Fabrice Gatambiye

Automating Firewall Testing Using Network Traffic Simulation Tools

Wärtsilä Finland Oy

School of Technology and Innovations
Master's Thesis in Computing Sciences
Master of Science in Technology

Vaasa 2026

UNIVERSITY OF VAASA
School of Technology and Innovations
Author: Fabrice Gatambiye
Title of the thesis: Automating Firewall Testing Using Network Traffic Simulation Tools
Degree: Master of Computing Sciences
Degree Programme: Master's Programme in Computing Sciences
Supervisor: Janne Koljonen and Timo Mantere
Wärtsilä Supervisor: Mathias Karlå
Year: 2026 **Pages:** 72

ABSTRACT:

As cybersecurity is increasingly emerging as a critical aspect across the technology industry, Wärtsilä engine network systems cyber-resilience is recognized as a significant factor in ensuring the cybersecurity of engines and the engine auxiliaries. As Wärtsilä marine engine network systems rely on firewalls as core defence mechanism against external threats, it is crucial to ensure both the bidirectional traffic between the engine and engine auxiliaries and to prevent access by unauthorized devices that are separated from each other by firewalls.

Thus far, Wärtsilä firewall testing is primarily performed manually. Therefore, the objective of this thesis is to investigate as well as discover tools to automate the firewall testing process. As the objective of this thesis is not to redesign firewall configuration but centres its efforts on discovering a traffic simulation tool combination to simulate end-to-end communication between client and server hosts. This study examines which tools can send and receive network traffic along with which tools are limited to one-way packet generation

In the methodology section of this research the literature review is combined with the experimental work conducted in a lab environment. In the methodology section traffic generation tools are reviewed. Tools which were not found to be effective were subsequently excluded. In addition, a Scapy based answering machine was implemented to enable the solution on the server host side, while Nmap appeared to solve the client host side traffic generation problem.

This thesis contributes to Wärtsilä engines cybersecurity through the approach invented during the experimental work conducted by the author. The automated firewall testing results demonstrate how the simulation tools can minimize manual efforts.

In conclusion, this work delivers a solution to automatically test all ports, which may extend to 60000 per device, in comparison with the preceding tests that were conducted only for ports in use. The automation proposed in this research provides a convenient solution for testing new system updates, such as firmware or configuration updates.

KEYWORDS: Firewall Test Automation, iPerf3, Scapy, Maritime Cyber Security, Port scanning, Answering machine

Acknowledgements

Starting a project with abstract goals, not knowing the possible outcome, yet trusting in its success requires a lot of effort, hard work, and constant critical thinking on the goals set at the beginning. Much of trial-and-error of the author might not be seen in the completed work of six months. Therefore, I wish to highlight the efforts that are not mentioned in this thesis.

The nature of experimental research is both challenging and rewarding. It is crucial to not only appreciate the successful hard work, moreover the hard work that was set aside after realizing that it did not match the research objectives.

I would like to take this opportunity and thank Wärtsilä for the opportunity to collaborate and expand my knowledge in the field of cybersecurity through this research.

It is worthy to honour the individuals who impacted the flow of the research. I would like to express my gratitude to my thesis supervisors Mr. Janne Koljonen and Mr. Timo Mantere for without their professional feedback and support the research objectives would not have been reached.

I wish to extend my special gratitude to my Wärtsilä thesis supervisor, Mr. Mathias Karlå for his instructions and insightful feedback throughout the thesis work. His collaborative way of work had a significant impact in the success achieved during both the literature review and the experimental work.

As this milestone may perhaps mark the end of my academic journey, I would like to express my heartfelt gratitude to my mother for her efforts and sacrifices throughout my life, which have enabled me to attain my educational goals. This achievement is dedicated to her.

Contents

Figures

Tables

Algorithms

Abbreviations

ACK	Acknowledgement
ARP	Address Resolution Protocol
CBS	Computer-Based Systems
CIA	Confidentiality, Integrity, Availability
COM	Communication Module
DiD	Defence in Depth
DNV	Det Norske Veritas
DPI	Deep Packet Inspection
FTP	File Transfer Protocol
GNS3	Graphical Network Simulator-3
GUI	Graphical User Interface
GW	Gateway
IACS	International Association of Classification Societies
IAS	Integrated Alarm System
ICMP	Internet Control Message Protocol
IPS	Intrusion prevention systems

IDS	Intrusion Detection System
IEEE	Institute of Electrical and Electronics Engineers
IMO	International Maritime Organization
IP	Internet Protocol
LAN	Local Area Network
MAC	Media Access Control
MITM	Man-in-the-Middle
ML	Machine Learning
NGFW	Next-Generation Firewall
NIST	National Institute of Standards and Technology
NVE	Network Virtualization Environment
OS	Operating System
QF	Quotient Filter
RST	Reset
SEQ	Sequence
SYN	Synchronize (TCP flag)
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
UNIC	Unified Controls
VM	Virtual Machine
VLAN	Virtual Local Area Network
WDCU	Wärtsilä Data Collection Unit
ZTM	Zero Trust Model

1 Introduction

This master's thesis was conducted in collaboration with the Finnish marine engine manufacturer Wärtsilä (www.wartsila.com). As global maritime technology continues to advance, the need to ensure that Wärtsilä's marine engine systems remain secure and cyber-resilient has grown accordingly. Engines and engine auxiliaries are increasingly exposed to various types of cyberattacks, making the firewall a critical component for network systems protection. Thus far, firewall testing has been performed manually, consuming both time and financial resources. The goal of this thesis is not to redesign the network architecture or improve firewall configurations, but to improve the efficiency and reliability of the firewall testing processes. Firewall testing is conducted for all Wärtsilä engine types and auxiliaries.

1.1 Motivation for Automated Firewall Testing

Modern maritime and industrial networks rely on firewalls to protect themselves from attacks and unwanted data traffic. However, while engine firewall testing is still performed manually at Wärtsilä, it exposes the engine to various error risks, thus putting the engine network systems at risk. Engineers at Wärtsilä have recognised the limitations caused by current testing methods, and therefore this thesis topic was initiated to provide a reliable solution. Since the tests need to be planned, executed and analysed afterwards, the concept of automated firewall testing using simulation tools offers a modern and efficient way to test firewall behaviour, improving both accuracy and reliability (Longo et al. 2023)

1.2 Research Questions and Objectives

This study will first examine the research problem from scientific perspective. By drawing on previous scientific studies on firewall security and testing. The target is to build a

strong understanding about earlier studies in the field. This study aims to study various simulation tools that can bidirectionally communicate through the firewall by both sending and receiving data. The objective is to support network testing engineers to filter out certain simulation tools that are ineffective and to propose simulation tools that are found to be effective to reach the research objective. This significantly improves the quality of testing within the Wärtsilä firewall testing context. This research is going to study following problems:

- Which simulation tools can send and receive network traffic through a firewall, and which tools can only perform one of these actions?

- How to effectively ensure cybersecurity in marine engine network systems while considering maritime regulations?

- What are the benefits of automating firewall testing from Wärtsilä's perspective?

1.3 Scope and Structure of the Study

The scope of this thesis is limited to the improvement of firewall testing practices for Wärtsilä's engine and auxiliaries network systems. The focus is on the testing process itself, with particular emphasis on increasing its efficiency by reducing manual procedures. The proposed solution concentrates on the use of simulation technologies to develop Wärtsilä's scope of supply network systems behaviour in marine vessels. Chapter two goes through previous studies on firewall technologies particularly in maritime sector, focusing on its existing firewall testing approaches, to establish a robust theoretical background for the study. Chapter three describes the research methodology and performs test cases using the selected simulation tools. In addition, Chapter three presents current firewall testing practices used at Wärtsilä and identifies the main limitations that occur with current testing procedures. Chapter four presents test results of the selected simulation tool setup and Chapter five discusses both contributions and

factors to consider in the proposed testing approach. Finally, Chapter six concludes the thesis by highlighting the main findings and outlining potential directions for future research.

2 Literature Review

This chapter presents the fundamentals of information security and the technologies associated with the field. It provides a comprehensive review of the development of information security throughout history to establish an understanding on modern firewall testing methods and the future direction of firewall testing. The objective of this chapter is to construct a theoretical framework for this research, which provides a foundation for the research methodology in the third chapter. This chapter reviews existing firewall testing simulation methods and aims to find weaknesses from prior research to develop a coherent understanding of the research. These analyses help define and understand the research perspective and guide how the research problem is addressed. This chapter will conclude with exploring emerging trends and potential future developments in the field of firewall testing.

2.1 Defence in Depth Model Fundamentals

Defence in Depth model (DiD) plays a vital role in computer and network systems, mitigating cybersecurity threats. The DiD model applied in current network systems focuses on readiness to confront different kinds of attacks by having multiple defence layers (Groat et al., 2012). According to Liu et al. (2013), the DiD model originates from military defence strategies, where the maximum resilience against the attacks is achieved by implementing various and unpredictable defence lines from attackers' point of view instead of a single layer. To ensure the security of network systems, multi-layer defence is applied to reduce the probability for an attacker to reach the network's core.

However, the study by Liu et al. (2012, p.267) states that a network cannot be completely safe from attackers. Multi-layer defence can be applied to fight the attackers, and when an attacker breaks one-layer, other defence layers can protect the system by learning the attacker's behaviour. This approach improves system's resistance against the attackers and protects network systems at many levels, including database servers, internal

networks and the network structure itself. While DiD increases the complexity of network system, it does not eliminate risks. This leaves a research gap for DiD systems developers to ensure better techniques for handling network attacks. The current DiD technology focuses not only on preventing attacks, but also on detecting them. This may expose the network system to risk in case of sophisticated cyber-attackers.

As illustrated in Figure 1, the network is protected by multiple layers, where each layer provides a different defence mechanism from the previous one. According to Liu et al. (2012, p.268) the defence layers' capability to resist attacks increases progressively as they approach the core network. This design enables the defence system to analyse the attack when it penetrates the outer layer, thereby significantly reducing the probability of an unauthorized access to the network core.

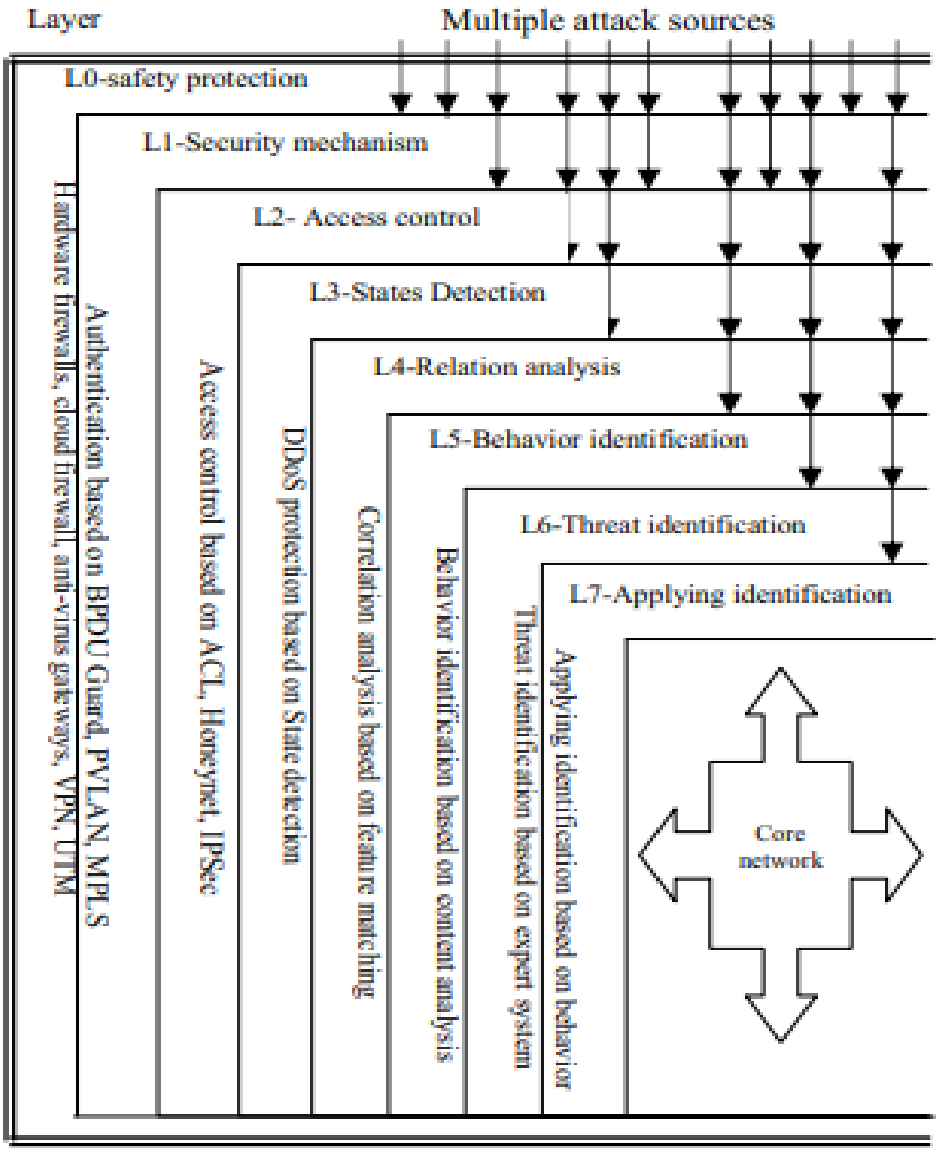

Figure 1. Multi-layer Defence in Depth structure (Liu et al., 2012)

2.2 Zero Trust Model Fundamentals

Alnoaimi and Alomary (2024) state that we are living in an era in which cyber threats have evolved to such an extent that the security models used in previous years can no longer be relied upon. The traditional security models assume that anything inside a network can be trusted with the only threat to the system coming from outside. Instead of trusting the internal system, the Zero trust model (ZTM) challenges this conventional

approach to information security by never trusting internal systems by default, but by continuously checking and verifying them.

As mentioned in Section 2.1, according to Groat et al. (2012), Defence-in-Depth (DiD) is a network security strategy in which multiple layers form a strategy to defend the core network. Alnoaimi and Alomary (2024) connect the relationship between Zero Trust Model and Defence-in-Depth model by further stating that Zero Trust Model (ZTM) is a method used within the security layers.

Even though ZTM implementation is increasing across network security systems, it has already faced multiple challenges and questions due to its lack of flexibility in its procedures because of its strict rules and policies. These methods can reduce productivity or cause the systems to overly restrict access and might even expose the network to other malicious activities. However, to address this issue, the U.S National Institute of Standards and Technology (NIST) Special Publication recommends in their guidelines evaluating trust-based scores and contextual information. This approach gives network system's device or user a trust score, dynamically evaluating how trustworthy they seem to be based on the device's recent behaviours and adjusts access accordingly (Amanlou et al., 2025, p.404). This approach leaves an open research challenge due to its complexity of accurately integrating it into current trust model technologies. In this trust model approach, the data and user's behaviour must accurately be described. However, with contemporary technologies this remains a challenge.

To address the challenge, Amanlou et al., (2025, pp.404-405) proposed a trust evaluation model using fuzzy logic. This model can address both dynamic and other network behaviour, while also addressing the uncertainties of real-world scenarios. Additionally, fuzzy logic model does not require training data the same way as machine learning models. In this approach, fuzzy logic evaluates data attributes based on parameters such as location, connection, past behaviour, login time, network usage and more detailed

contextual information. This approach enables the Zero Trust Model to achieve higher accuracy (Amanlou et al.,2025, pp. 404-405).

Figure 2 illustrates the proposed Zero Trust model using fuzzy logic. The module calculates trust scores according to data inputs. In addition, it is to continuously adjust the data, giving them trust scores and allowing the system to adapt to changes in user behaviour (Amanlou et al., 2025, p.407)

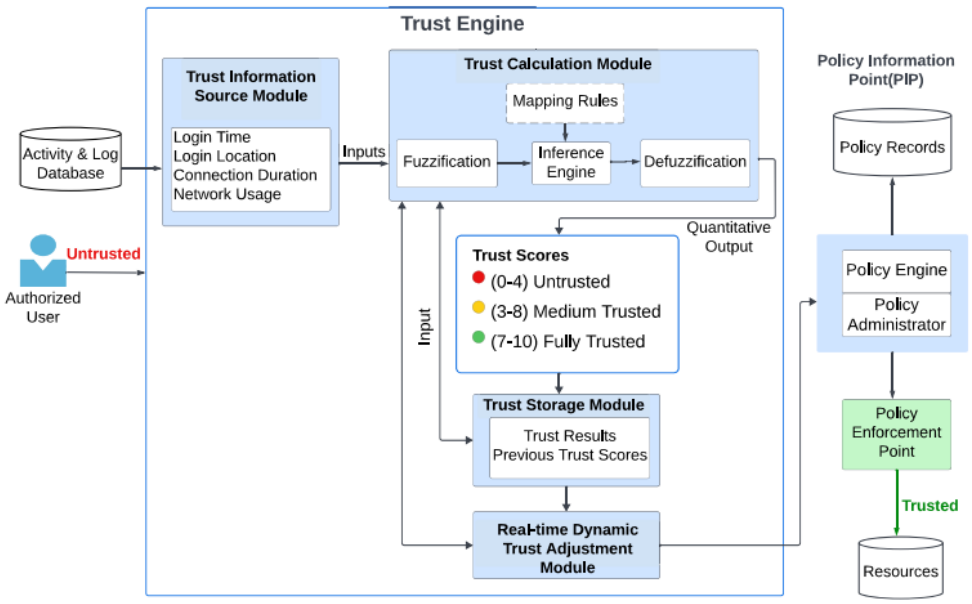

Figure 2. The proposed Zero Trust module to compute trust scores and Real-Time Data Trust Adjustments (Amanlou et al., 2025, p.407).

2.3 Confidentiality, Integrity, Availability (CIA Triad)

According to Osazuwa (2023, pp.1946–1947), network systems security plays a crucial role in contemporary society, where the exchange of information and data is increasing daily. Data security is essential for protecting systems in the digitalized world in which we live. Confidentiality, integrity, and availability (CIA) form the fundamental ethical principles of network security.

During the early days of the internet, network security was not considered a critical issue. At that time, the primary security concern focused on system functionality and reliability to ensure that systems operated as intended. As the internet evolved and its usage expanded, attention shifted toward protecting systems from external threats (Samonas & Coss 2014, pp. 23).

Samonas and Coss (2014, pp.23-24) continue in their study noting that the CIA triad originates from a military defence mindset, where the primary focus was on defending against external threats. As mentioned in Section 2.1, network security is therefore linked to military thinking, as the earliest research on computer security was initiated by US government and military agencies (Samonas & Coss, 2014, pp.23-24). Since CIA Triad mindset's objective is to protect against external threats, it remains vulnerable to system's internal threats. In this context, the proper integration of the Zero Trust Model, as discussed in Section 2.2, could significantly improve CIA triad security approach.

2.4 Firewall Models

Gouda and Liu (2005, p.1) propose in their research the first stateful firewall model. The firewall model consists of stateful and stateless sections. The firewall first extends itself by adding an additional field and uses the stateful section to compute values according to its current state. Moreover, the stateful section is used to check that current state is not affected by a previous packet.

According to Joaquin et. al., (2013, p.65), Stateless firewalls refer to the previous generation security policies and focuses on packet filtering in the lower layers of the Open Systems Interconnection (OSI) model. The stateful firewall model proposed by Gouda and Liu (2005, p.2) has three different functionalities. The first functionality is that it can track connections that already have been established. It does not solely track states but also supports tracking firewall behaviours (Gouda & Liu, 2005, p.2). The

second functionality occurs when the firewall is separated into a stateful and a stateless section, enabling existing results to be analysed by the stateless section alone, as the stateless section is in practice a stateless conditioned firewall as its name suggests. In the third functionality, the firewall is very straightforward, easy to understand and implement in real-world systems (Gouda & Liu, 2005, p.2). It is mainly used in stateful firewalls, but it can also function as a stateless firewall. This can be achieved by leaving the stateful component empty, and in this way, it can also be used as stateless firewall. In this case, the stateless model analyses and makes decisions solely based on individual packets. Each checking result of the stateful section for a packet is stored in an additional field called a tag (Gouda & Liu, 2005, p.3).

Stateful Section:

$$R_1 : I \in \{0\} \wedge P \in \{icmp\} \wedge T \in \{pong\} \wedge S = D' \wedge D = S' \wedge ID = ID' \wedge SN = SN' \rightarrow tag := 1$$

Stateless Section:

$$\begin{aligned}
r_1 &: I \in \{1\} \wedge P \in \{icmp\} \wedge T \in \{ping\} \wedge tag \in all \rightarrow accept; insert \\
r_2 &: I \in \{1\} \wedge P \in all \wedge T \in all \wedge tag \in all \rightarrow accept \\
r_3 &: I \in \{0\} \wedge P \in \{icmp\} \wedge T \in \{pong\} \wedge tag \in \{1\} \rightarrow accept \\
r_4 &: I \in \{0\} \wedge P \in \{icmp\} \wedge T \in \{pong\} \wedge tag \in \{0\} \rightarrow discard \\
r_5 &: I \in \{0\} \wedge P \in all \wedge T \in all \wedge tag \in all \rightarrow accept
\end{aligned}$$

Figure 3. Stateful and stateless section tracking protocol (Gouda & Liu, 2005, p.3).

As illustrated in Figure 3, depending on how many tag values a firewall needs, when a packet reaches the firewall, it checks both the packet and the memory state to see if the corresponding ping exists in the firewall. If the firewall remembers having the ping, it marks the tag value as valid by setting the tag value to 1. The tag is therefore a way to label packets based on history, enabling firewalls to block or allow network traffic (Gouda & Liu, 2005, p.3).

Joaquin et. al., (2013, p.65) add that although stateless firewall is fast in its operations, its lack of tracking packets state is a challenge that exposes it to risk. Joaquin et. al., (2013, p.65) note that stateless firewalls cannot stop certain attacks within an ongoing stream of traffic. In contrast stateful firewalls solve this problem by tracking connection status.

They can block the packets that do not meet the valid state as Figure 3 Illustrates. To address the challenge, a stateful filter takes manages the traffic, intercepting packets at network layer in case they do not match the defined security rules (Joaquin et. al., 2013, p.65).

2.4.1 Packet Filtering and Deep Packet Inspection

Deep Packet Inspection (DPI) is commonly used in packet processing, and as networks develop, the demand for more advanced DPIs increases. DPI is used to detect malicious or unwanted programs in the packet. DPI tools use many different strategies to detect and filter out unauthorized programs. Al-hisnawi and Ahmadi (2016, p.2217) highlight Software Defined Network (SDN) as one of the most promising DPI techniques; however, researchers remain sceptical, as SDN security cannot be fully trusted.

According to Al-hisnawi & Ahmadi (2016, p.2217), there have been many studies on Bloom filters to speed up DPI to meet today's requirements for high-speed networks. In the proposed new design, bloom filter's every two memory addresses are compressed together into a single main memory. The proposed bloom filter design is expected to work in parallel with Field-Programmable Gate Array (FPGA) to reach its full potential.

Nevertheless, Al-hisnawi & Ahmadi (2016, p.2217) propose a Quotient filter (QF) for DPI. QF inspects incoming data stream with specific signature. QF employs only one hash function, making traffic inspection fast, even faster than bloom filter. QF approach inspects all incoming network traffic by filtering all possible data traffic to check if they match any known signature.

In Figure 4, the DPI approach proposed by Al-hisnawi and Ahmadi (2016, p.2217) illustrates the filter's functionality. QF handles the incoming stream and responds if a specific signature exists. The input manager handles the data stream after it has been accepted by QF. In case of "signature match making", QF notifies match manager. Match

manager sends the packet forward to output manager. Finally, QF finds the matching string putting the packet back to its original path (Al-hisnawi & Ahmadi, 2016, p.2219).

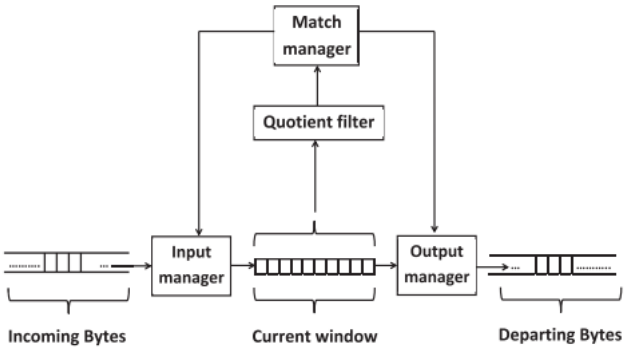

Figure 4. The proposed Quotient filter (Al-hisnawi & Ahmadi, 2016, p.2219).

2.4.2 Next-Generation Firewall (NGFW)

The next generation firewall (NGFW) is a firewall specialized in handling network threats on the application layer (Wang & Song, 2022, p. 585). According to Wang & Song (2022, p. 585), a famous IT research and consulting company, Gartner proposed the next generation firewall in 2009, dramatically improving the previous generations' efficiency in network security. NGFW is widely used because it offers strong security protection in application layer and simplifies the network architecture from user's point of view. Table 1 showcases the evolution of Firewall's generations, as it is critical to understand different types of firewalls and how they operate. The next generation firewall uses deep packet inspection, combining stateful firewall's network traffic flow control and intrusion prevention systems (IPS). According to Table 1, the strength of the next generation firewall is its capability to efficiently manage the application layer. Table 2 further compares the differences between previous firewalls to NGFW. In contrast, Neupane et al., (2018) argues that the radical change in application layer creates vulnerabilities in traditional network security. Palo Alto Next Generation Firewall security platform is therefore proposed to address the port-based network threats due to its capacity to trace all network traffic (Neupane et al., 2018).

Table 1: Firewall generations (Neupane et al., 2018).

Generation	1^{st}	2^{nd}	3^{rd}	Next Generation
Firewall Type	Packet Filter	Stateful Packet Inspection	Application Proxy	Deep Packet Inspection
OSI Layer	Transport Layer	Transport Layer	Application Layer	Application Layer
Main Functions	Filter packets based on source and destination IP addresses, port and protocols	Filter based on state and context of packets. Keeps track of each traffic using state table	Different proxy required for each service allowed. Acts as middleman between source and destination to reestablish a new session	Looks deeps into packet and makes granular access control decisions based on packet header and payload. Excels in managing application and data driven threats. Incorporates intrusion detection and prevention technology features.

Table 2: Comparison between Traditional firewall and Next Generation Firewall (Neupane et al., 2018).

Goals	Traditional Firewall	Next Generation Firewall
Prevent Advanced Persistent Attacks	• Only part of network security supplemented with IPS, URL filtering, gateway antimalware-malware products • Separately managing security tools is expensive	• Offer complete set of security technologies in one package • Combine all features of traditional firewall • Integrated package is easy to install, configure, deploy and manage as a unit which reduces administrative cost
Inspect SSL Traffic	• Cannot decrypt and inspect SSL traffic • Attacker can create SSL tunnels inside out to exchange command and control message	• Use Deep Packet Inspection technology to decrypt and inspect SSL traffic in both inbound and outbound direction • Detect and block botnet command and control message • Prevent advanced persistent threats using SSL
Control Web Applications	• Not application aware • Application control is a serious deficiency	• Offer application intelligence and control • Recognize specific application • Provide chart to visualize and control traffic by application
Manage Users & Use Policy	• No correlation of network traffic with users	• Allow application control at user group and individual level • Impose acceptable policies at high level of granularity • Allow to identify traffic by user and user group who pose security threats or involuntarily affect productivity through traffic visualization
Trade off Security vs Performance	• Administrator turn off monitor on specific ports, disable firewall rule and limit deep packet inspection which affect performance	• Parallel processing hardware architecture • Apply efficient approaches

2.5 Network Traffic Simulation

Network virtualization is regarded as an effective solution for addressing the ossification of contemporary networks. The development of a growing internet environment and the implementation of multiple tools and devices are essential aspects for achieving a highly flexible network environment. Virtualization enables researchers to study, develop and test network performance in virtualized environments. Since network traffic generation

tool are based on the internet protocol (IP) they can be supported by virtualized environments (Zhang et al.,2015, p.400).

Zhang et al. (2015 p.400) propose a network operation simulation platform to develop a network virtualization environment (NVE). The proposed network operation platform is an automated simulation platform. This approach supports this research and is further reviewed in Section 2.7. Unlike the traditional network simulation model, the proposed network operation simulation platform provides a double-layer network modelling approach.

2.5.1 Scapy and Ostinato

With the contemporary cyber-attack technologies, Local area networks (LANs) are at significant security risk due to its vulnerabilities in detecting malicious activities within the network (Hemanth Babu et al.,2024). According to Hemanth Babu et al. (2024) LAN threats originate from Address resolution Protocol (ARP), where ARP is used to identify the MAC address by which it can gain access to the IPv4 network infrastructure using the corresponding IP address. This attack is extremely severe, and it puts LAN devices at risk, exposing them to various attacks, such as man-in-the-middle (MITM) attacks. Once the attacker successfully employs man-in-the-middle methods in LANs, the LAN is hijacked and it is continuously being monitored by the attacker. The study by Rohith et al., (2018) suggests Scapy as an effective tool to decrypt packets of different kinds of protocols. Scapy handles multiple tasks such as tracerouting, scanning, testing, probing, attacking and discovery of networks.

Scapy is a Python program with an objective to manipulate network packets. Scapy creates objects using the Python programming language. Using Scapy, network packets can be effectively modified and manipulated in a variety of ways (Rohith et al., 2018). Rohith et al. (2018) highlight that Python-based tools are generally simple and flexible in programme code. When implementing packets as objects, Scapy only uses one line of

code, where even C program with its best libraries takes multiple line of code with the corresponding operation. In the study by Moharir et al. (2020, p.294), Scapy tool is studied using different protocols such as TFTP, FTP, HTTPS. File Transfer Protocol (FTP) is considered as the ideal tool for both sending and receiving information between computer systems over TCP/IP. The Transmission Control Protocol (TCP) function is optimized using a passive mode, where the server sends the client information it needs to open a data channel. Moharir et al. (2020, p.294) note that FTP connections are initiated by client and function well with firewalls and Network Adress Translation gateways.

The objective of this research is to automate firewall testing using network traffic simulation tools, where according to Wärtsilä network engineers, it has been challenging to establish a bidirectional data exchange between client and server hosts. According to Rohith et al. (2018), Scapy could provide a solution to the challenge, as it enables users to send sets of packets and receive replies. In brief, Scapy builds a set of packets, sends them, collects replies and analyses them (Rohith et al.,2018). Scapy's methods in Figure 5 illustrate the applied techniques, where a packet is created by considering the datatype and destination MAC address. The packet is then sent to the destination, or it can also be used to sniff packets using the defined parameters. In this thesis work, Nmap sends the packages, the answering machine is implemented with Scapy. Scapy moreover includes an answering machine, that allows users to create automated network responders by defining how the program should listen for incoming packets and automatically generate and send appropriate replies based on custom logic. This is further reviewed in Chapter three.

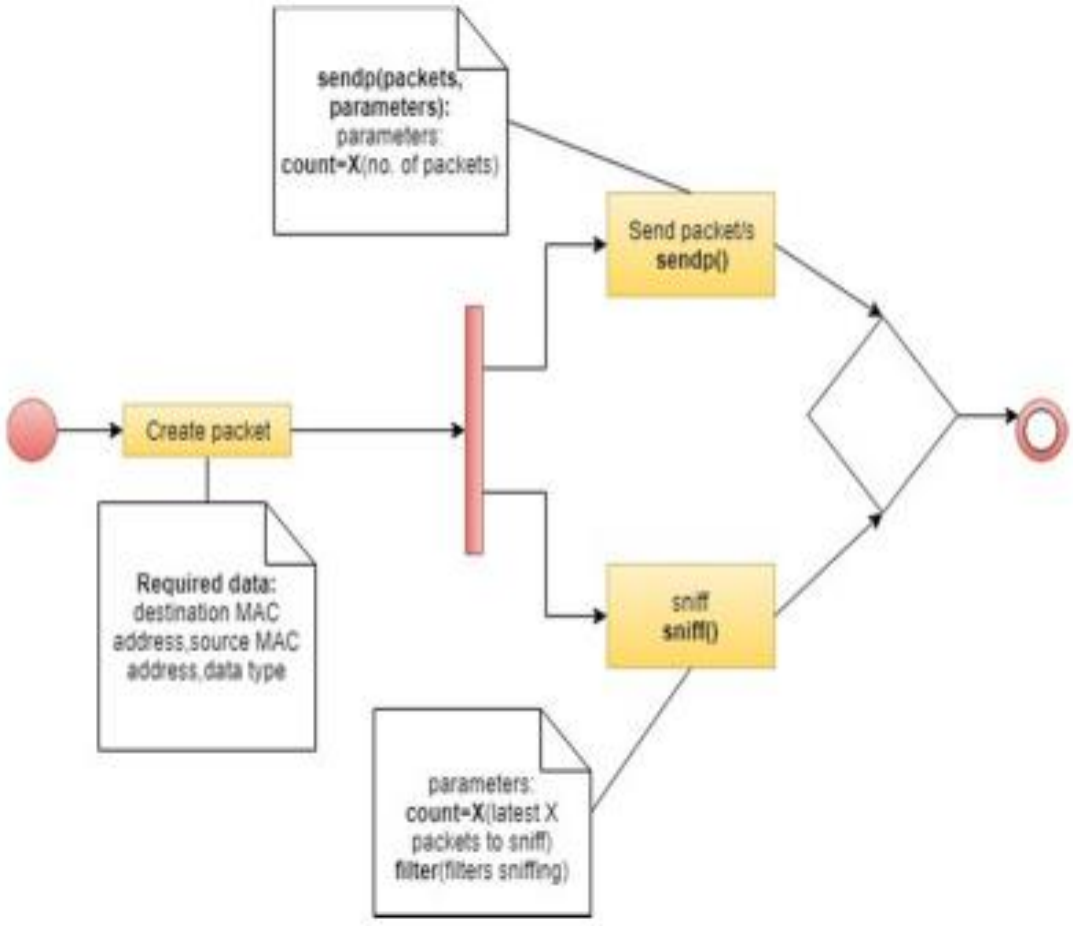

Figure 5. Brief illustration of packet manipulation tool, Scapy (Rohith et al.,2018).

Srivastava et al. (2014) state that network protocols have become increasingly diverse. Therefore, the need for flexible traffic generators across different network environments is vital. While Scapy focuses on packet manipulation (Rohith et al., 2018), Ostinato is designed to analyse network traffic and capture network streams, while also providing alternatives for packet manipulation (Patil et al., 2017, p.210). According to Patil et al., (2017, p.210), Ostinato supports a wide range of commonly used network protocols, such as Ethernet/802.3/LLC SNAP, VLAN (with QinQ), and ARP. It can perform automated tasks using a python API. Ostinato is an open-source network traffic generator with a simple graphical user interface (GUI). Ostinato's advantages include its ability to operate across different platforms and devices. Ostinato's architecture is built on an automated

Python script-based controller and drones. The controller functions as the Graphical User Interface (GUI), which manages the drones as Figure 6 illustrates (Patil et al., 2017, p.210).

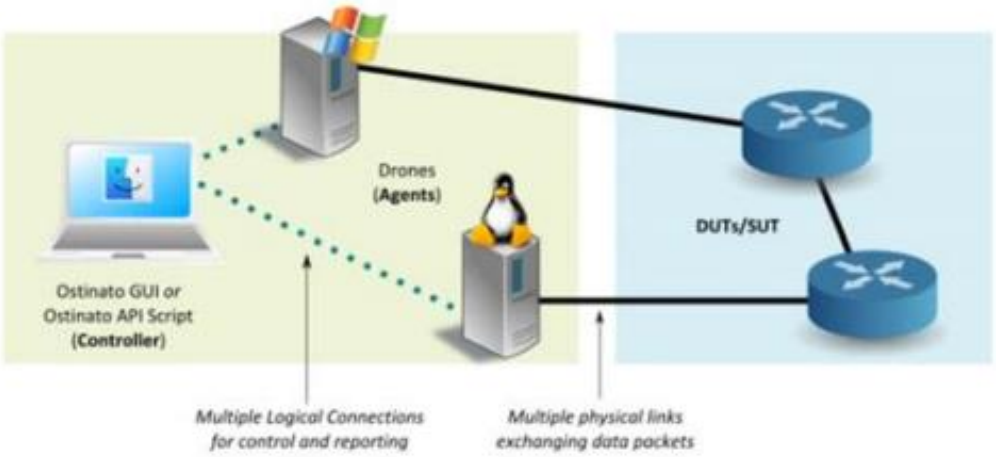

Figure 6. Ostinato architecture (Patil et al., 2017, p.210).

Wireshark (Patil et al., 2017, p.210) can also be used to analyse network traffic. The combination between Wireshark and Ostinato form an ideal setup to achieve the optimal results in network traffic analyses. Ostinato actively generates network traffic, while Wireshark can be used to analyse network traffic generated by Ostinato.

2.5.2 Traffic generators

In recent years, the need for network traffic generators has increased with the development of network technologies (Srivastava et al., 2014). Traffic generators are used for network development, design and network traffic testing purposes. According to O.A. Adeleke et al., (2022, p.11), network traffic generators are designed to perform different tasks with specific objectives. They can model, generate, and craft different types of traffic packets. There is no single network traffic generator that can perform all network traffic behaviours. Instead, different traffic generators are designed to perform specific traffic behaviours.

2.5.3 iPerf3

iPerf is a network performance measurement tool that measures bandwidth, packet loss and jitter and it is used to synthetically generate Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) traffic (Srivastava et al., 2014). Iperf is also an open-source software, like Ostinato, as mentioned in the previous Section 2.5.1. Iperf measures TCP or UDP data between client and server to determine the throughput of the network, meaning that it performs end-to-end measurements by sending traffic and collecting results as Figure 7 illustrates (Abolfazli et al., 2015).

iPerf is one of the most widely used tools for network measurement, and it has two independent versions, iPerf2 and iPerf3 (Zielenski, 2023, p.525) The main differences are highlighted in Figure 8. In the study by Zielenski, (2023, p.525), the data transmission is initiated by client, and travels from server to the client, using both iPerf2 and iPerf3. iPerf3 can also perform the reverse test. iPerf2, however, run reverse test in different manner, but it's not believed to influence the results.

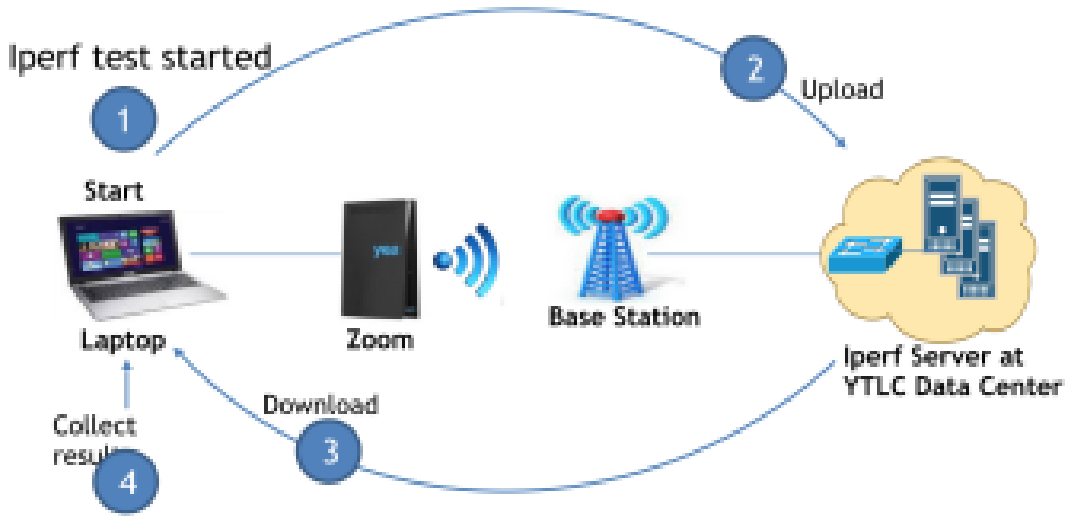

Figure 7. iPerf data collection **(**Abolfazli et al., (2015).

According to Abolfazli et al. (2015), Iperf is one of the promising throughput measurement tools. Its capabilities are frequently being studied to optimize its operational range. However, the study by Abolfazli et al., (2015, p.2) further states that even though Iperf is a widely deployed throughput measurement tool, its result accuracy cannot yet be relied on. Especially in industrial network environments as Iperf's measurement accuracy is hugely dependent on its own configurations. Abolfazli et al., (2015, p.2), note that to optimize the accuracy, an additional network behaviour monitoring tool is needed to predict future network performance and bottlenecks.

Feature	Iperf 2	Iperf 3
Traffic types		
TCP traffic	Y	Y
UDP traffic	Y	Y
SCTP traffic	N	Y
IPv4	Y	Y
IPv6	Y	Y
Multicast traffic (including SSM)	Y	N
TCP connect only	Y	N
Layer 2 checks	Y	N
Output options		
Human format	Y	Y
JSON output	N	Y
CSV (basic only)	Y	N
Hide IP addresses in output (v4 only)	Y	N
Client side server reports	N	Y
Traffic profiles		
Fair queue rate limiting	Y	Y
Write rate limiting	Y	Y
Read rate limiting (TCP)	Y	N
Bursts	Y	Y
Isochronous (video) TCP/UDP	Y	N
Reverse roles	Y	Y
Bidirectional traffic	Y	Y
Full duplex same socket	Y	N
TCP bounceback w/optional working load(s)	Y	N
Low duty cycle traffic with server side stats	Y	N
TCP_NOTSENT_LOWAT with select() (using the --tcp-write-prefetch option)	Y	N
TCP near congestion (experimental)	Y	N
Metrics		
Throughput	Y	Y
Responsiveness per second (RPS)	Y	N
UDP packets (total/lost)	Y	Y
UDP Jitter	Y	Y

Figure 8. Comparison between iPerf2 and iPerf3 (Zielinski, 2023)

2.5.4 Tcpreplay

Tcpreplay is an open-source tool originally designed to replay malicious network traffic. It is used to edit and replay previously captured network traffic. It stores network traffic data provided by different traffic generation tools (Parry et al.,2016).

According to Li et al. (2019, p.439), modern network applications often experience TCP performance that is difficult to diagnose. Since network applications are increasingly relying on high throughput and low latency TCP performance, it is being studied to optimize its performance in different traffic scenarios and applications.

Tcpreplay is used to test firewalls and security systems. Its objective is to take each packet dump and replay it without any connection awareness of transport layer or higher protocols (Cheng et al., 2004). However, since the objective of this research to identify tools that support end-to-end data exchange, tcpreplay does not meet this requirement. Catillo et al., (2020), states that tcpreplay tool is completely stateless, hence unable to support data exchanges between a host to another endpoint. (Zielinski, 2023)

2.5.5 Nmap

According to Al- Khazaali et al. (2024), network port scanning techniques are crucial in device cybersecurity as network ports are used in communication between devices. Port scanning techniques are widely used in developing network communication, but they have been exploited by harmful hackers in recent years. Port scanning aims to scan the state of ports. Whether the port is closed, filtered or open. The study by Al- Khazaali et al. (2024) describes Network Mapper (Nmap) as a convenient tool for this purpose. Nmap scans the target device, detects activities and services on the target device. The sS and sT Nmap port scanning techniques are the default scanning techniques. They will be reviewed in Chapter three as these techniques are used to effectively generate traffic.

During the TCP connect scan Nmap sends probes to target device. TCP connect scan and SYN scan are utilized in port scanning. The target receives a packet request asking if the port is open or closed. The scanner sends an SYN ACK indicating that the port is open, or an RST ACK if port status is closed.

2.6 Network Emulators

Network emulators are modern network virtualization technology, where multiple interconnected computer systems can interact in the manner comparable to real world scenario. It is widely used as a critical tool in the designing, testing, evaluation of network systems and protocols. The objective of emulators is to provide a real-world network environment such as all topologies, components, protocols and traffic behaviour. It also helps network engineers to study system behaviour in operational conditions (El Bouanani et al., 2024, p 1).

2.6.1 Mininet

The study by El Bouanani et al. (2024, p 1) presents Mininet as an easy-to-use emulation tool that offers an efficient solution. Network emulation tools and virtualization are not the same thing. Virtualization tools allow emulators to provide environments for different network applications, such as in previous sections mentioned network systems and protocols. Modern emulators including Mininet allow researchers to test their systems in real-word complex network environments. In addition, El Bouanani et al. (2024, p 2) state that Mininet allow users to run any emulated hosts or network node for network management. According to Hardin et al., (2023, p.5) Mininet emulation technology has already been in use with iPerf simulation Tool. The combination of Mininet and iPerf have been applied to simulate different network components including firewall designs. The study by Hardin et al., (2023, p.5), however found that the results of iPerf and Mininet combination could not be fully relied on. The client host device may

report unexpected throughput values that are higher than the link's capacity. Incorrect throughput results were found dependent on the TCP window size.

2.6.2 GNS3

In the contemporary landscape of cyber warfare, multiple countermeasures against network systems attacks are being investigated (Sirijaroensombat et al., 2019). Sirijaroensombat et al. (2019), propose a widely accepted Graphical Network Simulator-3 (GNS3) for the development of a network emulator testbed. Unlike Cisco Packet Tracker and Huawei eNSP, GNS3 is supplier-independent and can therefore be used to emulate firewall behaviour. Moreover, GNS3 can enable virtual environments by connecting to virtualization software such as VirtualBox and VMware (Sirijaroensombat et al., 2019).

In the Study by Korniyenko et al. (2019, p.245), GNS3 is used to develop a simulation model for enterprise network protection. The protection system includes components like firewalls. Korniyenko et al., (2019, p.246) selected GNS3, due to its capability to simulate virtual network with more than 20 different network component manufactures as well as its support for third-party devices to analyse network packets.

2.6.3 EVE-NG

Eve-NG is a tool that is used for testing network environments and operates with an original firmware. Eve-NG functions with network devices, including firewalls. Eve- NG is however limited to lab environments and might face limitations in real-word scenarios (Sharma et al., 2024, p. 918). The study by Sharma et al., (2024, p.918) compares Cisco Packet Tracer with the Eve-NG tool. Even though Cisco Packet tracer appears to be a convenient solution, Eve-NG tool is considered more effective in the lab environments. Cisco Packet tracer is limited due to its lack of required features in network design when

designing network topology. Moreover, it is limited when operating with original firmware. Whereas in Eve-NG can work with original firmware.

2.7 Virtualization platforms: VirtualBox and VMware

Modern communication technology is constantly conquering new areas in technology industry, while the need to find convenient tools to simulate system behaviour is becoming increasingly significant. Virtual machines aim to create networks, hardware, or software environments. Virtual platforms provide several benefits. They primarily demonstrate how real computer or network system components function and how such systems can be implemented. A virtual machine (VM) represents a virtual environment that is being controlled by software called Virtual machine Monitor (VMM) or hypervisor (Vojnak et al.,2019). The study by Khan et al., (2022, p.58) describes VirtualBox as a virtualization software, that enables virtual environments in which multiple Operating systems (OS) can operate simultaneously.

According to Vojnak et al. (2019), VMware offers both desktop and server-level solutions depending on user needs. For desktop use, VMware provide various tools such as VMware player, VMware Workstation and VMware Fusion for MacOS, which enables users to run virtual machines on Windows, MacOS and Linux systems. VMware hypervisors such as ESXi are designed for servers and typically run on Linux or Windows systems (Vojnak et al.,2019).

Vojnak et al. (2019) states that the most used hypervisors are VirtualBox and VMWare, and they are widely used to analyse and compare system's performances. According to Vojnak et al., (2019) Oracle VirtualBox is an Open-Source Software and is freely available. It is also considered one of the most powerful virtualization platforms in the field of technology.

2.8 Firewall Testing Approaches in Marine Engine Systems

Firewall testing aims to verify that communication between specific security range is constantly being controlled according to defined security policies as Figure 9 demonstrates. Unauthorized traffic must effectively be blocked to both prevent and allow certain traffic between security zones as highlighted in Figure 9. Network scanning techniques are commonly used to confirm security design. Performance testing ensures that security controls do not affect required operational communications (IACS, 2022). The firewall testing requirements by DNV (2021) provide basis for firewall testing with a focus on configuration validation, controlled traffic testing, and cyber security monitoring (DNV, 2021).

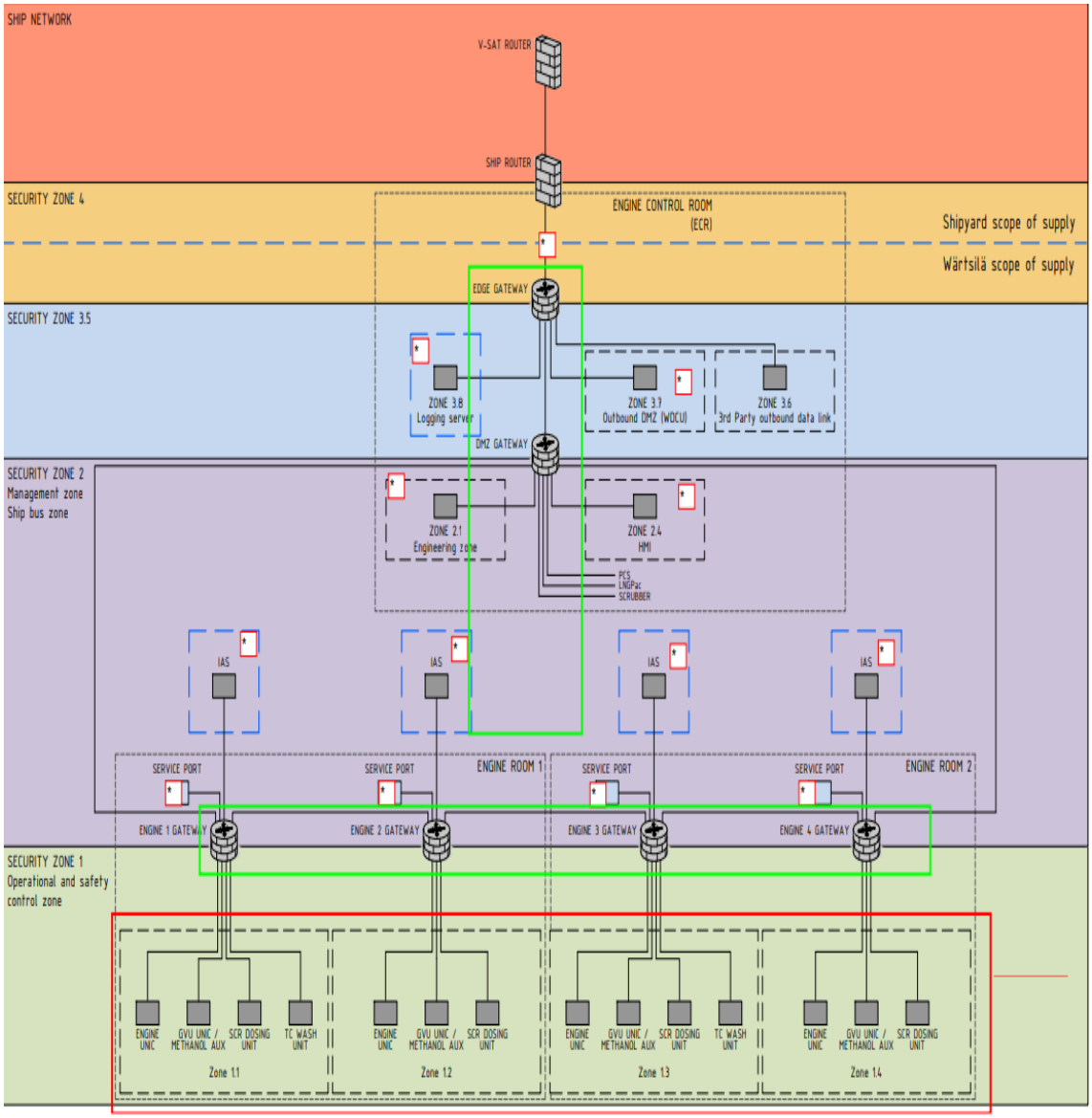

Figure 9. Network security zones in ship systems (Wärtsilä,2025)

2.9 Information Security Standards for Marine Engine Systems

Maritime cyber security standards objective is to protect maritime systems from cyber-related incidents. Cyber risks for maritime computer-based systems (CBSs) are all expected or unexpected malicious activities. Threats can be both internal and external. For instance, outdated software or ineffective firewalls are among the internal system threats. According to IMO (2025) guidelines, both threats should be considered to

efficiently manage cybersecurity systems. According to IMO (2025) implementing security measures such as firewalls in any ship digital systems is extremely important, and additionally, clear policies and procedures should be implemented to govern the appropriate use of cybersecurity systems (IMO,2025). The authorized or unauthorized network traffic may be expressed as “allow” or “block”. Firewalls are administered by host devices and end users are typically unable to affect the applied firewall rules, unless the end user functions as host in the computer system (RFC, p.9).

International standards highlighted by Martinez et al., (2024, p. 1451), provide a framework for improving information security, while IEC 62443 standards cover the security management aspects in maritime information systems. The POSEIDON management cycle component presented by Martinez et al., (2024, p. 1451) is a crisis management tool in maritime environment and it is considered as one of key areas, since it helps addressing the main challenges in maritime information security. POSEIDON’s objective is to direct, monitor and continuously evaluate information systems. It establishes ship cyber security objectives aligned with existing regulatory requirements.

2.10 Research Gaps and Future Trend

According to Shahraki et al., (2021), understanding and quickly addressing the challenges in network traffic behaviour in rapidly developing network technology, plays vital role in shaping network systems’ future. Modern network systems are becoming increasingly dependent on machine learning (ML) techniques, with topics from security aspects to performance (Xi Jiang et al., 2024). According to Shahraki et al., (2021), machine learning can effectively discover different network traffic patterns using various ML algorithms, however there are limitations caused the by large amount of labelled data since most of the real-world scenarios prefer supervised or semi-supervised, and it’s thereby considered ineffective in terms of costs by the authors.

The study by Xi Jiang et al., (2024), however considers ML as an opportunity to revolutionize the aspects of traffic generation using Scapy simulation tool. The existing traffic generation methods suffer from lack of data similarities with real-world statistics due to the limited methods that contemporary traffic generators offer. Xi Jiang et al., (2024) propose NetDiffussion, a synthetic diffusion-based machine learning model. It is a raw traffic generator approach with an objective to capture raw packet and visually represent them. To improve similarity to real network traffic, authors use controlled generation techniques to maintain the consistency of protocols to observe the real data.

Network traffic generated by NetDiffusion can be converted into raw packet captures, enabling its use in traditional network traffic analysis. Using tools like Wireshark, Scapy and Tcpreplay for retransmission, researchers realized that network operations could be effectively extracted from generated traffic, giving a significant improvement in applying ML network behaviour analysis. Finally, one of the key factors in NetDiffusion generated traffic is its ability to generate synthetic traffic using Scapy, giving it a huge advantage to the other ML-based traffic generation methods. Nevertheless, NetDiffusion cannot be applied in this research due to its lack of bidirectional traffic operation as it operates solely as a traffic sender Xi Jiang et al., (2024).

3 Research Methodology

This chapter presents the research methods used to conduct the research and is based on the observation and findings of the literature review. The objective of this research is to conduct practical experiments using different simulation tools and to evaluate implementation considerations in Wärtsilä firewall testing. The methodology is aligned with the research objective, which is to identify effective simulation tools for firewall testing procedures. The simulation tool experimental research is carried out by the author. The experiments focus on evaluating the applicability of the selected tools within the context of Wärtsilä marine engine firewall testing environment.

3.1 Research Approach

Firewall testing and implementation present significant challenges due to the complexity and variability of real-world configurations (Brucker et al., 2014). According to Brucker et al. (2014), while firewall functionality is tested by manufacturers, the firewall behaviour for each application in network systems must be configured in an application-specific manner. As a result, each firewall deployment requires individual validation to ensure that it operates as intended within a given network environment.

This research uses an experimental approach to study firewall testing in a real industrial laboratory environment. Therefore, IP and MAC addresses shall remain hidden from this document due to their confidential nature. The study focuses on Westermo (www.westermo.com) firewall embedded within Wärtsilä's marine engine systems. However, firewall configuration phase is performed by Wärtsilä network systems engineers.

Network traffic and security scenarios are simulated using Nmap, iPerf3 and Scapy to examine end-to-end communication between client and server under different conditions. The objective is to select a simulation tool setup that can both send and

receive network packets through a firewall. Previously, there have been challenges in establishing a reliable and efficient method for testing the bidirectional communication between the client and server host. Therefore, this study does not focus on sending packets from the client to the server but also on performing reverse communication from server host to client. This approach does not only allow the study to go beyond simple performance measures but also introduces a method for how firewall testing could be developed in future firewall testing procedures.

This research mainly explores network traffic simulation tools and their functionality in firewall testing; therefore, the literature review is in some cases analysed and re-evaluated based on the findings of the experiments of this research. Overall, the chosen research approach addresses both the technical and practical aspects of automated firewall testing as the study aims to comply with classification society cybersecurity requirements.

3.2 Simulation Tool Selection

Simulation is a widely used approach to design and analyse complex systems. It is a safe and cost-effective way to study real-world systems. The testing is conducted in an abstract manner, thereby mitigating risks to the real-world systems (Garrido, 2005, p. 266).

Since the aim of the study is to select simulation tools that can perform end-to-end communication between a client and a server bidirectionally. The research does not only aim to identify simulation tools that can perform this task, but also those that are unable to effectively communicate through a firewall. Ostinato was considered ineffective for the research objective due to its limited incapabilities to reply, as it is mainly used for traffic analysis and network stream capturing (Patil et al., 2017, p. 210). Tcpreplay was moreover excluded since it does not support data exchange from a client to another endpoint (Zielinski, 2023).

Thus, iPerf3, Scapy and Nmap were selected to provide a solution to the challenge addressed in the research problem in Section 1.2. iPerf2 was excluded because of its complex procedure in performing reverse traffic from server to client, as highlighted in Section 2.5.3. Instead, iPerf3 is selected due to its ability to effectively perform bidirectional traffic generation from client to server, offering an efficient alternative to reach the research objective (Zielenski, 2023, p.525).

During the simulation tool experiment conducted by the author, Scapy was selected due to its efficiency in port scanning, and ability to perform bidirectional end-to-end traffic generation. Scapy can reply even when a port is not open due to its ability to interact directly with the TCP/IP stack. When a TCP packet is sent to a closed port, the operating system (OS) kernel replies with an RST ACK (reset), indicating that the port is not being listened to. Scapy also replies when a port is open. According to Infosec (2013), a SYN ACK response indicates that the connection between the client and server is established, referring that the port is replying.

3.3 Experimental Environment

The research environment was conducted on Linux-based platforms that consisted of Kali Linux and Ubuntu platform. The test setup consisted of three main components. The components used to form the test set up were the client host and server host with Westermo firewall positioned between them. This structure enabled the evaluation of the research approach. The emulators and virtual machines were excluded from research scope since emulators cannot fully model or perfectly simulate firewall behaviour due to the firmware configurations that are case specified. To get a reliable solution and precise test results, the focus was shifted to simulation tools testing in real physical systems.

The network traffic performance was conducted using Scapy, iperf3 and Nmap within a Linux-based laboratory setup. The client and server systems were on separate Linux machines to generate and receive network traffic. Scapy, iPerf3 and Nmap simulation tools were used to generate and test traffic behaviour. Scapy library and Python 3 were used to send TCP and UDP packets for port scanning analysis.

The script in algorithm 1 was used to perform TCP SYN scans over a specific range of ports. The provided pseudo code in algorithm 1 corresponds to the algorithms implemented in the Python script using Scapy. The script generates IP/TCP packets and transmits them through a firewall to determine whether ports are open, closed, or filtered.

```
  INPUT target_IP, port_range
  SET open_ports = []

  FOR each port in port_range DO
      SEND TCP SYN packet to (target_IP:port)
      WAIT for response, timeout 0.4

      IF response == SYN ACK
          MARK port= "open"
          ADD port to open_ports
          SEND RST packet to end connection

      ELSE IF response == RST ACK
          MARK port = "closed"

      ELSE
          MARK port as filtered
  END FOR

OUTPUT open_ports
```

Algorithm 1: TCP scanning using Scapy

Since iPerf3 is configured to listen to a specific port as Figure 10 indicates. In this study, TCP connections were established between the client and server to listen to port traffic. The research environment provided a platform for analysing TCP traffic behaviour as well

as port scanning tools, which clarified our research objectives as the research progressed. Network performance was studied using and iPerf3 within a Linux laboratory setup.

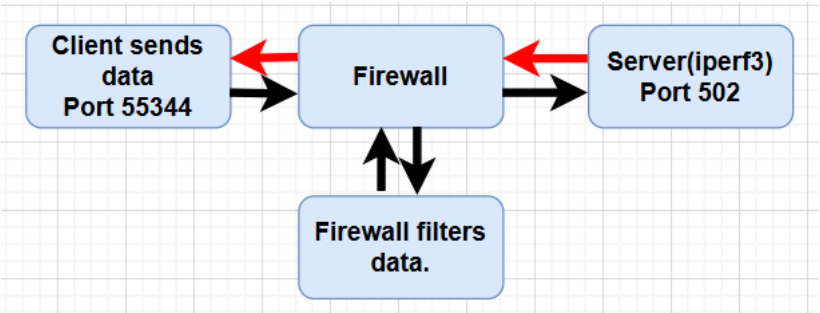

Figure 10. Port listening using iPerf3

3.4 Test case design

Testing is the most essential part when developing software. It has great impact on results and provides a framework for future direction (Rafique & Bin Faiz, 2023). Testing is usually expected to provide certain results, while aiming at the same time to find errors and provide solutions before deployment. Testing is needed even after deployment to analyse and track its functionality in real operational systems (Sighal et al., (2021). The test case design of this study follows a black box testing scenario, because the firewall behaviour is evaluated based on results provided by host devices. Port scanning experiments were implemented using Nmap as traffic generator due to its ability to effectively scan thousands of ports in short time. Nmap offers limited multi-protocol support for scanning purposes. The answering machine proposed in this study is a packet manipulation tool that intercepts incoming traffic and generates predetermined protocol-specific replies to the client. The answering machine operates as a fake server that responds to ARP, ICMP, TCP, and UDP network packet requests. The firewall is placed between the client host device and the answering machine as Figure 11 illustrates.

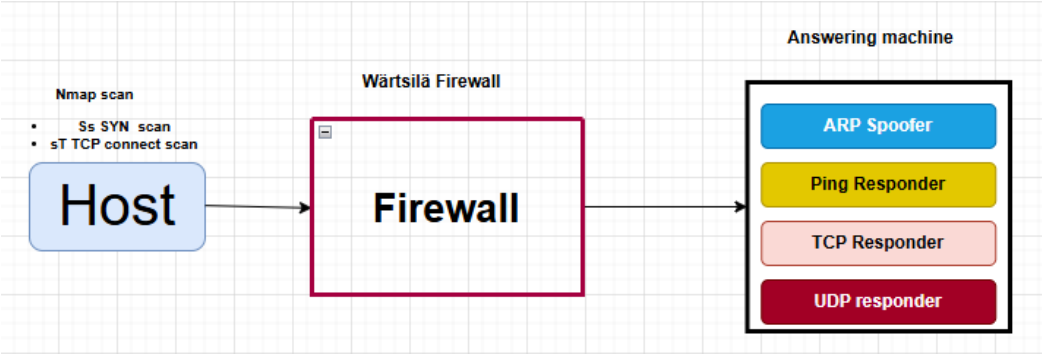

Figure 11: Automated Firewall Testing Setup

3.4.1 ARP Responder

Since Scapy's objective in this study is to manipulate packets, a special packet manipulation tool was developed to support automated firewall testing. An ARP spoofing Python script was implemented for a Scapy based answering machine to perform packet manipulation.

ARP spoofing that is implemented in this research operates as an answering machine. It operates as a man-in-the-middle (MITM), detecting ARP requests sent to the server host. Scapy then sends spoofed ARP replies to the client host device.

Scapy is therefore able to capture network packets as well as bidirectionally communicate with the ARP requester after passing through the firewall. The answering machine's ARP responder replies by sending the server host's MAC address to the host device. In this ARP spoofing method, the ARP responder replies to the client host by associating the server's IP address with its own MAC address thereby falsely claiming ownership of server host's IP address. The ARP spoofer pseudocode described in algorithm 2 indicates that if a device requests an IP address that do not belong to the answering machine, it will reply falsely claiming to be the requested IP address.

```
START

    If ARP request
    target IP is NOT server host IP:

        Create a fake ARP reply
        Tell client host: "I am the IP you are looking for"
        Send fake ARP reply back
END
```

Algorithm 2: ARP responder pseudocode

3.4.2 ICMP Responder

The implemented Python script includes ICMP responder that detects ICMP Echo requests (type 8) and ICMP Echo (type 0). The request in the ICMP answering machine was designed to identify incoming ICMP Echo Request packets and the Echo Replies.

As algorithm 3 illustrates, the ICMP packets are forged and therefore they are not the server host's real IP address. The proposed implementation enables firewall testing to perform ICMP packet capturing as well as capturing ICMP traffic and perform bidirectional communication within the automated firewall test setup.

```
START

    If ping request
    If IP is NOT server host IP:

        Create a ping reply
        Swap src and dst addresses

        Send reply back
END
```

Algorithm 3: ICMP Responder pseudocode

3.4.3 TCP Responder

The TCP manipulation is implemented within the answering machine using Scapy. The objective of TCP responder is to respond to incoming TCP connection attempts sent by the client host, while the answering machine replies with spoofed TCP packets.

In this research the TCP responder operates as part of multi-threaded packet manipulation system alongside ARP, ICMP, and UDP responders. As illustrated in algorithm 4, TCP responder uses a handshake mechanism in bidirectional communication. Together, these protocol responders simulate and model the host device's behaviour within the network test environment of this research. The TCP responder evaluates whether the incoming packet traffic satisfies the predefined conditions. Timeout for both Nmap and answering machine port scanner may be adjusted to get more accurate and reliable TCP port scanning results.

The predefined packets must satisfy following conditions:

- Ethernet framework
- IP layer
- TCP segment
- SYN flag
- ACK flag
- IP address that corresponds to host's address

```
START

    If TCP SYN request (handshake connection)
    if NOT server host IP:

        Create SYN-ACK reply
        Swap src and dst
        Swap address and port
        SEQ = server_seq
        ACK = client_seq + 1

        SEND reply
END
```

Algorithm 4: TCP responder pseudocode

3.4.4 UDP Responder

The UDP responder is furthermore implemented within the Scapy answering machine. Unlike TCP packets, which relies on a three-way handshake, the UDP protocol does not need connection to operate. The responder continuously scans incoming UDP traffic requests, while the incoming packet requests must include Ethernet framework, IP layer, UDP segment. When packet's predefined conditions are met, the responder replies. At the Ethernet layer, the destination MAC address is set to the source MAC address of the incoming packet, while the source MAC address is replaced with the MAC address of the server host.

At the IP layer, the source and destination IP addresses are swapped as algorithm 5 demonstrates. This allows the answering machine to appear as the host that was initially targeted by the incoming packet. At the transport layer, the UDP source and destination ports were reversed so that the reply appeared to come from the expected service port.

```
START

    If UDP packet
    If NOT for server host IP:

        Create UDP packet
        Swap src and dst
        Swap address and port

        If Data:
            Copy Data

        Send reply

END
```

Algorithm 5: UDP responder pseudocode

3.5 Firmware, and configurations

Firmware plays a vital role as an operating system in embedded systems. It is described as a software for hardware that enables and provides control of the device (Falas et al.,2021). This research studies Westermo firewall that operates with its own firmware, WeOS version 4.34.1. However, since the firewalls are supplied by Westermo and sold as part of Wärtsilä engine systems, the firewall configurations and implementations are defined by Wärtsilä. The examined firewalls were preconfigured according to Wärtsilä-specific configurations. Furthermore, the firmware and configurations used in this research correspond to those deployed in all Wärtsilä 4-stroke engine systems.

3.5.1 Functional Test Setup Cases

The functional tests conducted in this study have an objective to verify whether the tools reviewed in literature review function as expected. The purpose is to design a reliable connection between the client and server, and to provide reliable measurement results. As Figure 12 highlights data packets are sent to the server to listen to a specific port. Since iPerf3 can scan solely one port at a time, it does not support the research target. As highlighted in Figure 9, marine vessel systems can have multiple devices, each device with many ports. Testing each port separately was time-consuming and inefficient in real world operations.

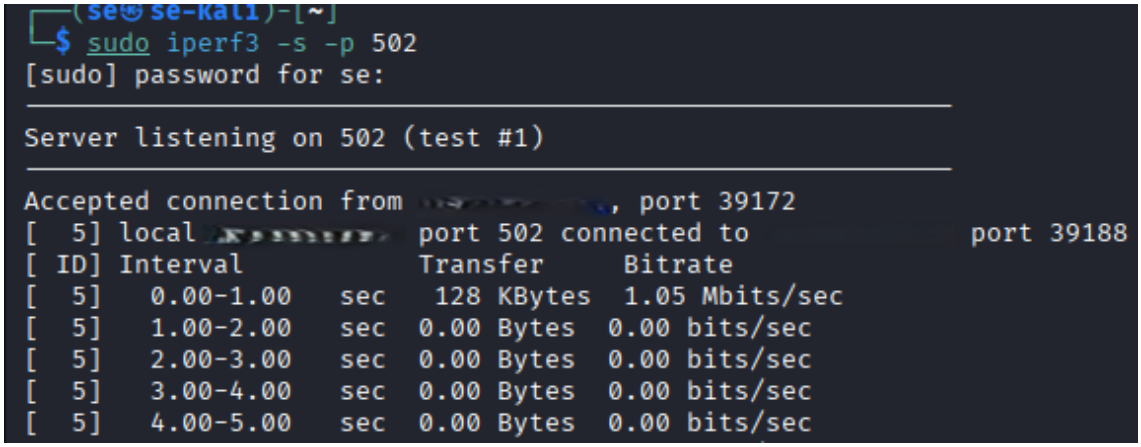

Figure 11. iPerf3 server receives traffic on port 502.

For port-scanning purposes, iPerf3 faces significant limitations to effectively scan multiple ports simultaneously. Therefore, Scapy was suggested to address the challenge with an implemented answering machine. Using Scapy in this experiment, the client host sent packets using Scapy to scan ports between a specific range of ports. In this study, ports between 1-600 were first selected for TCP scanning as Figure 13 illustrates. However, in the actual tests, the test would be conducted for 60 000 ports. Figure 14 demonstrates that Scapy was able to monitor all selected TCP ports at once. TCP SYN scanning was performed using a Python script by establishing a three-way handshake as shown in Figure 15.

```
Enter the target IP adress:
Enter the starting port: 1
 Enter the ending port:600
Port 1 is filtered (no response)
Port 2 is filtered (no response)
Port 3 is filtered (no response)
Port 4 is filtered (no response)
Port 5 is filtered (no response)
```

Figure 12: TCP Port Scanning command using scapy.

```
Port 498 is filtered (no response)
Port 499 is filtered (no response)
Port 500 is filtered (no response)
Port 501 is filtered (no response)
 Port 502 is closed
Port 503 is filtered (no response)
Port 504 is filtered (no response)
Port 505 is filtered (no response)
Port 506 is filtered (no response)
```

Figure 13: TCP Port Scanning using Scapy.

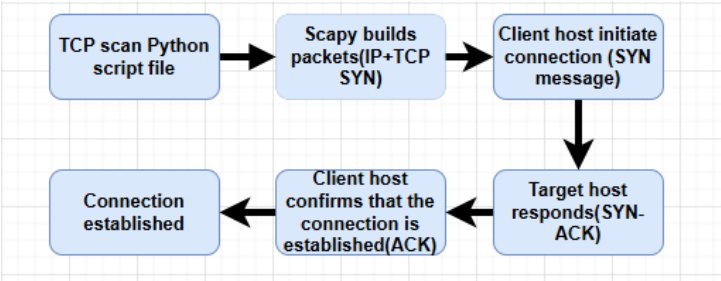

Figure 14. TCP Scanning Script Integration.

3.5.2 Performance Test Setup Cases

The firewall performance test cases were evaluated using iPerf3, Scapy, and Nmap. This research test set up consisted of a client host, firewall and server host. Firewall network performance test cases were performed to measure throughput, latency and retransmission using iPerf3. Firewall filtering behaviour was validated using Scapy with an implemented python script, which had an objective to scan ports as Figure 15

illustrates. The test set up experiment was performed using Scapy, iPerf3 and Nmap provide solutions to effectively evaluate firewall performance in real marine vessel operational environments. Tools were not used simultaneously. Each tool performance was experimented separately.

3.6 Automated Firewall Test Setup

As Figure 16 illustrates, a client host, a server host and Westermo firewall are the devices used in this research. The objective of automation framework is to enable automated bidirectional test executions through the firewall, scanning ports and reporting network traffic results.

Testing is initialized by client host with predefined IP address of server host. Network traffic is then generated using Nmap. With Scapy answering machine, the automation test approach is based on black box testing method that analyses the packet traffic without the knowledge of firewall configuration. However, the Scapy answering machine is predefined to reply to specific protocols to perform port scanning replies. This automated testing approach aims to scan multiple ports to enhance testing efficiency, in case of multiple devices in the ship engine network systems. The answering machine used to perform replies was configured to operate as a fake server host, as illustrated in Figure 11.

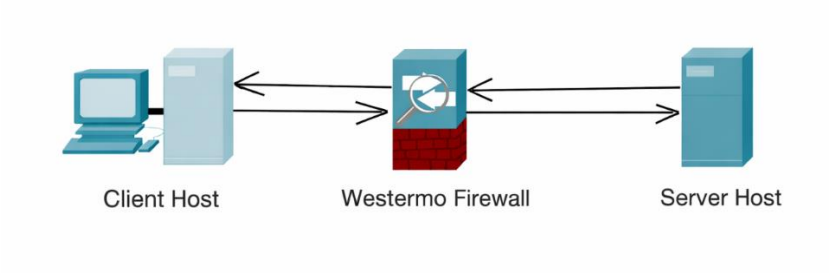

Figure 15. Research Automation System Architecture

3.6.1 Automated Firewall Testing Tool

To mitigate the manual interventions even further in the firewall testing automation, a tool to automatically execute all test cases was implemented in the client host as algorithm 6 demonstrates. The objective of the proposed automation toolbox was to simplify the testing process so that the person performing the testing does not necessarily need to be a field expert. The tool was implemented using a Bash script illustrated in algorithm 6. The scanning is performed using Nmap. Finally, a user manual document was compiled to guide the automated firewall testing users.

```
START

ASK user to connect cable to correct firewall port

IF user confirms (by pressing "y")
    SET UP network connection

    FOR each {engine,DMZ,Edge}
        SCAN {engine,DMZ,Edge}
    END FOR

    SCAN gateway

ELSE
    FINISH program

END
```

Algorithm 6: Pseudocode for the bash script implementation in client host.

3.7 Expert Interviews

The expert interviews are used to gather insights from professionals in the field. The objective of interviews part in this study is to attain practical perspectives from maritime engine system cybersecurity specialists. Experts' opinions collected in this study are Wärtsilä network system specialists.

The interviews conducted in this research were held by the author. The questions were based on research findings, while still enabling the experts to provide information based on their experience in the field. The challenges faced in firewall testing were addressed and the proposed firewall testing automation in this research was analysed.

The interview aims to document the key observations expressed by the experts. After completing the interviews, the collected responses were analysed by the author. The analysis focused on identifying challenges that might occur in firewall testing automation possible implementation in Wärtsilä marine engine network systems. These insights were to spot technical findings of the study and to provide practical context for the experimental results.

4 Automated Traffic Test Results

This Chapter presents the test results performed by the author. Experiments were performed using Nmap as traffic generator, Westermo firewall, and a Scapy-based answering machine as Figure 11 illustrates. The objective is to evaluate and to critically analyse the results of the solution proposed in this research. The Scapy answering machine operates as a server host.

4.1 Nmap Performance Results

As Figure 17 illustrates, the host device is reachable, establishing a connection between the client and the server host device. Nmap detects whether the ping request is being answered by the answering machine, while the actual test focuses on identifying which TCP and UDP ports are open. A ping request is initiated by Nmap to verify that the answering machine is reachable. It is reasonable to first send the ARP request to determine, which MAC address to ping, as Nmap utilizes ARP requests to access the targeted IP address. A total of 12 000 UDP and TCP ports were successfully scanned. The scanned IP addresses were all filtered by the firewall, as the objective was to prevent unauthorized access from external devices.

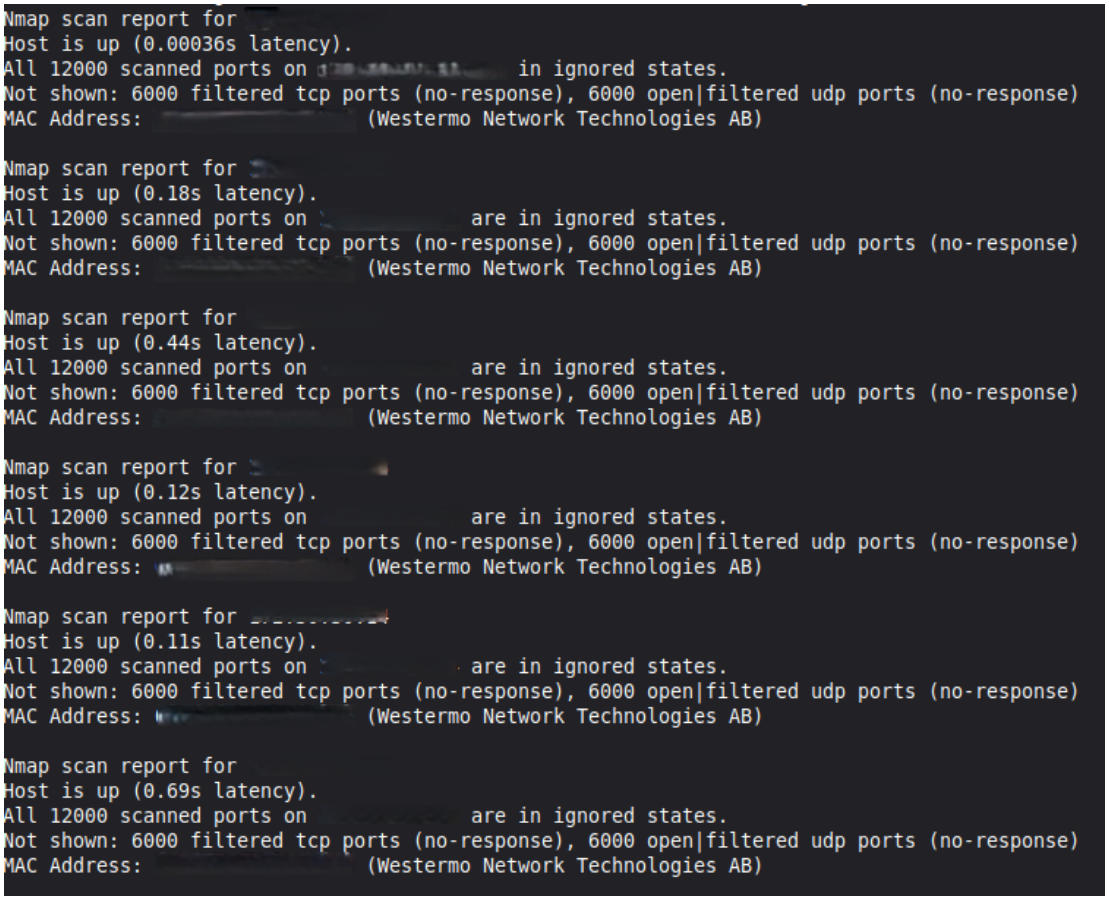

Figure 16: Nmap Scanning Results

4.2 Firewall Performance Results

The functionality of the firewalls used in all test cases were likewise successfully tested. The port functionality was tested for 1-6000 UDP and TCP ports. The results showed that the host was up, and it was responding. As expected, the preconfigured TCP and UDP ports were either reported as closed or open. All other ports were filtered. As Figure 18 illustrates, all port conditions were effectively scanned in 63.84 seconds. The firewall functionality test was conducted for Edge, DMZ, and Engine gateway (GW) that were utilized in this research.

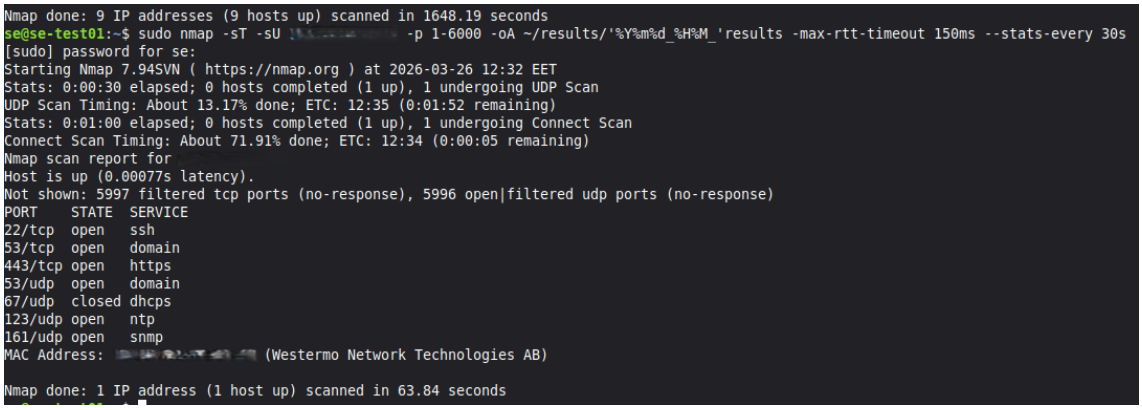

Figure 17: Firewall Ports Scanning

4.3 Answering Machine Performance Results

The answering machine designed by the author operated as expected. It could listen and process requests from the client host. Figure 19 illustrates the answering machine traffic information. As Figure 19 further illustrates, the test setup enabled ping requests, enabling Nmap to continue all other protocol scanning. The request primarily waits for ping replies. If the ping request is not answered, Nmap concluded that there is no connection between the devices. Therefore, the ping request plays essential part in network traffic testing on the server host. Nmap was capable to scan server host's ports even when target device does not respond to ping request. However, Nmap gets in this case significantly slower, and therefore ping replies from the server host are preferred. Table 3 summarizes the functionality of the answering machine, where ARP protocol is spoofed by sending a spoofed ARP reply. The answering machine claims all IP addresses belong to the server device except the server host's actual IP address. This indicates that if an ARP request is for the actual IP of the server host, then the server host will reply to it. All other ARP requests are answered by the answering machine developed during this study. Unlike TCP, UDP does not require connection establishment or a three-way handshake between the client and server host. In this thesis automation setup, Nmap sends a UDP packet request. The answering machine only responds, if the packet gets through the firewall. If the answering machine responds to the UDP packet request,

Nmap concludes that UDP ports are open. If there is no response, Nmap cannot conclude whether the request is filtered by the firewall or if it was received by the answering machine but silently dropped. Therefore, UDP can often be difficult to interpret when sent to a server host, because the correct reply could be that the UDP packet traffic was filtered by a firewall, if no replies arrived at the client host.

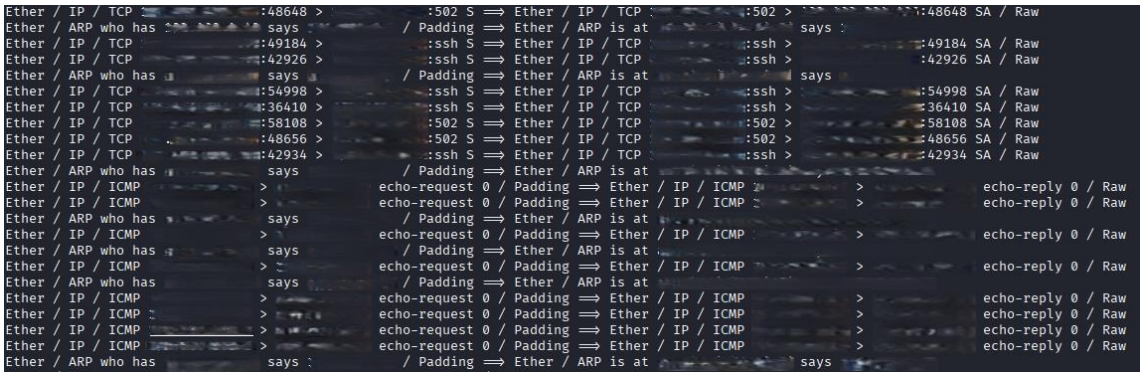

Figure 18: Answering machine responses.

Table 3: Answering machine responses

Protocol	Nmap Request	Response from answering machine	Meaning
TCP	SYN	SYN-ACK (SA)	Ports are open
ARP	Who has this IP?	Echo reply (Server IP)	I am the device you are looking for (ARP spoofing)
ICMP	Ping (Echo request) Can you hear me	Echo reply	Yes, I can hear you (Echo reply)
UDP	What is the UDP ports' condition? Are they closed or open	Responds or does not respond	If the answering machine responds, Nmap concludes that UDP ports are open. If no response, Nmap cannot determine whether the packets were filtered by a firewall or received but silently dropped

4.4 Engine Port Scanning Results

As shown in Figure 20, multiple engine ports were simultaneously scanned. All Wärtsilä scope of supply network systems were successfully scanned. The port scanning results identified both open and closed TCP ports. All UDP ports across all engines were marked as filtered, indicating that access from external services was denied. All engine systems had a common open port, namely Modbus port 502, as it is widely utilized in industrial communication systems.

Overall, the port scanning process operated as expected, allowing access only to preconfigured ports, while denying access to unused ports as shown in Figure 20. This provides a reliable solution to the research problem, as the previous approach was limited to scanning only used ports (20, 21, 22 and 502), leaving unused ports untested. The simulation experiment replies provide a new method to effectively analyse and evaluate the network traffic through the firewall. In this experiment 6000 ports were scanned, whereas in real-world marine engine network system scenarios port scanning may involve up to 60 000 ports per device.

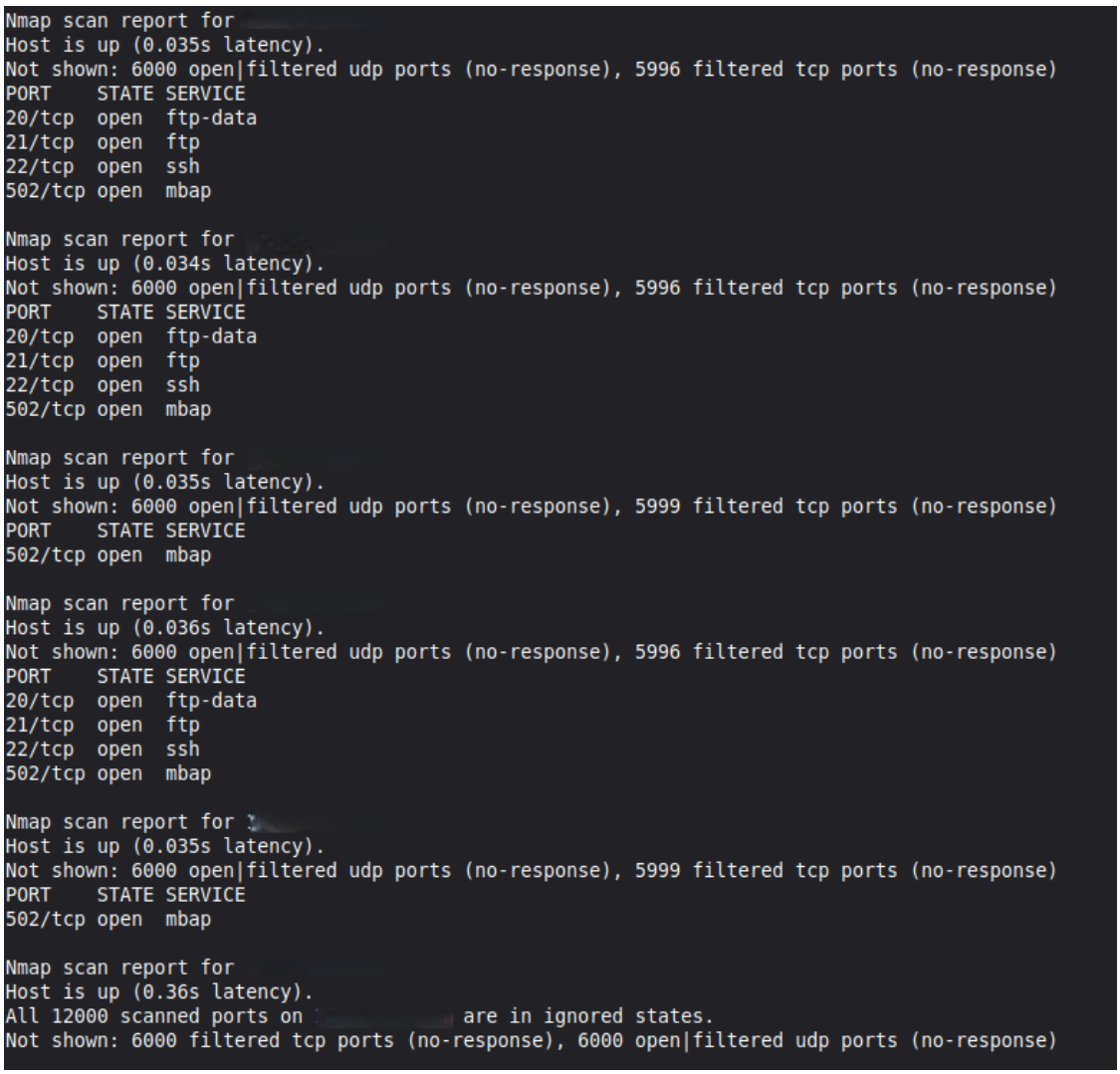

Figure 19: Engine Port Scanning results

4.5 Automated Firewall Testing Results

As shown in Table 4, the automated firewall testing system test scenarios experimented by the author were successful. Testing environment consisted of three main components: a client host, a firewall and an answering machine, as illustrated in Figure 11. The tests were performed using the automation toolbox described in Section 3.6.1.

Three different firewalls were utilized in the experiment after their functionality was verified as stated in section 4.2. Endpoint A refers to the client host, endpoint B

represents the target host device, meanwhile description highlights the traffic direction. As Figure 18 illustrates, both TCP and UDP port scanning were performed simultaneously. The command-lines *sudo sT and sU* used in every test case, indicate that the scanner performs both TCP and UDP port scans in parallel.

This automated testing system is a significant advancement compared to earlier Wärtsilä firewall testing, where the only information collected about the firewall behaviour was whether certain ports are open or closed, while all other port conditions remained without a guarantee that they functioned as expected.

Table 4: Performed Automated Firewall Test Scenarios

Firewall	Endpoint A	Endpoint B	Description	Expected Results	Observations
Edge GW	WDCU	Engine + aux	Data traffic to WDCU	Access allowed	Hosts are reachable. ICMP and UDP traffic are not blocked. TCP ports are filtered except port 502(Modbus)
DMZ GW	DMZ Service Laptop	Engine + aux	Engine Control Room Service access	Access allowed	Hosts are reachable. All TCP and UDP ports are filtered except TCP port 20, 21, 22, and 502 (Modbus)
Engine GW	IAS	Engine + aux	External IP (NAT)	Access allowed	Hosts are reachable. All 12 000 UDP and TCP ports are filtered except port 502 (Modbus)
Engine GW	IAS	Engine + aux	Internal IP	Access denied	Hosts are not reachable
Engine GW	IAS	Engine + aux	External IP, wrong IAS	Access denied	Hosts are reachable (replies ping request). All 12 000 UDP and TCP traffic are filtered.
Engine GW	Engine Service port	Engine + aux	Local service access	Access allowed	Hosts are reachable. All TCP and UDP ports are filtered except TCP port 20, 21, 22, and 502 (Modbus)

4.6 Comparative Analysis of Manual and Automated Testing

Previously, at Wärtsilä, an effective solution to reliably test all firewall ports had not been developed. Firewall testing has been conducted manually and therefore was limited,

since the ports not in use were not tested. Thus far, at Wärtsilä, the manual firewall testing procedures have typically been performed using real-world scenario tools such as Modbus scanners, UniTool, and PuTTY, which focus on verifying the functionality of selected ports.

In contemporary firewall testing, the Communication Module (COM-10), one of the Unified Controls (UNIC) modules, operates in the same role as the answering machine component developed in this research, allowing validation of preselected ports. While this approach provides a solution for usable ports in the actual marine engine systems, it is significantly limited, as it cannot verify the status of all ports. Consequently, the manual testing exposes the system to network security risks due to the large number of ports that are not tested.

The automated firewall testing approach proposed in this research utilizes Nmap in combination with a Scapy answering machine to enable large-scale port scanning. This method allows simultaneous scanning of thousands of TCP and UDP ports, providing a solution to reliably inspect network traffic within marine engine systems. The answering machine further enhances the testing process by simulating network behaviour on the server host, including ARP, ICMP, and TCP protocol replies. UDP packets are not displayed in the answering machine, because it does not necessarily reply, as the packets are either silently dropped or they are filtered by the firewall. In case a UDP port open, the answering machine is designed to reply.

Finally, automated testing results provided an advanced method of securing devices by verifying the complete behaviour of all the ports. Therefore, the suggested test setup provides an efficient alternative to address the research problem.

4.7 Experts' Analysis

From a value perspective, the automated firewall testing proposed in this research introduces a reliable method to evaluate firewall configurations. Moreover, the system reduces dependency on experts, as it instead enables larger usage for other personnel to conduct firewall testing (Karlå,2026). According to Karlå (2026), the test setup does not solely enhance operational efficiency but also significantly reduces time resources. Senior system expert, Gaupp (2026) considers the method for testing the entire engine network system as a significant advancement, as it allows the whole system to be evaluated at once instead of the previously used method of testing the preselected spots.

Key concerns when utilizing the proposed automation testing system were identified. A loose or faulty cable might expose the system to inconsistency and misleading test results. The manual reconfiguration of connections between tests at the client host exposes the testing to human error, affecting the end results (Karlå, 2026). In addition, Gaupp (2026) highlights that knowledge about the testing system must be maintained for users to prevent misinterpretations of test results. Karlå (2026) suggests the integration of a managed switch solution to eliminate the need for manual cable switching during different firewall test cases to increase test efficiency and reduce the risk for errors.

5 Analysis and Discussion

Although the automation system developed in this research appears robust and functioned as expected, no system can be considered entirely faultless (Nmap, 2025). The techniques provided by Nmap are occasionally considered unethical due to their features to evade firewall rules. Furthermore, Nmap appeared to be capable to sneak past intrusion detection systems (IDS), that are commonly exploited by network system attackers (Nmap, 2025).

However, according to (Nmap, 2025) documentation , Nmap can further be utilized as a defensive counter measure to secure network security. According to Nmap's web page, Nmap does not provide any predefined mechanism to protect or detect firewall or IDS systems. As applied in this research, the Nmap traffic generation and the answering machine greatly depend on user expertise and experience to apply Nmap's different features as the official Nmap guidelines highlight (Nmap, 2025). The official tutorial provided by Nmap lists only the relevant options and describes functions, while attacks or network security defensive measures can be adapted according to specific application context (Nmap,2025).

In the context of this study, Nmap offers a highly adaptable environment for firewall testing, enabling the system scripts and functionality of the answering machine to be modified, when necessary, particularly in the case of system or product upgrades within the network or across the entire ship system.

5.1 Accuracy of test results

The reliability rate of the experiments conducted in this research is expected to be high because the research was carried out in real industrial testing environment. The firewalls used in the experiments were the actual Westermo firewalls with Wärtsilä-specific

configurations. Thus, increasing the validity of the test results and providing a reliable solution to the research problem.

However, certain limitations exist. Although Nmap is an efficient traffic generator, some port scanning protocols, such as UDP scan might be time-consuming. This may lead to decreased efficiency in Nmap's operations, resulting in increased testing duration.

5.2 Applicability in Wärtsilä Engine Network Systems

It is noteworthy to emphasize that the exhaustive automated firewall testing is not required for every delivery project. Instead, it is primarily conducted when changes occur in the network system, such as new release of updated firmware or testing template. The underlying assumption is that firewalls configured with identical parameters behave consistently across both legacy and updated environments. Usually, the complete firewall testing is typically performed approximately twice per year, unless specific changes necessitate additional validation.

Firewall configuration and different variations in protocol settings, such as enabling or disabling specific ports, may affect system behaviour and therefore require reverification to prove they meet the expectations. The answering machine studied in this research is therefore a convenient tool to effectively validate the functionality of new firmware or configurations. Human error factors might cause some inconsistencies. For instance, entering incorrect IP address or unintended activation of network interfaces, such as an activation of Wi-Fi during testing, might influence test results or prevent the testing environment to carry out tests. These factors may lead to test scenarios where expected outcomes are not achieved.

Moreover, it is noteworthy to note that configuration management becomes critical when different individuals are involved in different firewall validation processes. If a configuration has been implemented by one engineer, test carried out by another and

later applied by another individual based solely on general guidelines or testing template. Misinterpretations of system behaviour may therefore arise due to differences as interpretations of the test results may differ between individuals. Testing firewall ports and their behaviour after firmware or test template updates is therefore a critical aspect of ensuring engine network system security.

5.3 Compliance With Maritime Cybersecurity Standards

The cybersecurity rules are not necessarily checked in every customer project. However, the answering machine proposed in this research appears to be highly practical across different testing scenarios. It is expected that subcontractors may also benefit significantly from the answering machine, as it significantly simplifies the testing processes.

Since each project is different, firewall behaviour differs according to specific configurations. The primary advantage the automated firewall testing is the ability to consistently execute tests with different configurations and firmware. When test results output differs from previous or expected results, it is a clear indication that the firewall is behaving differently and therefore firewall behaviour would necessitate new tests or reconfiguration. Furthermore, classification society rules define requirements for network access, specifying that certain services and ports are to remain open, while all the rest of the ports must be closed (IACS,2022). This is noted in Wärtsilä firewall testing templates.

5.4 Future Direction for Firewall Testing

Future development of firewall testing must furthermore consider challenges originating from the operator side. As testing practices may vary between different stakeholders, it is essential to establish a consistent testing approach. The methodology presented in

this thesis provides a means to ensure firewall's reliability and validation, regardless of differences in operational procedures.

As periodic inspections are required in industrial systems, annual check-ups are carried out to ensure proper status of the system. If no updates or configuration changes are made, it can be difficult to demonstrate them without systematic testing. Therefore, the need for repeatable testing method to verify system integrity is essential.

6 Conclusion

This thesis addresses the challenges within the contemporary Wärtsilä marine engine network firewall testing procedures. The lack of a reliable bidirectional communication between host and server device structured the research approach into a black-box testing. The information about the functionality of the Wärtsilä firewall manual testing approach could not be relied on, as they primarily focused on predefined ports.

The network simulation tools provided a robust foundation for developing flexible and controlled testing environments. As presented in the Methodology section, not all tools were suitable to perform an effective bidirectional communication. The Scapy answering machine proved to be a great solution to the research problem. Furthermore, the combination of Nmap and Scapy based answering machine was found to be effective in achieving the research objectives. The conducted experiments provided a reliable solution for testing Wärtsilä's engine scope of supply.

The development of a Scapy-based answering machine enabled real-world simulation of server host behaviour, enabling ARP, ICMP, TCP, and UDP responses. This solution allowed the system to model real network interactions within marine engine network interfaces. Nmap complemented this approach by providing an efficient large-scale port scanning.

The experimental results demonstrated that the proposed automated firewall testing can successfully scan thousands of ports simultaneously across the test environment in short time. This is a significant improvement over previous manual methods, indicating that port scanning objectives were also reached.

The use of Westermo firewall with production-level configurations confirms the validity and reliability of the proposed automation testing approach, providing consistent test results across different firmware and configuration updates.

However, limitations concerning Nmap were identified. Nmap's UDP scanning was found time-consuming, as UDP scans occasionally require long time, as highlighted in Section 4.3. This may perhaps affect the accuracy of the port scanning results.

It is important to note that the integration of Scapy, Nmap, and the proposed answering machine not only enhance testing efficiency but also increase customer confidence in cybersecurity in the advanced Wärtsilä four-stroke engines. Furthermore, this approach addresses the research gap in automated cybersecurity testing and offers a framework for integration of machine learning-based traffic generation techniques as Xi Jiang et al., (2024) highlight.

References

Abolfazli, S., Sanaei, Z., Shen, Y. W., Tabassi, A., & Rosen, S. (2015). Throughput Measurement in 4G Wireless Data Networks: Performance Evaluation and Validation. *IEEE Symposium on Computer Applications & Industrial Electronics (ISCAIE)*.

Adeleke, O., Bastin, N., & Gurkan, D. (2022, p.11). Network Traffic Generation: A Survey and Methodology. *ACM* , 11.

Al- hisnawi, M., & Ahmadi, M. (2016). Deep Packet Inspection Using Quotient Filter. *IEEE COMMUNICATIONS LETTERS*, 2217-2220.

Alnoaimi, S., & Alomary, A. (2024). Zero Trust Security: A Comprehensive Comparative Analysis of Zero Trust Maturity Models. Zallaq, Bahrain: IEEE.

Amanlou, S., Doss, R., & Li, J. (2025). Implementing A Dynamic and Context-Aware Trust Evaluation Model for Zero Trust architecture (ZTA): A Fuzzy Logic Approach . *International Wireless Communications and Mobile Computing (IWCMC)* (p. 404). Geelong, Australia: IEEE.

Bouanani, E., Houssam, B. C., Dabbous, W., & Turletti, T. (2024). Fidelity-aware large-scale distributed network emulation. *Elsevier*.

Brucker, A. D., Brugger, L., & Wolff, B. (2014). *Formal firewall conformance testing: an application of test and proof techniques.* SOFTWARE TESTING, VERIFICATION AND RELIABILITY.

Catillo, M., Pecchia, A., & Villano, U. (2020). Towards a Framework for Improving DoS.

DNV. (2021). *Veracity by DNV*. Retrieved from https://standards.dnv.com/explorer/document/0ED73B3209DA42CDA6392BC3946585C9/4

www.rfc-editor.org. Retrieved from https://www.rfc-editor.org/rfc/rfc7288.html#page-5

Falas, S., Konstantinou, C., & Michael, M. K. (2021). *A Modular End-to-End Framework for Secure Firmware Updates on Embedded Systems.*

Gaupp, R. (2026). Expert Analysis.

Gouda, M. G., & Liu, A. X. (2005). A Model of Stateful Firewalls and its Properties. *Proceedings of the 2005 International Conference on Dependable Systems and Networks.* Austin, Texas: IEEE.

Groat, S., Tront, J., & Marchany, R. (2012). Advancing the Defense in Depth Model. *International Conference on System of Systems Engineering (SoSE).* Blacksburg, Virginia: IEEE.

Hardin, H., Comer, D., & Rastegrania, A. (2023). On the Unreliability of Network Simulation Results FROM Mininet and iPerf. *International Journal of Future Computer and Communication*, 5-13.

Hemanth Babu, T., Vidula, N., Gopi, K., & Kavitha, C. (2024). Detecting and Preventing VM-based and Mininet-based ARP Spoofing Attacks using Scapy. *IEEE*, 501-507.

IACS, U. E. (2022). *IACS*. Retrieved from https://iacs.org.uk/resolutions/unified-requirements/ur-e

Joaquin, G.-A., Cuppens, F., Cuppens-Boulahia, N., Martinez, S., & Cabot, J. (2013). Management of stateful firewall. *Elsevier*, 65.

Karlå, M. (2026). Expert Analysis.

Khan, R., Al Habri, N., & Al Ghamdi, G. (2022). Virtualization Software Security: Oracle VM VirtualBox. *Fifth International Conference of Women in Data Science at Prince Sultan University (WiDS PSU)* (pp. 58-60). Riyadh: IEEE.

Korniyenko, B., Galata, L., & Ladieva, L. (2019). *Research of Information Protection System of Corporate Network Based on GNS3.* IEEE.

Li, Y., Miao, R., & Alizadeh, M. (n.d.). DETER: Deterministic TCP Replay for Performance Diagnosis. *Proceedings of the 16th USENIX Symposium on Networked Systems Design and Implementation (NSDI '19).*

Liu, S., Yang, H., & Wang, F. (2013). Design of Network Security Early-Warning System. (pp. 355-359). Xi'an: IEEE.

Liu, S., Zhang, P., & Sun, H. (2012). Research on Defense in-depth Model of Information Network Confrontation. (pp. 267-270). Xi'an: IEEE.

Longo, G., Orlich, A., Musante, S., Merlo, A., & Russo, E. (2023). MaCySTe: A virtual testbed for maritime cybersecurity. *ScienceDirect*.

Martinez, F., Sanchez, L. E., Santos-Olmo, A., Rosado, D. G., & Fernandez-Medina, E. (2024). Maritime cybersecurity: protecting digital seas. *International Journal of Information Security*.

Minal, M. D., Adyathimar, K. B., Dr. Shobha, G., & Soni, V. (2020). Scapy Scripting to Automate Testing of Networking Middleboxes. *ASTES*, 293-298.

Neupane, K., Haddad, R., & Chen, L. (2018). *Next Generation Firewall for Network Security: A Survey.* IEEE.

Nmap. (2025). *Firewall/IDS Evasion and Spoofing*. Retrieved from nmap.org: https://nmap.org/book/man-bypass-firewalls-ids.html

Organization(IMO), I. M. (2025). Retrieved from wwwcdn.imo.org: https://wwwcdn.imo.org/localresources/en/OurWork/Security/Documents/MSC-FAL.1-Circ.3-Rev.3.pdf

Osazuwa, O. M. (2023). Confidentiality, Integrity, and Availability in Network . *Innovative Science and Research Technology* , 1946-1947.

Parry, J., Hunter, D., Radke, K., & Fidge, C. (2016). A Network Forensics Tool for Precise Data Packet Capture and Replay in Cyber-Physical Systems. *ACM*.

Patil, B. R., Moharir, M., & Pratik kumar, M. (2017). Ostinato - A powerful traffic generator. *2nd IEEE International Conference on Computational Systems and Information Technology for Sustainable Solutions 2017* (pp. 210-212). Bengaluru, India: IEEE.

Peerdh. (2024). *peerdh.com*. Retrieved from www.peerdh.com: https://peerdh.com/blogs/programming-insights/creating-a-port-scanner-using-python-and-scapy

Rafique, S., & Bin Faiz, R. (2023). *Guidelines for the Development of Automated Test Case Designing Generation Tool(s).* KIET Journal of Computing & Information Sciences [KJCIS].

Rohith, R. S., Rohith, R., Moharir, M., & Shobha, G. (2018). SCAPY- A powerful interactive packet manipulation program. *IEEE*.

Samonas, S., & Coss, D. (2014). THE CIA STRIKES BACK: REDEFINING CONFIDENTIALITY, INTEGRITY AND AVAILABILITY IN SECURITY. *JISSEC Journal of Information System Security*, 23-24.

Shahraki, A., Abbasi, M., Taherkordi, A., & Jurcut, A. D. (2021). *Active Learning for Network Traffic Classification: A Technical Study.* IEEE.

Shalvi Srivastava, S. A., Dr. A.M, S., Tarun, B., Gupta, A. K., & Vinodh, K. (2014). Comparative study of various Traffic Generator Tools. *Proceedings of 2014 RAECS UIET Panjab University Chandigarh.* Pune, India: IEEE.

Sharma, A., Kumar, A., & Kumar, M. (2024). *Comparison of Packet Tracer and EVE-NG Tools for Efficient Network Design .* IEEE.

Shinan Liu, X. J., Gember-Jacobson, A., Nitin Bhagoji, A., Schmitt, P., Bronzino, F., & Feamster, N. (2024). NetDiffusion: Network Data Augmentation Through Protocol-Constrained Traffic Generation.

Singhal, S., Jatana, N., Suri, B., Misra, S. F.-S., & Luis. (2021). *Systematic Literature Review on Test Case Selection and Prioritization: A Tertiary Study.* MDPI.

Sumeth, S., Nangsue, C. P., & Aswakul, C. (2019). Development of Software-Defined Mesh Network Emulator Testbed for DDoS Defence Study. *International Conference on Computer and Communication Systems.* IEEE.

Vojnak, D., Dordevic, B. S., Timcenko, V. V., & Strbac, S. (2019). *Performance Comparison of the type-2 hypervisor VirtualBox and VMWare Workstation.* IEEE.

Wang, Z., & Song, H. (2022). Research on performance test method for the next generation firewall. *2022 IEEE International Conference on Advances in Electrical Engineering and Computer Applications (AEECA)* (p. 585). IEEE.

www.westermo.com. (2024). *Westermo/WeOS/Downloads WeOS/WeOS 4*. Retrieved from Westermo: https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&cad=rja&uact=8&ved=2ahUKEwi6yu7H2-CSAxUDFRAIHbarBi0QFnoECB0QAQ&url=https%3A%2F%2Fwww.westermo.com%2Fsolutions%2Fweos%2Fdownload-weos%2Fdownload-weos-4&usg=AOvVaw2g5VDpTZdNfVgKdloNbwFT&opi=8997

Zhang, H., Wang, Y., Qui, X., & Zhong, Q. (2015). Network Operation Simulation Platform for Network Virtualization Environment . *IEEE*, 400.

Zielinski, B. (2023). Assessment of iPerf as a Tool for LAN. *INTL JOURNAL OF ELECTRONICS AND TELECOMMUNICATIONS*.