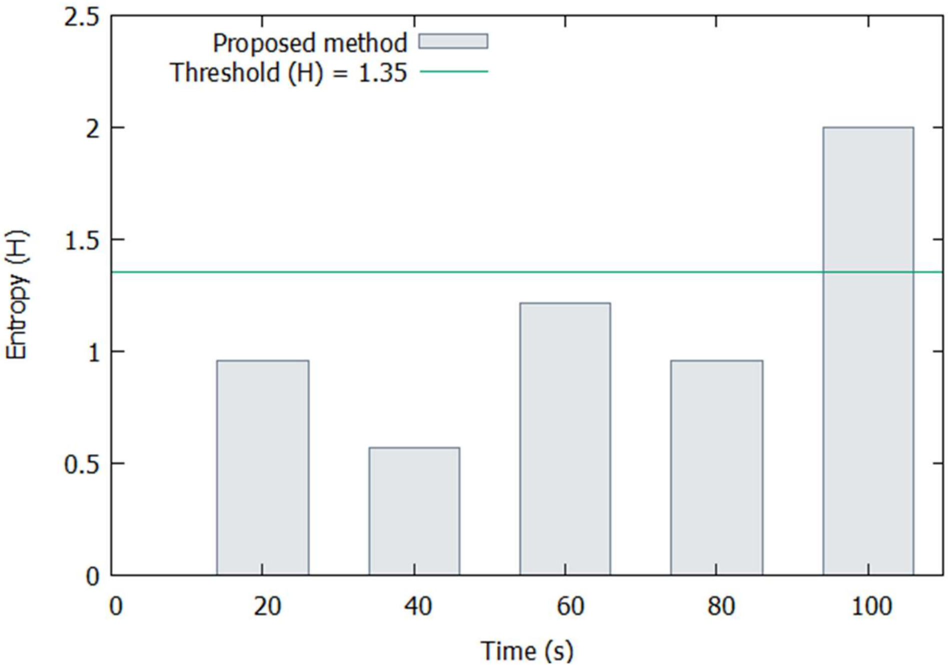

Figure 4.6: Entropy (H) of proposed method for Mode-1 scenario.

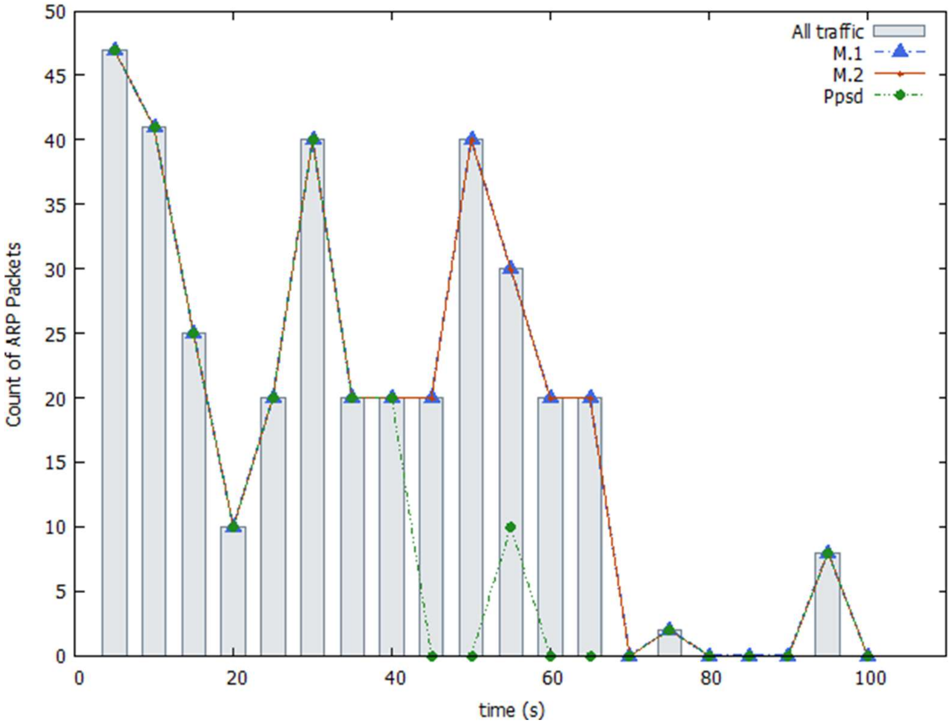

Figure 4.7: Comparison result of Mode-1 scenario.

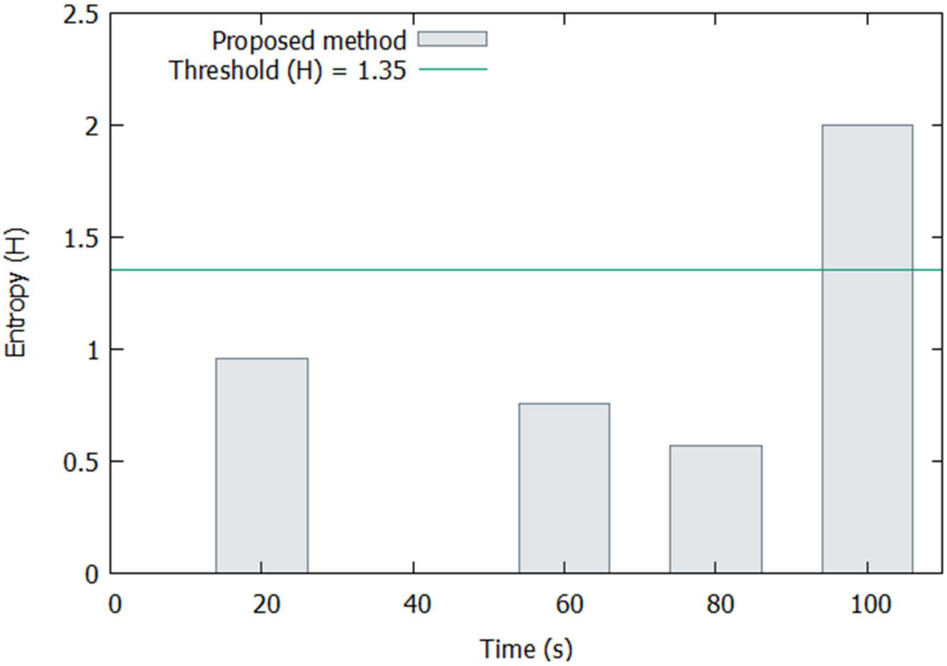

Figure 4.8: Entropy (H) of proposed method for Mode-2 scenario.

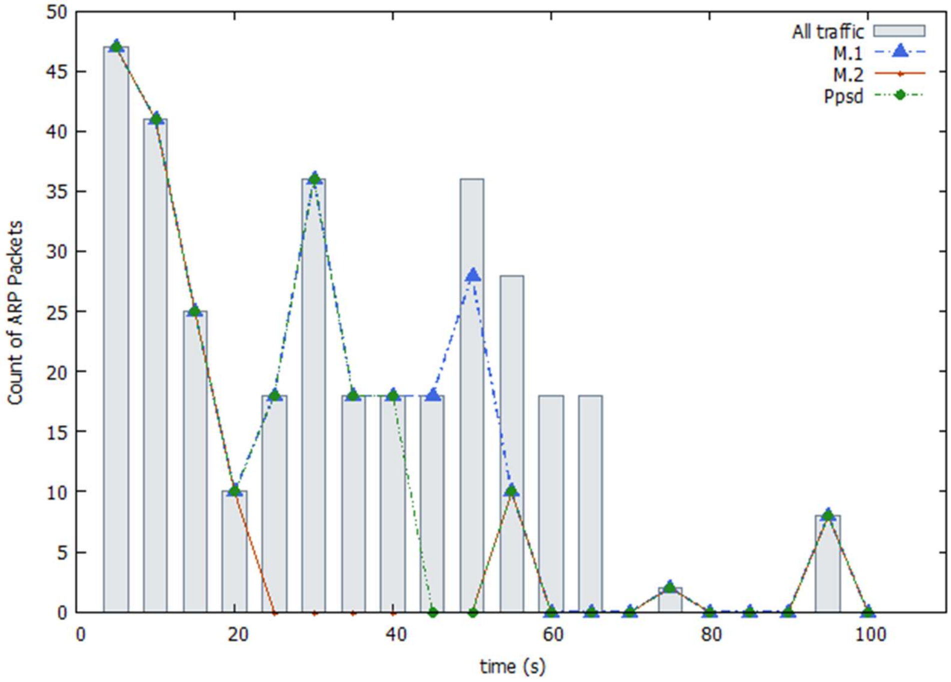

Figure 4.9: Comparison result of Mode-2 scenario.

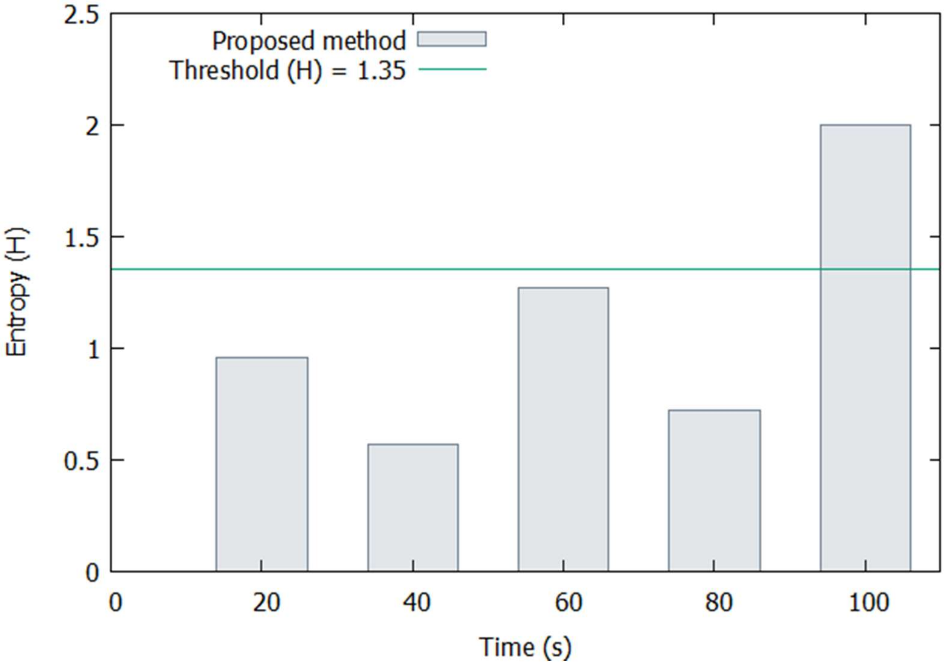

Figure 4.10: Entropy (H) of proposed method for Mode-3 scenario.

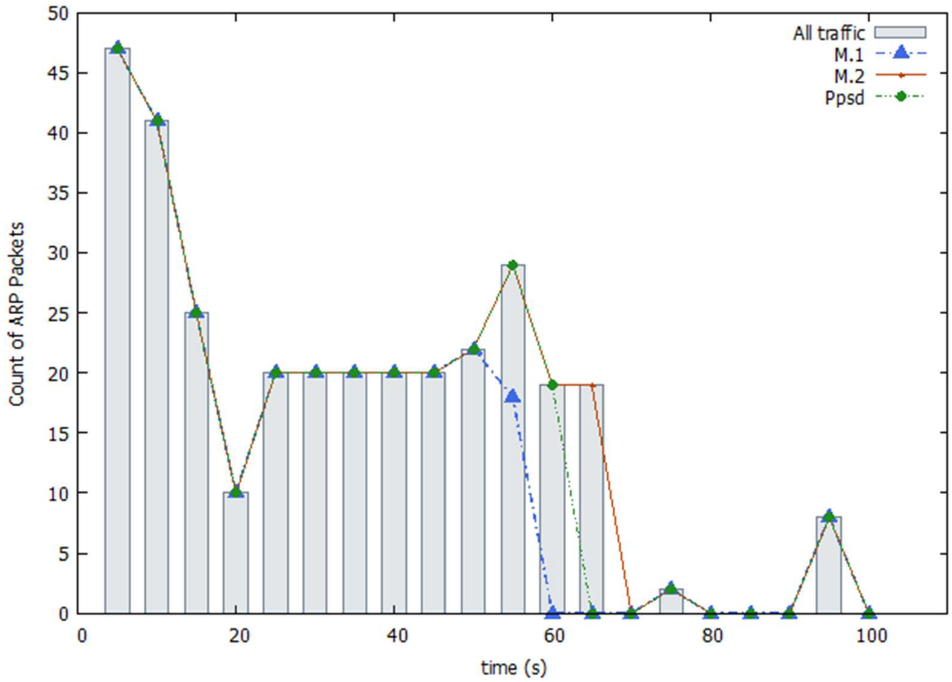

Figure 4.11: Comparison result of Mode-3 scenario.

4.6 Concluding Remarks

ARP spoofing can occur when an attacker presents itself as a legitimate node in a local network and starts to learn the topology. The ARP protocol lacks authentication, and the destination node updates its ARP cache table from any ARP packet received without verification of sender identity. This vulnerability allows the attacker to adopt different ARP packets to launch an attack. In this model, the attacker sends ARP packets into the local network. An entropy value below the threshold of 1.35 shows an abnormal surge in ARP activity and is classified as a critical timeslot. In this model, there are three modes of attack, each representing the different use of an ARP packet to attack the destination node's ARP cache, either sending a request packet or replying to a legitimate request with a spoofed packet. In the scenarios where victims participated in the ARP process, this can lead to a higher entropy with a high volume of ARP packets from the attacker. On the other hand, benign or normal traffic from multiple local nodes was highly distributed, and entropy reached above the predefined value. The gap shows the difference in the attack compared to the regular traffic. In the critical timeslot, the algorithm stored the suspicious MAC and identified duplicate IP or a mismatch of IP-MAC mapping in the previous or adjacent critical time window. The protection filtered and blocked the attacker's MAC in the local network traffic. The proposed method allows the detection of ARP spoofing with scalability and timely warming and is relatively easy to implement. The disadvantages of implementing other methods, such as a Cisco security port to confront ARP spoofing, are high installation cost, some ports may be extended beyond capacity and need more resources to implement and difficulty in the configuration process to cope with cybersecurity threats.

CHAPTER 5

IP FRAGMENTATION ATTACK

5.1 Simulation Model

The fragmented IPv4 and IPv6 traffic are vulnerable to attacks such as DoS. Nevertheless, fragmentation cannot be avoided even in the IPv6 environment. Commonly, IoT sends small data in communication. The data must be fragmented to pass from a source to a destination. IoT traffic usually consists of outbound data transmission consisting of a few bytes per packet. There are situations where the server's communication has larger data, such as firmware updates, upgrades, or patches.

The scenario describes a firmware update process that pushes from the server to the IoT device. The IoT has been reprogrammed OTA to allow firmware updates or bug fixes.

Firmware update on IoT with OTA programming possesses many challenges that can impact the quality of the patching process. Based on Figure 5.1, fragmentation happens when the server sends 4kB of firmware in the network to the destination sensor or IoT device. The data pass through a gateway before reaching the destination IoT. The communication is transmitted in a WiFi signal in an IoT network. In the simulation of an IoT network, the data are sent in small data sizes, commonly found in IoT communication. A sensor node can represent an IoT node. It assumes that the attacker is adjacent to the IoT network sensor and has acquired all the details necessary to conduct an IP fragmentation attack. The details include Identification ID, source IP address, destination IP address, and transport protocol.

Regarding the timing to launch the attack, the attacker is assumed to know when the server will push the firmware update to the victim's IoT device. The server initially sends a copy of the firmware to update the destination IoT. The data undergo fragmentation before reaching the destination IoT. Meanwhile, the attacker creates a fake IP fragment representing a similar firmware packet. With this fragment, the attacker sets the ‘More Fragment’ (MF) flag equal to false. The victim node receives this fragment before any original firmware fragments from the server. The fake fragment is treated as incomplete at the victim node and stored in a buffer. The fragments represent the original firmware that reaches the gateway and is ready to transmit in the IoT network. Once the first fragment is received, it immediately integrates with the one currently stored at the victim node because they have similar ID and details representing a similar packet. Both the sender and receiver IP addresses are the same. However, because the fragment is fake and manipulated by the attacker, it cannot represent the correct firmware. As a result, the victim IoT cannot pass the complete packet up to the application level, and the updating process fails. The remaining incoming fragments continue to be received by the victim node. These fragments are stored in the buffer due to the missing first fragment. These fragments are considered incomplete and fill the victim buffer long enough before expiration.

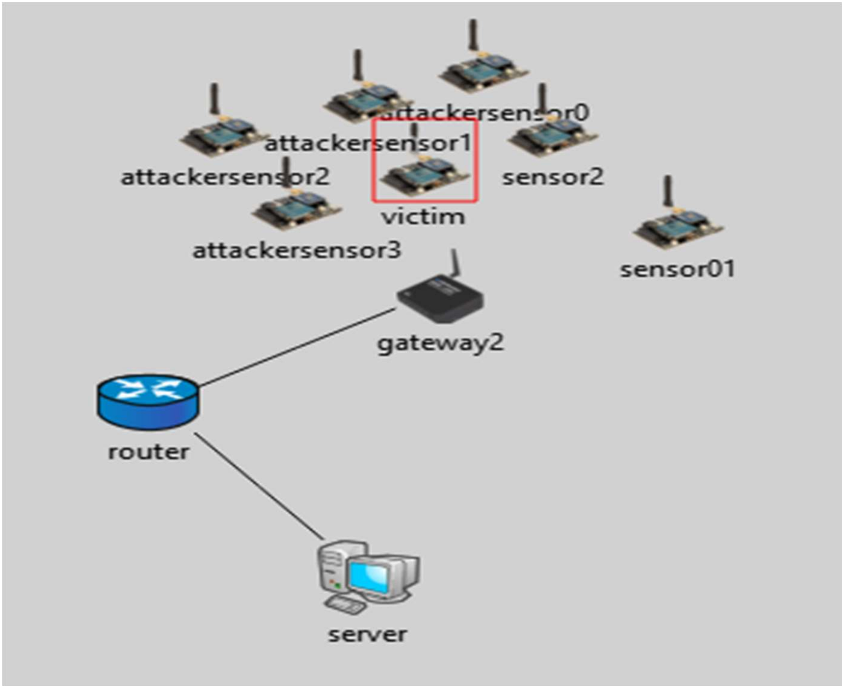

Figure 5.1: Fragmentation attack model (predictable packet header identifier).

5.2 Detection

In fragmentation, the packet is split into multiple fragments at the network layer before being sent into the network. The receiving node reassembled those fragments with similar Identification. Attackers perform the fragmentation attack by sending fragments with same Identification. Figure 5.2 shows that the server sends the original firmware data to the destination. At the destination, the attacker's fragment is reassembled with the original fragments, forming a packet containing 1580 bytes at the transport layer. These fragments are dropped and show 0 bytes received at the application layer of the victim in Figure 5.2. As a result, the fragments' validity can only be verified after the reassembly process and the fragmentation attack is detected.

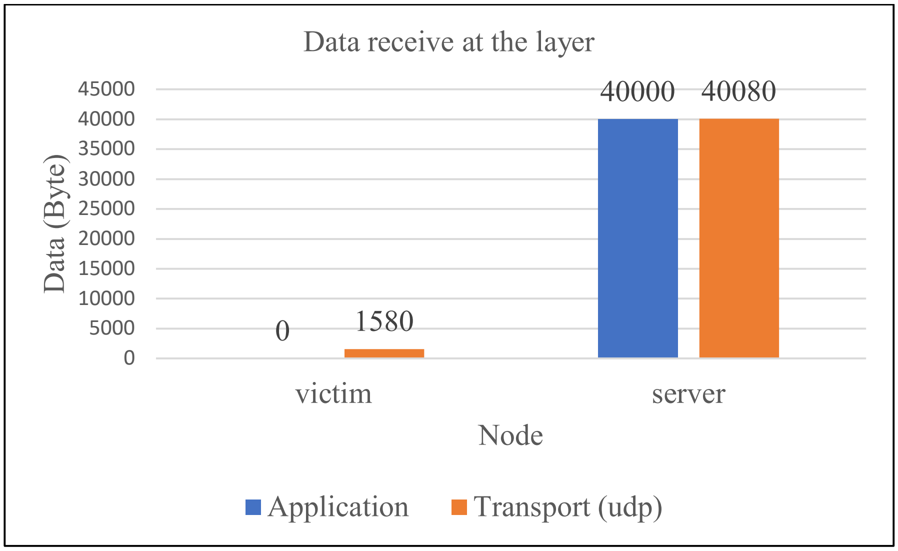

Figure 5.2: Comparison of data reassembly for victim and server.

5.3 Verification Plan

In the simulation run for 100s, where four attackers send a total of 40 fragments, and the server sends 10 times of firmware packets. Figure 5.3 shows the victim IoT received all the fragments sent from gateway2 and attackers. In Figure 5.4, victim IoT does not complete the reassembly process, and no data are received at the application layer. Thus, this model explains the fragmentation attack based on the misassociation of fragmentation caused by a predictable packet header identifier.

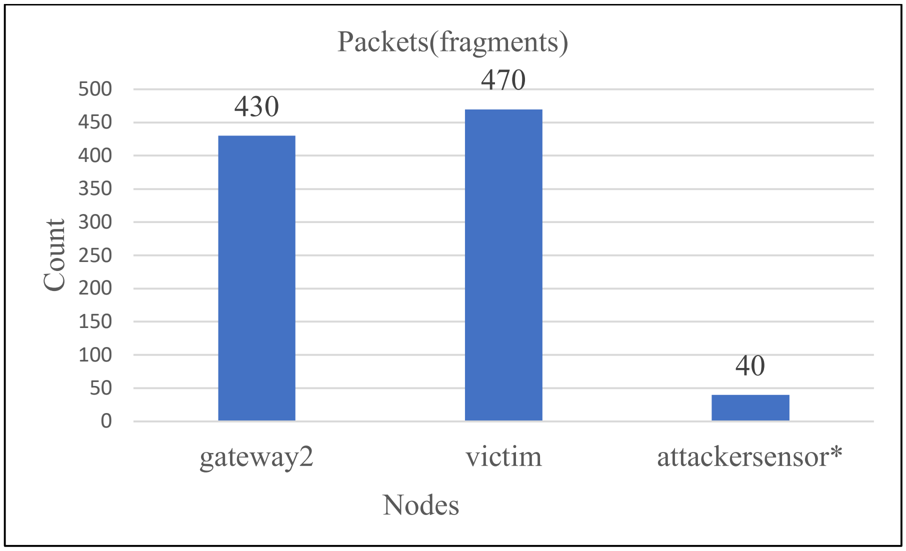

Figure 5.3: Fragments involves in each node (send/receive).

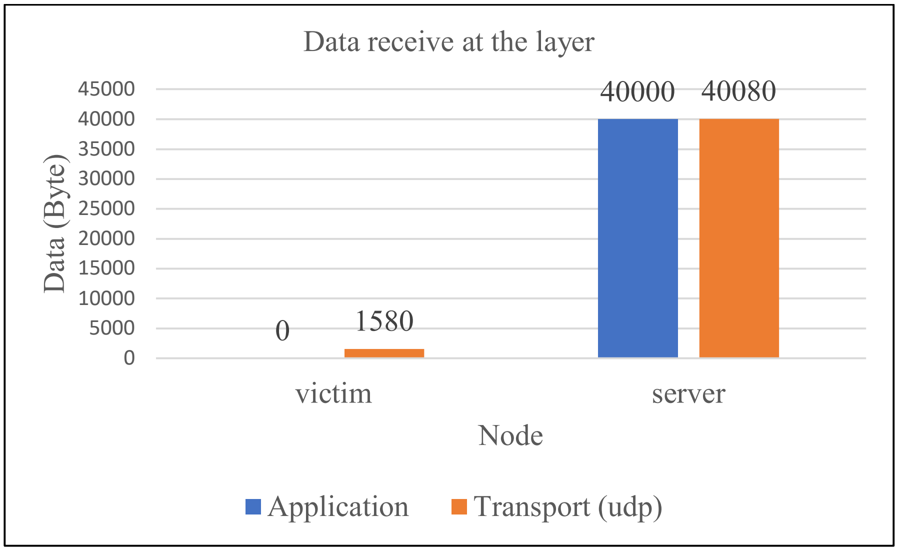

Figure 5.4: (Duplicate) Comparison of data reassembly for victim and server.

5.4 Protection

Fragments are vulnerable. When the firmware is fragmented and transmitted to a low-power network of IoT, it is subject to limited computation and memory. Packet header identifier can be guessed by carefully formatting fake fragments and flooding the target host. Once an Identifier is exposed, an attacker can launch a further attack on the target victim. The fake fragment should not exist in normal traffic. The attackers send fake fragments with the same packet header identifier but with incorrect offset or other validation fields. When the attacker's fragments merged with some of the fragments, this led to the wrong reassembly, and therefore the correct packet could not be formed and passed to the application layer at the destination node. The process is interrupted, and no complete data are received at the node.

The proposed mechanism is adding a One-time code (OTC) and timestamp in the fragment to prevent a predictable packet header identifier and protect against the IP fragmentation attack. The OTC and timestamp consume a maximum of 40 bytes in the option fields. The OTC is a randomly generated number parallel to the timestamp. When extensive data are sent into the networks with different MTUs to deliver to the destination IoT, the packet is split into smaller data called fragments. The first fragment will consist of a high layer of packet information, such as protocol. With the protection implemented, OTC and timestamp with the Identification uniquely represent fragments from a packet, increasing the difficulty for attackers to spot the correct values. In this case, the attacker cannot force the reassembly of the fragments with a lack of complete details that only depend on the source IP, destination IP, packet header identifier, and protocols. Figure 5.5 shows the implementation of OTC and timestamp.

Type of service (TOS) or Differentiated Service Code Point (DSCP) specifies the different services like Voice over IP (VOIP). If the Explicit Congestion

Notification (ECN) is set, it allows notification of network congestion between endpoints to prevent packet drop. The Don't Fragment (DF) field allows the discovery of path MTU and prevents fragmentation between communication nodes. In fragmentation, all the fragments except the last one will have the MF field set to true, indicating more fragments are coming.

Figure 5.5 shows the implementation of OTC and timestamp.

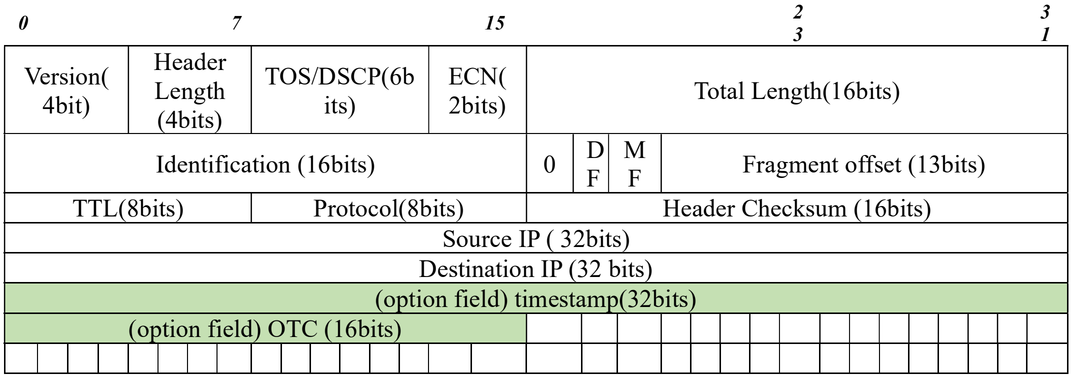

Figure 5.5: Proposed IPv4 header to include timestamp and OTC as enhancement for IP fragmentation attack (misassociation) protection.

5.5 Result

Table 5.1: IP fragmentation attack parameters in the simulation.

Parameter	Value
No. target IoT	1
No. of attacker	4
Total Cycle of simulation/length	40 cycle/400s
Data/cycle (Firmware size)	4kB
Protection	OTC and timestamp

The simulation with configuration in Table 5.1 with the enhanced protection implemented using timestamp and OTC to confront the fragmentation attack caused by fragment misassociation with predictable packet header ID.

Fragments with similar IP-ID, source IP, destination IP, and protocol belong to the same packet. The attacker launches the attack by sending the fragment with the same identity. As protection, the timestamp represents a log message marking the time of the occurrence of an event, and the OTC is a random number. Both attributes are attached to the packet header. Throughout the simulation, attackers attempt to cause the misassociation of fragments from the original packet initiated from the server and routes from the gateway to the IoT target. The attackers target the IoT by sending the formulated and crafted second fragment in each cycle to reassemble with the original first fragment routes from the gateway.

In order to implement the protection, some files with extension .h and .cc in the INET framework have been modified to generate the values of random number OTC and timestamp, passing these values as parameters in each of the functions and, finally, verification of the OTC and timestamp before reassembly of the fragments. The file named Ipv4FragBuf.h, which consists of a Boolean operator to match the key for a datagram in the reassembly buffer, is modified to accept timestamp and OTC and used in the verification.

Based on Figure 5.6, the destination IoT (target) received all the fragments sent from gateway2 and attackers. Figure 5.7 shows all the 160 forge fragments from the attacker were captured at the target without impacting the reassembly process. The OTC and timestamp create difficulty for attackers to guess the correct value of each field and protect the reassembly process at a destination node.

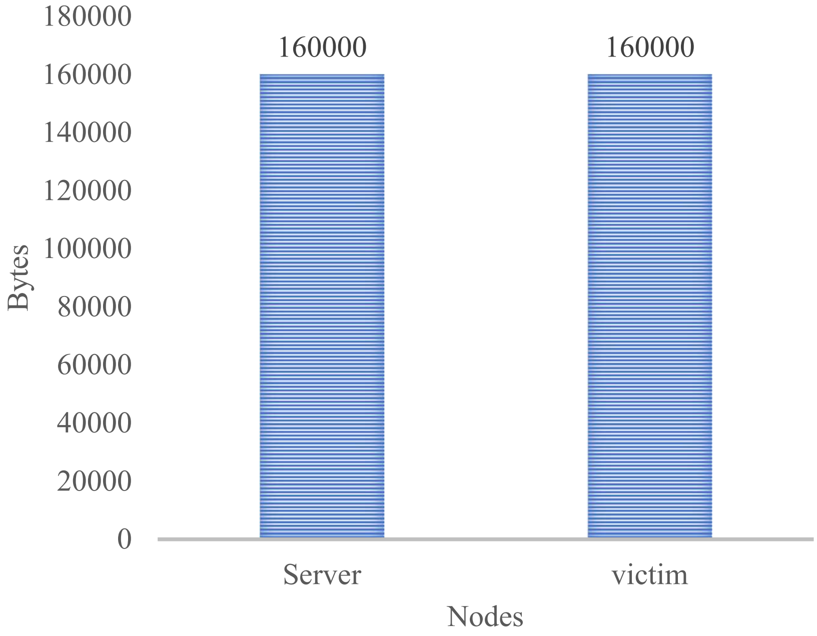

Figure 5.6: Data shown in the application layer is sent from the server to the victim.

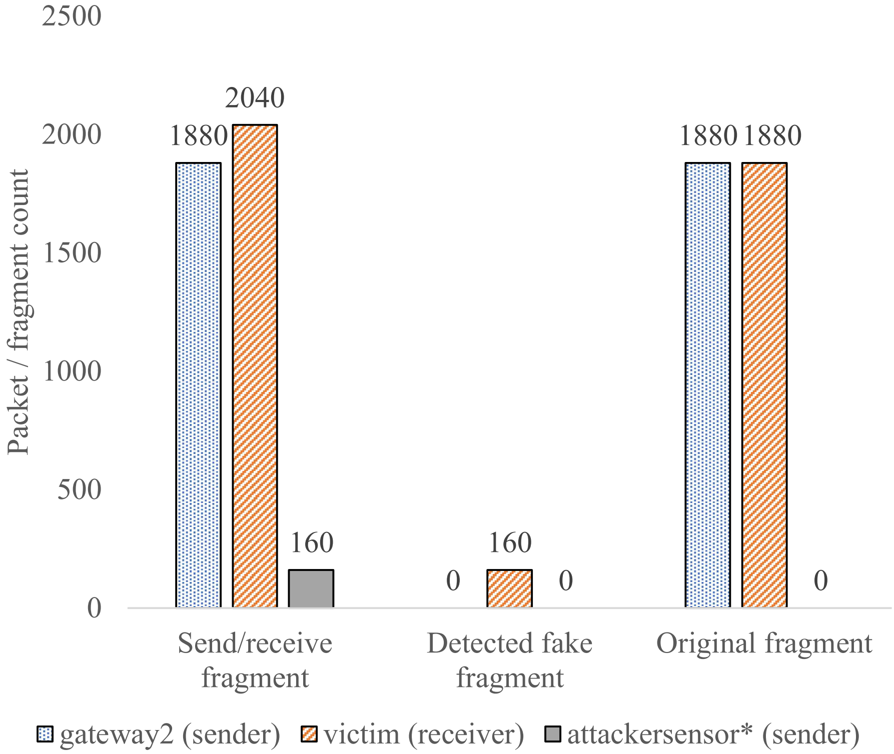

Figure 5.7: Fragment count and statuses.

5.6 Concluding Remarks

Fragmentation can be vulnerable. The huge size of a fragmented packet sent to the destination node allows IoT, which usually handles a few bytes of data, to receive the patching automatically and minimise human intervention. In the fragmentation attack model, the vulnerability depends on the predictability of the packet header identifier, which the OS implements in a simpler way.

In the case of firmware update, once an attacker successfully interrupts the process and if there is no proper handling of the firmware rollback, it can lead to a security breach. The predefined packet header identifier needs protection to prevent an attacker from being easily spotted and interrupting the reassembly process with a similar fragment. Two additional fields are attached to the packet header to protect packet header identifier. A randomly generated number of OTC, which the attacker does not easily spot. A timestamp attached with OTC provides additional detail of protection of the same group of fragments that, if a similar code is generated out of the time scope, is considered expiry, and will be rejected and treated as forged fragments. A protection simulation runs in the 400s where four attackers sent fake fragments out of 40 cycles. The test serves the objective that the protection can work on a high volume of the fragments sent into the network, such as firmware updates. None successfully interrupted the reassembly process at the destination node. Firmware was sent out a total of 40 times and was all received correctly at the destination node.

CHAPTER 6

CONCLUSION

6.1 Conclusion

In this work, the Smart Factory security is studied and focuses on the selected yet common attacks that target critical infrastructure such as CPS. Those attacks are DDoS, ARP spoofing, and IP fragmentation attacks. This work highlights the problems faced by the Smart Factory with IoT implementation. There exists a security threat in the current network and Internet environment. In the IoT network, which is highly interconnected in the heterogeneous environment, the number is still increasing, creating an even more complex situation. IoT is classified as a resource constraint with specific characteristics unsuitable for implementing complex protection that targets the network. The data collected from the IoT are presented in a different format and often require control devices or gateway to perform inaccurate conversions with missing values. Data inconsistency can lead to difficulty in identifying the potential attack traffics and cause inaccurate prediction if a huge portion of corrupted data exist with irrelevant attributes in the datasets. The communication of the IoT with various network devices such as servers or industrial Personal Computer adopts standard communication protocols, and the IP network continues to dominate the networking. Standard communication protocol vulnerabilities allow the attackers to adopt different methods to penetrate the Smart Factory system.

Based on the problem defined, this work aims to present the detection methods currently used to identify the selected attack. Secondly, protection is implemented in the system to recover the network from the attack situation. Third, the protection scheme can handle the network requirements and application by blocking the attack traffic and restoring the network to a normal state.

To meet the objectives, the research questions are answered in this work.

1. The selected attacks are models in the OMNeT++ with the INET framework. This tool simulates the network traffics consisting of the benign and attack followed by implementation of the detection. First, in DDoS, the classifiers are trained with a dataset split into 75 percent of the training set and 35 percent of the testing set. The convergence model solves the DDoS classification problem. Second, for ARP spoofing, the IP-MAC mismatch can detect the attacker’s ARP packet with the entropy flow-based batch processing. Third, a fragmentation attack is detected when the reassembly process of the attack fragment and normal fragment happens in the context. The corrupted packet is dropped and will not be received at higher layers, such as the application layer at the destination.
2. The protection for DDoS and ARP spoofing is to filter and block the attack traffic when the detection shows a positive signal. The changes reflect in the network traffic and restore the network after recovery. In a fragmentation attack, the additional fields are encapsulated in the packet header, thus preventing the packet header identifier from being spotted. As a result, all

the fragments successfully reassembly to the application layer at the destination node.

3. In the evaluation of effectiveness, DDoS is evaluated with the benchmark of predefined variables of F1-score, accuracy, recall and precision. The proposed RFPCA achieves the highest F1 and accuracy with a high recall rate to capture the attack traffic. In ARP spoofing, the proposed method recovers the network traffic by capturing attack traffic based on three scenarios. In this case, the entropy shows the ability to scale and identify the abnormality in the traffic due to the surge of the traffic volume in a particular time window. In protecting fragmentation, the model is simulated in 40 cycles with four attackers sending fake fragments to attack IP fragmentation and the reassembly process. All the packets are successfully reassembly, and thus none of the attackers is successful in the attack.

The research highlights potential vulnerabilities in the network, either the vulnerability of IoT or the one that exists in common communication protocols. Further study in each of the selected attack and contributed to the effect of each detection. The detections are shown, and the protections are implemented in the network. The effect of the protections are evaluated and presented in the recovery of the network traffic. Lastly, based on the proposed protections are implemented in the network. Based on the identified vulnerability in the network and protocols, the protection can then look for enhancements to solve the security issue that exists in the Smart Factory environment.

6.2 Future Work

This paper presents the detections and protections for the selected attacks: DDoS, ARP spoofing and IP fragmentation attacks. All the models are developed and run on the simulation tool, which is OMNeT++ with the INET framework. The data collected are constrained and tested on simulation. Currently, a limited number of datasets related to IoT communication are available as open source. Secondly, the data collected are limited, with limited scenarios covering attacks.

In the future, an experiment can be conducted for the attack. The experiment includes setting up the components representing the IoT network and collecting the data while executing the components for a long time. More scenarios and attributes in IoT communication can be covered and a significant dataset can be collected. As the dataset grows, this leads to further challenges, such as data requiring a compelling pre-processing method. Also, the IoT may contain considerable noise and irrelevant data that must be handled effectively. In terms of detection and protection, the research uses a limited number of classifiers involved in the performance of comparison prediction. With more significant and more attributes present in a dataset, an intelligent algorithm can be considered for developing more reliable, adaptable and scalable protection.

The selected security aspect has been studied based on models generated in simulation. This work can serve as a reference for implementing the security protection measure for the identified attacks.

REFERENCES

[1] D. Sinha and R. Roy, “Reviewing cyber-physical system as a part of smart factory in industry 4.0,” *IEEE Engineering Management Review*, vol. 48, no. 2, pp. 103–117, 2020, doi: 10.1109/EMR.2020.2992606.

[2] N. Tuptuk and S. Hailes, “Security of smart manufacturing systems,” *Journal of Manufacturing Systems*, vol. 47, pp. 93–106, 2018, doi: 10.1016/j.jmsy.2018.04.007.

[3] M. Lohstroh, H. Kim, J. C. Eidson, C. Jerad, and B. Osyk *et al.*, “On enabling technologies for the internet of important things,” *IEEE Access*, vol. 7, pp. 27244–27256, 2019, doi: 10.1109/ACCESS.2019.2901509.

[4] G. Hernandez, F. Fowze, D. J. Tang, T. Yavuz, P. Traynor, and K. R. B. Butler, “Toward automated firmware analysis in the IoT era,” *IEEE Security and Privacy*, vol. 17, no. 5, pp. 38–46, Sep. 2019, doi: 10.1109/MSEC.2019.2926462.

[5] Y. bin Zikria, S. W. Kim, O. Hahm, M. K. Afzal, and M. Y. Aalsalem, “Internet of things (IoT) operating systems management: Opportunities, challenges, and solution,” *Sensors*, vol. 19, no. 8, Apr. 2019, doi: 10.3390/s19081793.

[6] T. Nguyen, R. G. Gosine, and P. Warrian, “A systematic review of big data analytics for oil and gas industry 4.0,” *IEEE Access*, vol. 8, pp. 61183–61201, 2020. doi: 10.1109/ACCESS.2020.2979678.

[7] P. Mahesh, A. Tiwari, C. Jin, P. R. Kumar, A. L. N. Reddy *et al.*, “A survey of cybersecurity of digital manufacturing,” in *Proceedings of the IEEE*, vol. 109, no. 4, pp. 495–516, Apr. 2021. doi: 10.1109/JPROC.2020.3032074.

[8] K. D. Thoben, S. A. Wiesner, and T. Wuest, “‘Industrie 4.0’ and smart manufacturing-A review of research issues and application examples,” *International Journal of Automation Technology*, vol. 11, no. 1, pp. 4–16, 2017. doi: 10.20965/ijat.2017.p0004.

[9] M. Kocakulak and I. Butun, “An overview of wireless sensor networks towards internet of things,” in *Proceedings IEEE 7th Annual Computing and Communication. Workshop and Conference (CCWC)*, Jan. 2017,pp. 1-6. doi: 10.1109/CCWC.2017.7868374.

[10] Canadian Centre for Cyber Security, “Cyber threat bulletin the cyber threat to operational technology.” Government of Canada, Dec. 2021. [Online]. Available: https://cyber.gc.ca/en/guidance/cyber-threat-bulletin-cyber-threat-operational-technology.

[11] F. Meneghello, M. Calore, D. Zucchetto, M. Polese, and A. Zanella, “IoT: Internet of threats? a survey of practical security vulnerabilities in real IoT devices,” *IEEE Internet Things Journal*, vol. 6, no. 5, pp. 8182–8201, Oct. 2019, doi: 10.1109/JIOT.2019.2935189.

[12] B. B. Zarpelao, R. S. Miani, C. T. Kawakani, and S. C. de Alvarenga, “A survey of intrusion detection in internet of things,” *Journal of Network and Computer Applications*, vol. 84, pp. 25–37, Apr. 2017, doi: 10.1016/j.jnca.2017.02.009.

[13] B. Wang, X. Li, L. P. de Aguiar, D. S. Menasche, and Z. Shafiq, “Characterizing and modeling patching practices of industrial control systems,” in *Proceedings of the ACM Measurement and Analysis of*

Computing Systems, Jun. 2017, vol. 1, no. 1, pp. 1–23, doi: 10.1145/3084455.

[14] N. Tuptuk, P. Hazell, J. Watson, and S. Hailes, “A systematic review of the state of cyber-security in water systems,” *Water*, vol. 13, no. 1, Jan. 01, 2021. doi: 10.3390/w13010081.

[15] D. Sakhawat, A. N. Khan, M. Aslam, and A. T. Chronopoulos, “Agent-based ARP cache poisoning detection in switched LAN environments,” *IET Networks*, vol. 8, no. 1, pp. 67–73, Jan. 2019, doi: 10.1049/iet-net.2018.5084.

[16] S. Mantravadi, R. Schnyder, C. Moller, and T. D. Brunoe, “Securing IT/OT links for low power IIoT devices: Design considerations for industry 4.0,” *IEEE Access*, vol. 8, pp. 200305–200321, 2020, doi: 10.1109/ACCESS.2020.3035963.

[17] K. Patel and H. Upadhyay, “A rule based approach to mitigate DDoS attack in IoT environment,” vol. 4, no. 3, p. 2018. [Online]. Available: http://ijariie.com/AdminUploadPdf/A_Rule_based_Approach_to_Mitigate_DDoS_attack_in_IoT_Environment_ijariie8383.pdf.

[18] A. J. Lathrop and J. M. Stanisz, “Hackers are after more than just data: will your company’s property policies respond when cyber attacks cause physical damage and shut down operations?,” *Environmental Claims Journal*, vol. 28, no. 4, pp. 286–303, Oct. 2016, doi: 10.1080/10406026.2016.1197653.

[19] E. Staddon, V. Loscri, and N. Mitton, “Attack categorisation for IoT applications in critical infrastructures, a survey,” Applied Science, vol. 11, no. 16, 2021, doi: 10.3390/app11167228.

[20] L. Costa, J. P. Barros, and M. Tavares, “Vulnerabilities in IoT devices for smart home environment,” in *Proceedings of the 5th International Conference on Information Systems Security and Privacy (ICISSP)*, 2019, pp. 615–622. doi: 10.5220/0007583306150622.

[21] T. Alladi, V. Chamola, B. Sikdar, and K. R. Choo, “Consumer IoT: Security vulnerability case studies and solutions.” *IEEE Comsumer Electronics Magazine*, vol. 9, no. 2, pp. 17-25, Mar. 2020.

[22] Y. Seralathan, T. T. Oh, S. Jadhav, J. Myers, J. P. Jeong *et al.*, “IoT security vulnerability: A case study of a web camera,” in *2018 20th International Conference on Advanced Communication Technology (ICACT)*. 2018, pp. 172–177. doi: 10.23919/ICACT.2018.8323686.

[23] B. Bajic, A. Rikalovic, N. Suzic, and V. Piuri, “Industry 4.0 implementation challenges and opportunities: a managerial perspective,” *IEEE Systems Journal*, vol. 15, no. 1, pp. 546–559, Mar. 2021, doi: 10.1109/JSYST.2020.3023041.

[24] N. Neshenko, E. Bou-Harb, J. Crichigno, G. Kaddoum, and N. Ghani, “Demystifying IoT security: An exhaustive survey on IoT vulnerabilities and a first empirical look on internet-scale IoT exploitations,” *IEEE Communications Surveys and Tutorials*, vol. 21, no. 3, pp. 2702–2733, 3rd Quart., 2019, doi: 10.1109/COMST.2019.2910750.

[25] X. Yao, J. Zhou, Y. Lin, Y. Li, H. Yu, and Y. Liu, “Smart manufacturing based on cyber-physical systems and beyond,” Journal of Intelligent Manufacturing, vol. 30, pp. 2805–2817, 2019, doi: 10.1007/s10845-017-1384-5.

[26] M. Antonakakis, T. April, M. Bailey, M. Barnhard, E. Bursztein *et al.*,

"Understanding the mirai botnet," in *Proceedings of the 26th USENIX on Security Symposium*, Aug. 2017, pp. 1093–1110.

[27] V. Alcacer and V. Cruz-Machado, "Scanning the industry 4.0: A literature review on technologies for manufacturing systems," *Engineering Science and Technology, an International Journal*, vol. 22, no. 3, pp. 899–919, 2019, doi: 10.1016/j.jestch.2019.01.006.

[28] V. Mullet, P. Sondi, and E. Ramat, "A review of cybersecurity guidelines for manufacturing factories in industry 4.0," *IEEE Access*, vol. 9. Institute of Electrical and Electronics Engineers Inc., pp. 23235–23263, 2021. doi: 10.1109/ACCESS.2021.3056650.

[29] B. Chen, J. Wan, L. Shu, P. Li, M. Mukherjee *et al.*, "Smart factory of industry 4.0: Key technologies, application case, and challenges," *IEEE Access*, vol. 6, pp. 6505–6519, 2017, doi: 10.1109/ACCESS.2017.2783682.

[30] G. Murray, M. N. Johnstone, and C. Valli, "The convergence of IT and OT in critical infrastructure," in *Proceedings of the 15th Australian Information Security Management Conference (AISM)*, 2017, pp. 149–155, doi: 10.4225/75/5A84F7B595B4E.

[31] J. Sachs and K. Landernas, "Review of 5G capabilities for smart manufacturing," in *2021 17th International Symposium on Wireless Communication Systems (ISWCS)*, 2021, pp. 1-6. doi: 10.1109/ISWCS49558.2021.9562173.

[32] A. U. Mentsiev, E. R. Guzueva, and T. R. Magomaev, "Security challenges of the industry 4.0," in *Journal of Physics: Conference Series*, Apr. 2020, vol. 1515, no. 3, doi: 10.1088/1742-6596/1515/3/032074.

[33] J. D. Adriano, E. C. do Rosario, and J. J. P. C. Rodrigues, "Wireless sensor networks in industry 4.0: WirelessHART and ISA100.11a," in *Proceedings 13th IEEE International Conference on Industry Applications (INDUSCON)*, Nov. 2018, pp. 924–929. doi: 10.1109/INDUSCON.2018.8627177.

[34] X. Zhou, Z. Xu, L. Wang, K. Chen, C. Chen *et al.*, "APT attack analysis in SCADA systems," in *MATEC Web of Conferences*, 2018, vol. 173, no. 3, pp. 1–5, doi: 10.1051/matecconf/201817301010.

[35] D. H. Shin, G. Y. Kim, and I. C. Euom, "Vulnerabilities of the open platform communication unified architecture protocol in industrial internet of things operation," *Sensors*, vol. 22, no. 17, Sep. 2022, doi: 10.3390/s22176575.

[36] J. Prinsloo, S. Sinha, and B. V. Solms, "A review of industry 4.0 manufacturing process security risks," *Applied Sciences*, vol. 9, no. 23, Nov. 2019, doi: 10.3390/app9235105.

[37] M. Tavana, V. Hajipour, and S. Oveisi, "IoT-based enterprise resource planning: Challenges, open issues, applications, architecture, and future research directions," *Internet of Things*, vol. 11, Sep. 2020, doi: 10.1016/j.iot.2020.100262.

[38] F.-N. Yang and H.-Y. Lin, "Development of a predictive maintenance platform for cyber-physical systems," in *2019 IEEE International Conference on Industrial Cyber Physical Systems (ICPS)*, 2019, pp. 331-335.

[39] S. A. Bello, L. O. Oyedele, O. O. Akinade, M. Bilal, J. N. D. Delgado *et al.*, "Cloud computing in construction industry: Use cases, benefits and

challenges," *Automation in Construction*, vol. 122, Feb. 2021, doi: 10.1016/j.autcon.2020.103441.

[40] D. J. Ahn and J. Jeong, "A bigdata search engine based cloud computing network architecture in smart factory environment," in *Proceedings - 2019 2nd International Conference on Intelligent Autonomous Systems (ICoIAS),* 2019, pp. 91–95, doi: 10.1109/ICoIAS.2019.00023.

[41] M. Vidosav, S. Stojadinovic, B. Lalic and U. Marjanovic, "ERP in industry 4.0 context," in *IFIP Advances in Information and Communication Technology*, 2020, vol. 591 pp. 287–294. doi: 10.1007/978-3-030-57993-7_33.

[42] F. Tao, Q. Qi, L. Wang, and A. Y. C. Nee, "Digital twins and cyber–physical systems toward smart manufacturing and industry 4.0: Correlation and comparison," *Engineering*, vol. 5, no. 4, pp. 653–661, Aug. 2019, doi: 10.1016/j.eng.2019.01.014.

[43] D. A. Zakoldaev, A. v. Gurjanov, A. v. Shukalov, I. O. Zharinov, and O. O. Zharinov, "Cyber and physical systems topology for the industry 4.0 smart factory," in *IOP Conference Series: Materials Science and Engineering*, 2019, vol. 582, doi: 10.1088/1757-899X/582/1/012010.

[44] N. A. Shinwari, Nasrullah, A. Saxena, and N. Sharma, "Vendor lock-in situation in cloud computing," *International Journal of Creative Research Thoughts (IJCRT)*, 2018. [Online]. Available: https://www.ijcrt.org/papers/IJCRT1813411.pdf.

[45] A. Munsch and P. Munsch, "The future of API (application programming interface) security: The adoption of APIs for digital communications and the implications for cyber security vulnerabilities," *Journal of International Technology and Information Management*, vol. 29, no. 3, pp. 24–45, Jan. 2021, doi: 10.58729/1941-6679.1454.

[46] J. Hassan, D. Shehzad, U. Habib, M. U. Aftab, M. Ahmad *et al.*, "The Rise of cloud computing: Data protection, privacy, and open research challenges - A systematic literature review (SLR)," *Computational Intelligence and Neuroscience*, 2022, doi: 10.1155/2022/8303504.

[47] S. Parikh, D. Dave, R. Patel, and N. Doshi, "Security and privacy issues in cloud, fog and edge computing," *Procedia Computer Science*, 2019, vol. 160, pp. 734–739. doi: 10.1016/j.procs.2019.11.018.

[48] S. Khan, S. Parkinson, and Y. Qin, "Fog computing security: A review of current applications and security solutions," *Journal of Cloud Computing*, vol. 6, 2017. doi: 10.1186/s13677-017-0090-3.

[49] D. Roca, J. v. Quiroga, M. Valero, and M. Nemirovsky, "Fog function virtualization: A flexible solution for IoT applications," in *2017 2nd International Conference on Fog and Mobile Edge Computing (FMEC),* May. 2017, pp. 74–80. doi: 10.1109/FMEC.2017.7946411.

[50] Z. Xu, Y. Zhang, H. Li, W. Yang, and Q. Qi, "Dynamic resource provisioning for cyber-physical systems in cloud-fog-edge computing," *Journal of Cloud Computing*, vol. 9, 2020, doi: 10.1186/s13677-020-00181-y.

[51] P. K. Illa and N. Padhi, "Practical guide to smart factory transition using IoT, big data and edge analytics," *IEEE Access*, vol. 6, pp. 55162–55170, 2018, doi: 10.1109/ACCESS.2018.2872799.

[52] J. Protner, M. Pipan, H. Zupan, M. Resman, M. Simic *et al.*, "Edge computing and digital twin based smart manufacturing," *IFAC-*

PapersOnLine, 2021, vol. 54, pp. 831–836. doi: 10.1016/j.ifacol.2021.08.098.

[53] R.-H. Hsu, J. Lee, T. Q. S. Quek, and J.-C. Chen, “Reconfigurable security: Edge computing-based framework for IoT,” *IEEE Network*, vol. 32, no. 5, pp. 92-95, Sep. 2018.

[54] N. Kherraf, H. A. Alameddine, S. Sharafeddine, C. M. Assi, and A. Ghrayeb, “Optimized provisioning of edge computing resources with heterogeneous workload in IoT networks,” *IEEE Transactions on Network and Service Management*, vol. 16, no. 2, pp. 459–474, Jun. 2019, doi: 10.1109/TNSM.2019.2894955.

[55] P. P. Ray, D. Dash, and D. De, “Edge computing for internet of Things: A survey, e-healthcare case study and future direction,” *Journal of Network and Computer Applications*, vol. 140, pp. 1–22, Aug. 2019, doi: 10.1016/j.jnca.2019.05.005.

[56] S. Millar, “IoT security challenges and mitigations: An introduction,”, arXiv:2112.14618 [cs.CR], Dec. 2021.

[57] M. Yu, J. Zhuge, M. Cao, Z. Shi, and L. Jiang, “A survey of security vulnerability analysis, discovery, detection, and mitigation on IoT devices,” *Future Internet*, vol. 12, no. 2, Feb. 2020, doi: 10.3390/fi12020027.

[58] A. Antony and S. S., “A review on IoT operating systems,” *International Journal of Computer Applications*, vol. 176, no. 24, pp. 33–40, May. 2020, doi: 10.5120/ijca2020920245.

[59] L. Tawalbeh, F. Muheidat, M. Tawalbeh, and M. Quwaider, “IoT privacy and security: challenges and solutions,” *Applied Sciences*, vol. 10, no. 12, pp. 1–17, 2020, doi: 10.3390/APP10124102.

[60] K. Vikas, “IoT security-challenges & best practices,” 2020. Accessed: Feb. 26, 2023. [Online]. Available: https://www.happiestminds.com/wp-content/uploads/2020/12/IoT-Security-Challenges-and-Best-Practices.pdf.

[61] T. Abdelghani, “Implementation of defense in depth strategy to secure industrial control system in critical infrastructures,” *American Journal of Artificial Intelligence*, vol. 3, no. 2, p. 17, 2019, doi: 10.11648/j.ajai.20190302.11.

[62] IoT Security Foundation, *Router and IoT Vulnerabilities : Insecure by Design*, Aug. 2021, [Online]. Available: https://www.iotsecurityfoundation.org/wp-content/uploads/2021/08/ManySecured-SUIB-White-Paper.pdf.

[63] B. Cusack and F. Zhuang, “Vulnerability analysis : Protecting information in the IoT,” in *Proceedings of the 16th Australian Information Security Management Conference (AISM), 2017,* pp. 74–82, 2018, doi: 10.25958/5c526da166689.

[64] J. Wu, Y. Nan, V. Kumar, M. Payer, and D. Xu, “Blueshield: Detecting spoofing attacks in bluetooth low energy networks.” in *23rd International Symposium on Research in Attacks, Instrusions and Defences (RAID 2020)*, Oct. 2020, pp. 397-411, USENIX Association.

[65] I. Butun, P. Osterberg, and H. Song, “Security of the internet of things: vulnerabilities, attacks, and countermeasures,” *IEEE Communications Surveys and Tutorials*, vol. 22, no. 1, pp. 616–644, 2020, doi: 10.1109/COMST.2019.2953364.

[66] G. Nebbione and M. C. Calzarossa, "Security of IoT application layer protocols: Challenges and findings," *Future Internet*, vol. 12, no. 3, pp. 1–20, 2020, doi: 10.3390/fi12030055.

[67] D. Bhamare, M. Zolanvari, A. Erbad, R. Jain, K. Khan *et al.*, "Cybersecurity for industrial control systems: A survey." *Computer and security*, 2019, doi: 10.1016/j.cose.2019.101677.

[68] F. Maggi, M. Balduzzi, R. Vosseler, M. Rosler, W. Quadrini *et al.*, "Smart factory security: A case study on a modular smart manufacturing system," *Procedia Computer Science*, 2021, vol. 180, pp. 666–675. doi: 10.1016/j.procs.2021.01.289.

[69] M. Azrour, J. Mabrouki, A. Guezzaz, and A. Kanwal, "Internet of Things Security: Challenges and key Issues," *Security and Communication Networks*, vol. 2021. Hindawi Limited, 2021. doi: 10.1155/2021/5533843.

[70] A. Chehri, I. Fofana, and X. Yang, "Security risk modeling in smart grid critical infrastructures in the era of big data and artificial intelligence," *Sustainability (Switzerland)*, vol. 13, no. 6. MDPI AG, Mar. 02, 2021. doi: 10.3390/su13063196.

[71] T. Lojka, M. Bundzel, and I. Zolotová, "Service-oriented architecture and cloud manufacturing." *Acta Polytechnica Hungariaca*, 2016, vol. 13, no. 6, pp. 25-44, doi: 10.12700/aph.13.6.2016.6.2,

[72] A. Arshad, M. H. Zurina, S. Subramaniam, and R. Latip, "A survey of Sybil attack countermeasures in IoT-based wireless sensor networks," *PeerJ Computer Science*, vol. 7, pp. 1–33, 2021, doi: 10.7717/peerj-cs.673.

[73] F. Li and Y. Tang, "False data injection attack for cyber-physical systems with resource constraint," *IEEE Transactions on Cybernetics*, vol. 50, no. 2, pp. 729–738, 2020, doi: 10.1109/TCYB.2018.2871951.

[74] C. Alcaraz, G. Bernieri, F. Pascucci, J. Lopez, and R. Setola, "Covert channels-based stealth attacks in industry 4.0," *IEEE Systems Journal*, vol. 13, no. 4, pp. 3980–3988, 2019, doi: 10.1109/JSYST.2019.2912308.

[75] J. Li, M. Liu, Z. Xue, X. Fan, and X. He, "Rtvd: A real-time volumetric detection scheme for DDoS in the internet of things," *IEEE Access*, vol. 8, pp. 36191–36201, 2020, doi: 10.1109/ACCESS.2020.2974293.

[76] R. Bonica, F. Baker, G. Huston, R. Hinden, and O. Troan *et al.*, "IP fragmentation considered fragile," BCP 230, RFC 8900, pp. 1–23, Sep. 2020, [Online]. Available: https://www.rfc-editor.org/rfc/rfc8900.pdf.

[77] R. V. Rijswijk-Deij, C. Strotmann, and P. B. Koetter, "IP Fragmentation and measures against DNS cache poisoning (Frag-DNS)," 2022. [Online]. Available: https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Publikationen/Studien/Frag-DNS/Frag-DNS-Studie.pdf?__blob=publicationFile&v=3.

[78] F. Salutari, D. Cicalese, and D. J. Rossi, "A closer look at IP-ID behavior in the Wild," *Internation Conference on Passive and Active Network Measurement,* Springer, 2018, pp. 243-254. doi: 10.1007/978-3-319-76481-8_18.

[79] K. M. Sudar, M. Beulah, P. Deepalakshmi, P. Nagaraj, and P. Chinnasamy, "Detection of distributed denial of service attacks in SDN using machine learning techniques," in *2021 International Conference*

on Computer Communication and Informatics, (ICCCI), 2021, pp. 1-5, doi: 10.1109/ICCCI50826.2021.9402517.

[80] S. Dong and M. Sarem, "DDoS attack detection method based on improved KNN with the degree of DDoS attack in software-defined networks," *IEEE Access*, vol. 8, pp. 5039–5048, 2020, doi: 10.1109/ACCESS.2019.2963077.

[81] J. Pei, Y. Chen, and W. Ji, "A DDoS attack detection method based on machine learning," in *Journal of Physics: Conference Series*, 2019, vol. 1237, no. 3. doi: 10.1088/1742-6596/1237/3/032040.

[82] Z. Shah and S. Cosgrove, "Mitigating ARP cache poisoning attack in Software-Defined Networking (SDN): A Survey," *Electronics (Switzerland)*, vol. 8, no. 10, Oct. 2019, doi: 10.3390/electronics8101095.

[83] W. Gao, Y. Sun, Q. Fu, Z. Wu, X. Ma *et al.*, "ARP poisoning prevention in internet of things," in *Proceedings - 9th International Conference on Information Technology in Medicine and Education (ITME)*, Oct. 2018, pp. 733–736, 2018, doi: 10.1109/ITME.2018.00166.

[84] M. Abid and A. Singh, "ARP spoofing detection via wireshark and veracode," *International Journal of New Technology and Research (IJNTR)*, vol. 4, no. 5, pp. 27-30, 2018.

[85] A. S. Alghawli, "Complex methods detect anomalies in real time based on time series analysis," *Alexandria Engineering Journal*, vol. 61, no. 1, pp. 549–561, 2022, doi: 10.1016/j.aej.2021.06.033.

[86] T. Komazec and S. Gajin, "Analysis of flow-based anomaly detection using shannon's entropy," *2019 27th Telecommunications Forum (TELFOR)*, 2019, pp. 1–4, 2019, doi: 10.1109/TELFOR48224.2019.8971036.

[87] P. Bereziński, B. Jasiul, and M. Szpyrka, "An entropy-based network anomaly detection method," *Entropy*, vol. 17, no. 4, pp. 2367–2408, 2015, doi: 10.3390/e17042367.

[88] T. Dai, H. Shulman, and M. Waidner, "DNS-over-TCP considered vulnerable." In *Proceedings of the Applied Networking Research Workshop (ANRW)* , 2021, pp. 76-81, doi: doi.org/10.1145/3472305.3472884.

[89] N. A. Noureldien and I. M. Yousif, "Accuracy of machine learning algorithms in detecting DoS attacks types," *Science and Technology*, vol. 6, no. 4, pp. 89–92, 2016, doi: 10.5923/j.scit.20160604.01.

[90] A. M. Al Tobi, and I. Duncan, "KDD 1999 generation faults: A review and analysis," *Journal of Cyber Security Technology*, vol. 2, no. 3–4, pp. 164–200, 2018, doi: 10.1080/23742917.2018.1518061.

[91] X. Feng, Q. Li, K. Sun, K. Xu, B. Liu *et al.*, "PMTUD is not panacea: Revisiting IP fragmentation attacks against TCP," *in Proceedings of the Network & Distributed System Security Symposium (NDSS)*, 2022, doi: 10.14722/ndss.2022.24381.

[92] I. Suciu, X. Vilajosana, and F. Adelantado, "An analysis of packet fragmentation impact in LPWAN," *IEEE Wireless Communications and Networking Conference (WCNC)*, 2018, pp. 1–6, 2018, doi: 10.1109/WCNC.2018.8377440.