

Nabil Ouzahra MSc Project

by Nabil Ouzahra

FILE	1565_NABIL_OUZHARA_NABIL_OUZHARA_MSC_PROJECT_260366_1067048090.DOCX (517.1K)		
TIME SUBMITTED	23-APR-2021 05:43PM (UTC+0100)	WORD COUNT	11025
SUBMISSION ID	150295632	CHARACTER COUNT	62565

**EVALUATION AND ANALYSIS OF OPEN SOURCES IDS DEPLOYED AS DOCKER AND
VIRTUAL MACHINE**

*Project submitted as part of the requirements for the award of MSc Computer Networks &
Cybersecurity*

April 2021

Declaration of Originality

I, NABIL OUZAHRA declare that I am the sole author of this Project; that all references cited have been consulted; that I have conducted all work of which this is a record, and that the finished work lies within the prescribed word limits.

This work has not previously been accepted as part of any other degree submission.

Signed: Nabil Ouzahra

Date: 23/04/2021

FORM OF CONSENT

I NABIL OUZAHRA hereby consent that my Project, submitted in candidature for the MSc Computer Networks & Cybersecurity, if successful, may be made available for inter-library loan or photocopying (subject to the law of copyright), and that the title and abstract may be made available to outside organisations.

Signed: Nabil Ouzahra

Date: 23/04/2021

Copyright Acknowledgement

I acknowledge that the copyright of this project report, and any product developed as part of the project, belong to University of Wales Trinity Saint David, Swansea.

Table of Contents

ABSTRACT	6
CHAPTER 1 - INTRODUCTION	7
1.1 Introduction	7
1.2 Motivation	9
1.3 Aim.....	9
1.4 Objectives	9
1.5 Structure of the Project	10
CHAPTER 2 - BACKGROUND RESEARCH	10
2.1 Overview	10
2.2 VMware technology overview	11
2.3 Docker technology overview	11
CHAPTER 3 - LITERATURE REVIEW	12
3.1 Overview	12
CHAPTER 4 - RESEARCH METHODS	17
4.1 Overview	17
4.2 Research process	17
4.3 Research Philosophy	18
4.4 Research Approaches	19
4.5 Research Strategy	19
4.6 Choices	20
4.7 Time Horizon.....	20
4.8 Data Collection and Analysis Techniques/Procedures	20
4.9 Research deliverables	21
4.10 Resources and planning	21
4.11 Resources Required	22
4.12 Gantt Chart	23
CHAPTER 5 - DESIGN AND IMPLEMENTAION	24
5.1 Overview	24
5.2 Ethical considerations.....	25
CHAPTER 6 – CONTINGENCY PLAN (Due to COVID-19 Pandemic)	26
CHAPTER 7 – A TAXONOMY ON IDS IMPLEMENTATIONS USING DOCKER AND VMWARE	27
7.1 VMware Technology	27
7.2 Docker Technology	28

7.3	IDS as VMware or Docker	30
7.4	Taxonomy on the use of IDS as VMware or Docker	31
7.5	Comparison and Analysis of Previous Studies.....	33
7.6	Using Open-Source IDS as VMware or Docker.....	35
CHAPTER 8 – SUMMARY & CONCLUSION		36
CHAPTER 9 – RECOMMENDATIONS		38
REFERENCES		40
List of Figures		45
Index Terms:		46

ABSTRACT- The purpose of this research is to establish an understanding of how the Intrusion Detection System (IDS) can be deployed as a Docker container and Virtual Machine (VM) to monitor and analyses the network security of small and medium enterprises. This study focuses on the deployment and evaluation of some of the well-established open-source IDS systems using the new emerging trend that is Docker containers technology in comparison to the classic VM approach. Due to the lack of research been done in the field of IDS as Docker, the author is taken this great opportunity to highlight how this new emerging trend can be integrated to network security systems and have an impact on. Also, this would be a modest contribution to knowledge in this key field. This study comparison between Docker-based IDS and VM-based IDS using the top two open sources software Snort and Suricata would provide more clarity on the overall performance of each method and level of detection of this vital security system within every network enterprise when facing Denial of Service (DoS) and reconnaissance attacks.

CHAPTER 1 - INTRODUCTION

1.1 Introduction

Intrusion Detection systems (IDS) and Intrusion Prevention Systems (IPS) are widely used by all types of organisations and businesses of all sizes to monitor and analysis of their network security against unwanted traffic and any violation of security policies [1]. In the current hostile internet environment, the increase of the level of threats and the number of cybercrimes using a sophisticated exploit in a that starts rising alarms and concerns. Therefore, this topic has seen lots of attention from researchers in recent years, lots of tools, products and services have been developed to strengthen network security such firewalls, antiviruses, IDS, IPS, Network Behaviour analysis (NBA), etc. some of these solutions are commercial and other are open sources. When it comes to large companies and government organisation the cost of implementing a robust network security using some of the expansive and sophisticated tools and services such as Palo Alto, Sourcefire and SolarWinds is not an issue. However, when it comes to small and medium businesses the feasibility cost analysis would be key in deciding what type of security solution to be implemented [2]. Therefore Open-source solution sometimes is a good option.

Open-source IDS are a variety of security tools developed by communities and free to use and it can be adjusted to suit different business needs, a good IDS is only as good as the way it is been implemented and configured. The author will this experiment using the Network IDS (NIDS) approach simply because Host IDS (HIDS) will not scale well to small and medium networks, with NIDS it would be possible to monitor and analyse the entire traffic after the demilitarised zone from and to all the network hosts. When it comes to the techniques used by IDS to detects threats and malicious traffic, two major techniques can be identified as signature-based and statistical anomaly-based. The signature-based approach compares the incoming packets to the network against pre-stored traffic signatures that matches certain types of known attacks and create an alert to the administrator so actions can be taken, the problem with this technique is that IDS will not be able to detect attacks when no

signature has been made yet. As per the statistical anomaly IDS technique, a baseline is set up to determine the type of traffic, the protocols used, the ports and machines generating the traffic and the bandwidth, anything outside this baseline the IDS will generate alerts to notify the administrator [3].

The author will conduct these experiments to evaluate and analyses the performance of Snort and Suricata when it comes to detecting traffic network anomalies and attacks such as network scanning, ports scanning and Denial of Service (DoS) attacks [4], these types of attacks are considered among the notorious Cybersecurity threats that organisations still facing [5].

In terms of major issues that the author must overcome, it will be Docker complexity and heavy Linux coding to configure the snort and Suricata containers to perform a parallel analysis to the network traffics. However, this would have great benefits in developing new skills. Not to ignore developing a suitable experiment design in an isolated network while trying to mimic a good simulation of real network traffic. It is important to highlight that these experiments will have some important boundaries when it comes to other significant types of attacks that the network could face. Also, the simulated traffic can never match the real network traffic.

The key concept of this research is to establish whether containerised IDS would provide better protection to the network security against DoS and reconnaissance attacks comparing to a similar experiment using the VM approach. This is a unique experiment on IDS systems implemented as Docker containers, which has never been conducted the way the author intended to setup. However, there are similar evaluations for those open-source IDS in a different environment like VMware or dedicated servers. The author will conduct a novel performance evaluation of two open-source IDS installed as Docker containers to monitor a simulation of network traffic that mimics a small and medium enterprise when facing DoS and reconnaissance attacks [48].

1.2 Motivation

In this section the author examines some of the major motivation factors that have led to undertake this project. The first factor is the security nature and importance of the topic itself; the author has taken this opportunity to highlight the technological gap in the current market when it comes to small and medium enterprises defending their networks using open-sources software. Second, as new emerged technologies such as Docker or SDN, there is lack of updated research on the Docker deployment of networks IDS. Third, this subject was carefully selected to acquire a good understanding of the network security mechanisms to a typical small enterprise network by diving into the challenges that comes with conducting a real study comparison experiment. Fourth, this project was mainly selected to be utilised as a career path in the future.

1.3 Aim

This research project aims to evaluate a new deployment approach of the top two IDS open sources Snort and Suricata using the new emerging trend Docker technology. To gain a profound understanding, the author will conduct the required literature review of the different deployment and implementation methods of open-source IDS and the analysis of similar evaluation results of Snort and Suricata. This process will allow the author to establish a clear synopsis and overview of the practical elements of these experiments.

1.4 Objectives

In this project, the author will provide the readers with a clear explanation of each objective and a sound and justified methodology approach followed. Also, a structural guideline will be developed to conduct a successful implementation of this project. The researcher provides a carefully set list of objectives for this project:

- Demonstrate a good understanding of the deployment of open-sources IDS
- Conduct a performance analysis and comparison between Docker and VMware IDS deployment

- Conduct a performance analysis and comparison between Snort and Suricata using Docker and VM methods
- Outline the advantages and limitations for each method (Docker vs VMware)
- Draw a list of recommendations for the best IDS deployment method for small and medium networks.

1.5 Structure of the Project

The author has structured and divided this MSc research project into nine chapters to deliver all the required components to the readers, this done in systematic way to deliver a sense of flow and a better insight of the benefits, challenges and evaluation between open-source IDS deployed as Docker Verses VMware.

CHAPTER 2 - BACKGROUND RESEARCH

2.1 Overview

In the process of selecting some of the top open-source IDS systems deployed today, the author has chosen Snort and Suricata based on prior academic research and IT professionals' reviews.

Snort is single-threaded open-source IDS that was created in 1998 by Martin Roesch and developed by Sourcefire, it becomes a propriety of Cisco after purchasing Sourcefire in 2013, It is rules-based real-time traffic analyser compatible with most of operating systems (e.g. Linux, Unix, Windows, Mac OS, FreeBSD, etc.) [6]. It was chosen simply because it is still considered as the number one open-source IDS when it comes to the community support and rules update regardless of some critics about processing speed and the ability to adapt the rules to some of the challenging complex threats. The main difference between Suricata and Snort is that Suricata was developed in 2009 with multithreading processing ability, capable of handling the demands of modern network infrastructure, it was built to handle an effectively high volume of traffic while processing more rules. Like Snort it is compatible

with the most operating system. Both Suricata and Snort can be used as IDS or IPS when connected to the network using the same Syntax rules (not 100%) [7].

The emerging trend Docker has changed significantly today IT industry, this open-source container platform allows developers the flexibility to test, run and deploy their applications in an isolated environment [8]. Thanks to the lightweight, flexibility across different operating systems and management ease, Docker has become one of the fastest-growing trends in the last 5 years [9]. With the help of Docker containers, it would be possible to run multiple Snort/Suricata IDS in parallel in an isolated environment to mitigate the issue of engine become over-loaded and packet been dropped. Also, this could have a major impact on processing time and the usage of processing powers due to the lightweight nature of the Docker system [48].

2.2 VMware technology overview

VMware is basically an entity that develops virtualisation software. VM stands for virtual machine which is a physical computer's software-based representation and is known as VMware's base unit [20, 21]. To understand VMware, it is important to describe what virtualisation is - a concept initially developed in the 1960s by IBM [22]. Virtualisation is the separation of a service or resource request from the service's physical delivery. It allows the access of resource or service such as memory that is beyond what is physically available. Aside from virtual memory, virtualisation has also been applied in various other IT layers such as storage, networks, applications, operating systems and server or laptop hardware [23].

2.3 Docker technology overview

Containerisation is a novel kind of virtualisation and is basically a new approach for the delivery of software [27]. Currently, there is a trend indicating the migration of distributed infrastructures and applications from virtual machine-centric systems to container-centric systems. Containerisation is considered a lightweight virtualisation tool that allows edge/fog-, cloud- and IoT-

based distributed applications' deployment and execution [28, 29]. Containers are different from virtualisation hypervisors in that the former enables the running of user space above the kernel of an OS instead of virtually running machines on physical hardware through a layer of intermediation. Container virtualisation has therefore been described as virtualisation at the operating system-level. A single host can support multiple user space events through Containerisation [30].

CHAPTER 3 - LITERATURE REVIEW

3.1 Overview

In general testing and evaluation of open-source intrusion detection systems is not something new. However, the perfect implementation of IDS has always been difficult which makes the testing procedures of IDS an important research subject.

It is important to highlight that most open source IDSs are different when it comes to software design, configuration difficulties, optimisation capabilities, processing speed and the number of rules. All these differences would have a direct impact on the host hardware performance, the quality of detection and quantity of traffic can be handled.

The author is aware of these distinguish variables that would affect the performance and reliability of IDSs. However, the focus of this thesis is to evaluate advantages, disadvantages and performance results using this new Docker deployment approach in comparison to the classic VMware approach. The author will design different test experiments in an isolated network environment to perform different attacks on these IDSs using different generated traffic speeds, the practical part of this thesis will be designed to fulfil the below conditions:

- System effectiveness.

- Quantity of detection.
- Network traffic capability.
- Host machine performance.
- Test reliability.

The process of conducting this literature review would have great benefits on the author's knowledge and understanding of IDS deployment, design experimentation, set the limits, the scope and set the objectives intelligently for this project. Also, it would enable the author to gather enough information to talk about the topic with authority. Many related quality research papers and thesis have been found, some of these carefully selected studies and papers will be critically analysed and compared in this piece of work.

1st Paper: A Dynamic Scalable Parallel Network-based Intrusion Detection System using Intelligent Rule Ordering

The Network-based intrusion Detection System (NIDS) comprise software and hardware that help in identifying risks to a network. The complexity of NIDS results in some difficulties in processing network traffic. For NIDS to work efficiently, hardware and software must be up to date. The goal of the thesis is to come up with an architecture that reduces the processing time of NIDS in pattern matching. The thesis contains an introduction, background, approach, design, implementation, and experiment. It also contains measurement, analysis, and comparison, discussion and conclusion. The background talks of the technologies relevant to the thesis. The domains for the intrusion detection system are discussed. These domains are Host-based Intrusion Detection System (HIDS) as well as Network-based Intrusion Detection (NIDS). The jobs of these intrusion detection systems include checking critical system files and checking network flow [8]. To realise the purpose of the thesis, multiple steps are followed. The first is designing and implementing a suitable design of the steps to follow. Measurement, analysis, and comparison of results should also be appropriately done. A parallel

technique model that outlines why NIDS drops packets and how to reduce the packet drop rate is developed. A technical model is designed showing the features aimed at achieving the objectives of the dissertation. Two algorithms are designed as part of the technical model to test the functionality of the project. Once the technical model is designed, one must lay down technical infrastructure, which includes how the network traffic will be sent to the sensors. Once the set-up is complete, the actual experiment should be run to test the efficiency and usefulness of the layout [8]. Measurement, analysis and data comparison enhances accuracy in the project. The data obtained from the experiment is properly analysed to ensure the set goals are achieved.

2nd Paper: Open-source intrusion detection systems evaluation for small and medium-sized enterprise environments

The thesis compares three Intrusion Detection Systems (IDS) ranked by their importance to SMEs. Since the threats in the IT field are continually emerging, the tools required to counter the effects have to develop at a significant rate. Many firms that take security measures seriously normally employ IDS to maintain safety policies. The method used to conduct the research was a collective case study where set goals apply for an assessment of individual cases. Results obtained were applied for comparison of different IDS following factors set by Paytrail Company. This organization deals with payment through the internet and e-commerce services. The thesis mainly focuses on the security performance expected, trustworthy of the open-source project and integration. The study compares IDS solutions and different solutions based on these three factors. The thesis focuses on threats and vulnerabilities related to transactions within the network system. The risks that IDS can identify are grouped into Advanced Persistent Threat (APT) as well as malware [10]. Flow charts and figures are properly used to illustrate various issues in the thesis. In this collective case study, both qualitative and quantitative data are used. The evaluation of criteria in the thesis is conducted according to the factors set by the owner. The environment for the project was set with virtual servers that operate using the

oracle virtualisation software. The tests are run at the same time and the results documented from all NIDS. This simultaneous running and reading of results enhance results comparisons and minimises errors [10]. The author studies network attacks vectors and establishes tests for the same. Three open-source solutions that include Snort, Suricata as well as Zeek are analysed in the article to see how they relate as well as effectiveness in ensuring security to network environments run by SMEs.

3rd Paper: Evaluation of Intrusion Detection Systems under Denial-of-Service Attack in virtual Environment

Intrusion Detection Systems (IDS) identify malicious traffic in many organisations. The authors explore the diverse attacks, which are detectable by the IDS. The objective of the study was to appraise three commonly applied open-source IDS concerning performance. Other goals include understanding the performance of IDS and obtaining knowledge on cloud computing and cyber-attacks. The experiment is done using TCP, SCAN, ICMP, FTP attack. The assessment of the performance of any of the IDS is calculated using CPU usage, memory usage, and packet loss. This results in the generation of multiple alerts [11]. Various intrusion detection systems were analysed including Host-based and Network-based IDS. According to the authors, the IDS are designed to run in a specific environment, therefore, it is difficult to implement the thesis. The framework and architecture for the Suricata, OSSEC and Snort intrusion detection systems are laid down. The results obtained from the three IDSs are compared. The features compared include multithreading, global variable, windows support, and http accessing [11]. Some of the research questions are answered in the literature review method. Under this approach, it is possible to determine the performance of the IDS as well as the diverse attacks that the system can detect. The other criteria used include Population Intervention Comparison Outcome (PICO). The paper focuses on the comparison of three IDSs, which are Snort, Suricata and Bro.

The merits and demerits of each intrusion detection system are considered in choosing the best. The thesis also analyses the data for mobile networks using the Vulnerability Research Team (VRT) rules set through Snort and Bro software. Examples of attacks include bad traffic, scanning attack and denial of service attack. The environmental set up includes a network traffic generator and network switch. The experiment aids in determining accuracy as well as the effectiveness of IDS in performance. The results have been graphically demonstrated using flow charts hence it helps to make an easy comparison of the different intrusion detection systems.

CHAPTER 4 - RESEARCH METHODS

4.1 Overview

In this section, the author will explain the research methodology principles adopted and followed throughout this project. The research methodology is key when it comes to conducting any successful research project, it is a very important element to all professionals when it comes to gathering specific evidence and data to support researchers unique project. Many ways could be used to collect this information, for instance, interviews, surveys, case studies, questionnaires, experiments, optimisations, etc. this process helps the researcher to gain more control of the suitable literature that would fit the requirement of the project. Different projects require different methods, this could be a qualitative or quantitative research method. The author will conduct an empirical research process in this study to design practical experiments to produce some quantitative data that can be statistically analysed.

4.2 Research process

The research process has been adopted in this study would help the author to deliver a systematic model aiming to evaluate the Docker implementation of IDS in comparison to the classic approach of VM, this would benefit to comprehend the validity and reliability of information gathered through the literature. A well-structured research process has been the most appropriate for the proposed study. This process is informed by a layered approach that was developed by Saunders, Lewis, and Thornhill [12], and referred to it as the “Research Onion”. Use of the Research Onion allows researchers to systematically progress from the outmost layer towards the inner layers. The specific layers of the Onion include the following: philosophies, approaches, strategies, choices, time horizons, and techniques and procedures (see figure 1). Each layer has specific options from which the most appropriate for the study at hand are selected. Such a research process provides a clear insight into how a study was or will be conducted or the framework upon which the study was or will be based [13].

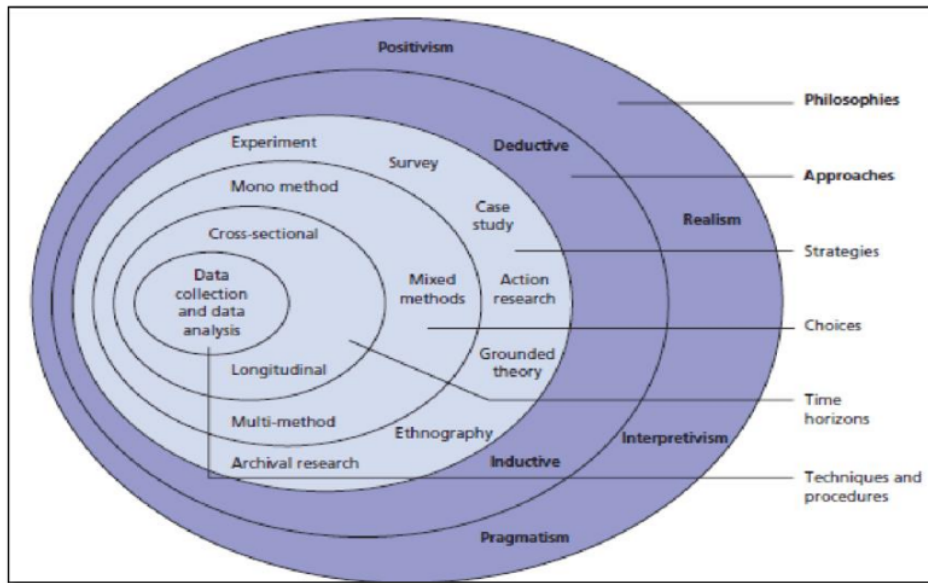


Figure 1: The research onion by Saunders

4.3 Research Philosophy

Research philosophy is the idea or belief about knowledge development, including any necessary assumptions about the understanding of the world and the collection, analysis, and interpretation of data. The selected philosophical standpoints reflect the overall perception and reasoning of the researcher concerning a particular study. Interpretivism, realism, positivism, and pragmatism are the most important research philosophies [14].

A combination of realism and positivism are selected as the research philosophies applicable to this study. Realism entails following a scientific path to draw theories and conclusions. This philosophical standpoint does not simply believe in perceptions without satisfactorily supportive scientific logic. Like realism, positivism is based on scientific investigation to observe and interpret reality. However, it specifically emphasises objective experimentation to capture trustworthy and/or factual knowledge. Consistent with this, realism entails the least possible interaction with study participants if there are any [15].

Considering that the current study attempts to experimentally measure the performance results using the Docker-based IDS deployment approach versus the conventional virtual machine or VM-based IDS deployment approach make the realism and positivism philosophical standpoints the most appropriate. No form of survey (such as interviews and questionnaires) would be needed as part of data collection. Rather, Snort and Suricata (open-source IDS variants) will be deployed in Docker and VMware environments to quantitatively measure the resultant performance realities [16].

4.4 Research Approaches

Saunders et al. focus on deductive and inductive reasoning – the main types of research approaches. Deductive reasoning leverages existing literature to discover theories that are then tested using collected data to draw conclusions. Contrastingly, inductive reasoning entails comprehensive observations (usually a collection of vast qualitative data) as the basis for developing one or more theories [17]. The current study focuses on evaluating whether Docker-based IDS deployment outperforms its VM-based counterpart. Precisely, it is hypothesised that Snort performs better than Suricata on both OS-level virtualized Docker containers and conventional hypervisor-level VM. Therefore, a deductive research approach is the most suitable for the current study.

4.5 Research Strategy

When it comes to the research strategy, the question shifts to what aspects the current study attempts to explore and the research design through which the objectives would be accomplished. Therefore, it involves the overall description of a study – is it experimental, action research, case study, survey, grounded theory? For the current study, an experimental research strategy is deemed the most appropriate [17]. The data captured from an experiment involving Snort and Suricata deployed in Docker and VM environments would be analysed and interpreted to draw practical conclusions regarding the performance capabilities of the two IDSs.

4.6 Choices

Generally, there are two distinct methodological choices – quantitative (numeric data) and qualitative (non-numeric or descriptive data). When used alone, a study is either quantitative or qualitative – the mono method [13]. When the two approaches are used simultaneously within a single study, the mixed methods approach applies. The multi-method approach applies more than one quantitative or qualitative method [13]. For the current study, a quantitative methodological approach is the most suitable. This is because the experimental investigation is intended to be conducted within a testbed (a small network) to quantitatively measure the performance capabilities of the two IDSs within Docker containers and conventional VMware.

4.7 Time Horizon

A cross-sectional or longitudinal time horizon may be adopted. In the case of cross-sectional studies, data is collected and analysed at a particular point in time. Contrastingly, longitudinal studies entail repeatedly collecting and analysing data about the same topic over a prolonged period. Longitudinal studies are particularly necessary when a certain change over time needs to be studied [13]. Considering that the current study is not intended to study any changes over time, cross-sectional research is considered as the most suitable option.

4.8 Data Collection and Analysis Techniques/Procedures

As highlighted above, empirical data would be collected experimentally. Two test environments will be conducted to measure the performance of Snort and Suricata when running on a Docker container and a conventional VMware environment. Performance parameters include the following:

- Ease of deployment and administration.
- Resource-efficiency – CPU and RAM of the host machine.
- Frequency of alert generation.
- Frequency of false alarms.

- Logging capability.

Similar alert rules will be applied to Snort and Suricata. Then, a series of commands will be run to evaluate how these IDSs will be generating added alerts

4.9 Research deliverables

The author shall validate the research by providing a profound analysis of the data extracted from the project experiments and the project outcomes. Upon completion of this research, the author is required to deliver the following: the research must answer the thesis key questions and provide evidence for it, help the reader establish a clearer overview on the subject, understand all the pros and cons of each deployment method, self-critic the overall work completed, provides recommendations to readers and conclude whether Docker implementation outweighs VMware when it comes to IDS efficiency for the two open sources Snort and Suricata.

4.10 Resources and planning

In this section, the author shall deliver and explain all the essential information related to resources and planning. Every aspect of the resources available and the plan developed shall outline the design and implementation phase and validate the research strategy adopted [14]. A concise analysis of all the resources identified will be explained to the readers to help rationalise the planning and execution procedure and help validate the research proposed method.

Good planning lies at the heart of any successful project, it is inevitable that at some point a few problems and issues will arise throughout this project. However, the key here is to carefully set appropriate milestones and deadlines that should be monitored and updated regularly and design a good contingency plan risk assessment plan to mitigate all possible scenarios.

The author shall conduct a thorough risk analysis to help identify all potential threats that this project might face:

- Budget constraints may become an issue if the author required to build a local cluster in the lab environment where the experiment will be conducted rather than connecting to the university resources available in the comms-room.
- Software licence limitations may impact in dividing the workload between home and University, some of the software is only for academic use.
- Out of hours, access restriction to the lab may delay the progress of this project, the author shall request the necessary access approval.
- Lost or failure to back-up the project data and the experiment results may have a damaging impact on the entire project.
- IT infrastructure failure or planned maintenance may cause delays to this project.

4.11 Resources Required

Every project would require different resources, identifying these resources at an early stage can significantly impact the running of any project and help mitigate any unnecessary delays. Resources vary from hardware, software to individuals. Please see below a detailed list of the necessary resources required for this project:

- Access to the University electronic and physical materials such as journals, books, papers etc.
- The research project supervisor advice and assistance when needed.
- Access privilege to Cisco lab out of hours.
- The latest version of Snort and Suricata VM image.
- The latest version of Snort and Suricata Docker image.
- VMware software.
- Kali Linux OS.
- Ubuntu OS.

- Open vSwitch software.
- TCP replay open-source suit.
- Docker Swarm cluster running on a local lab machine.
- Four Dell machines.
- Cisco router 1941 series.
- Cisco Layer two switch 2960 series.
- Windows 10 operating system.

4.12 Gantt Chart

Gantt chart is an important and very valuable tool when it comes to project management, it provides the teams and individuals involved in any project with the visual representation of timescales, resources, activities and the progress of each task. Also, it helps improve communication, collaboration, efficient resource management, time management, promote clarity, boost flexibility and create a sense of accountability, etc. [18]. As a best practice, in this project, the author shall produce a clear Gantt chart (as illustrated in figure 2) to provides the reader with an estimated time scale of each

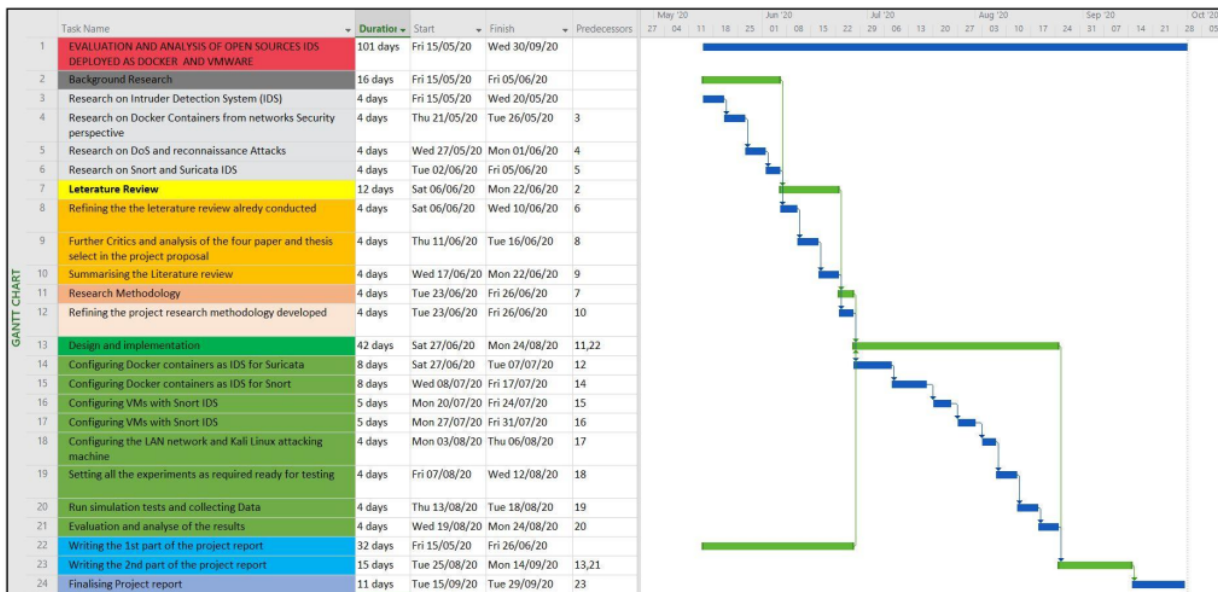


Figure 2: Project Gantt Chart

task and some clarity about the execution flow of the entire project. However, this Gantt chart is expected to endure changes and adaptation throughout the execution of this project. Please see below screenshots from the project Gantt chart developed using Microsoft project:

CHAPTER 5 - DESIGN AND IMPLEMENTAION

5.1 Overview

The project experimentation was designed to meet all the objectives in an isolated network to comply with the University rules, the author shall document both deployment approaches; Docker IDS and VMware experiments and compare the results at the end of this project. As illustrated in figures 3, two IDS servers will be set as Docker and VM to monitor the LAN network mimic by a traffic generator and an attacking Kali Linux machine. All the traffic first will be filtered using an ASA firewall with a basic In-Out filter rules configuration. As seen in figure 4 the many containers will be monitoring the network, each container would be running as sub-section of the rules that a normal IDS usually run.

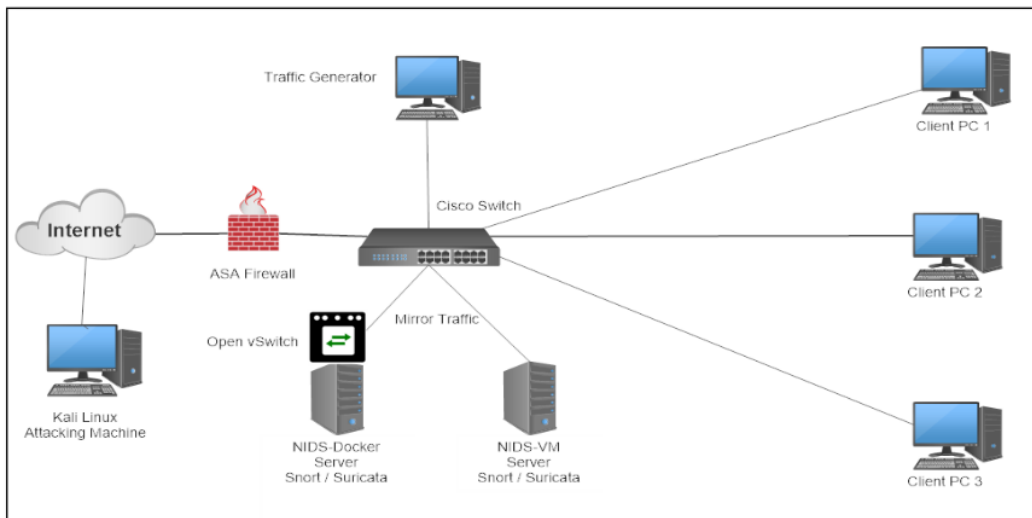


Figure 3: The project experimentation design

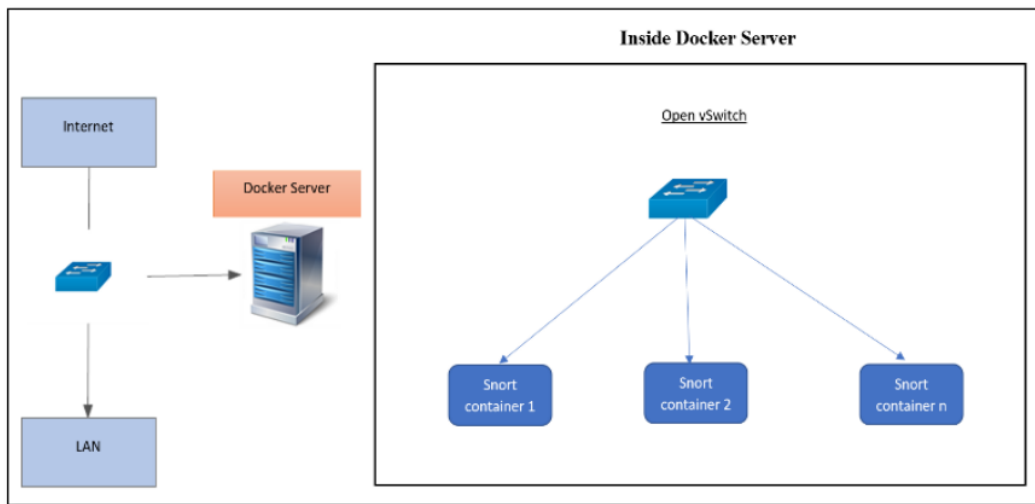


Figure 4: Inside Docker IDS Server

5.2 Ethical considerations

In every research project, major consideration and a great deal of attention should be considered when it comes to the ethical side of any project. Therefore, the author shall formally request an ethical approval from the University to implement the designed experiment in the isolated lab network environment and run different ethical cyberattacks to evaluate IDSs, the author shall explain to ethics committee all the benefits of pursuing this project and why this research is worthwhile while respecting the University ethical conduct. Besides submitting the “Application of Ethical Approval PG2-E1” form, the researcher is required to read and comprehend some important IT-related laws and regulation documents such as Computer Misuse Act 1990, Data Protection Act 1998, Serious Crime Act 2015, etc. [19] [48].

CHAPTER 6 – CONTINGENCY PLAN (Due to COVID-19 Pandemic)

It is essential to have a good contingency plan to ensure the success of any kind of project. The reality is that this unprecedented pandemic has changed our lives and caused enormous challenges to the entire planet. It has also impacted the academic journey of many students.

The initial project was set to start an experimental project using the university hardware resource available in the Cisco-Lab to analyse and study different open-source IDS deployed as Docker containers and Virtual Machines. However, the author always had a Plan B which consisted in continuing with the same experiment but using VMware software to create a virtualised topology to simulate the planned experiment in order to overcome the hardware access problem due to the long-term university campus closure. With all these challenges and in addition to the great impact of the pandemic on the author's family life, this project has been shifted from being an experiment dealing with quantitative data into a research Taxonomy on the use of IDS as Docker and VMware. Doing the Taxonomy research was not the author's preferred option.

CHAPTER 7 – A TAXONOMY ON IDS IMPLEMENTATIONS USING DOCKER AND VMWARE

7.1 VMware Technology

Virtualisation generates a virtual machine that virtualises the abovementioned IT layers. In this manner, one or more virtual machines can share physical hardware resources. Three criteria need to be satisfied in virtualisation. The first aspect is that an equivalent environment be made available by virtualisation to execute a program that is comparable to an original system. The next criterion is that the virtualisation be able to offer a virtualised resources' secured control. Finally, virtualisation need to be able to achieve ideal performance in completing privileged instructions through hardware or software support [24, 25]. VMware has provided the benefits of virtualisation since 1998 to most laptops, desktops and server deliveries. This technology has been used to create virtual infrastructure to address IT challenges that include the eradication of server sprawl, optimisation of system testing and development, ensuring continuity of business and securing end-user units such as unmanaged laptops, workstations and personal computers [23].

VMware enables running of unaltered OS as an application at the user level. Crashing, modifying, rebooting and reinstalling of the OS that is working inside VMware can be done without impacting other applications that are functioning in the computer. VMware allows the creation of virtual machine which serves as the environment for virtual hardware execution. Simultaneous usage of multiple virtual machines and individual virtual machines can provide isolated systems from the underlying system's activities and hardware [22]. A virtual machine having a VMware infrastructure can be likened to a physical server. The only difference is that a virtual machine is actually an array of software files rather than a box of electronics. A VMware virtual machine typically involves the following components: Ethernet, video card, sound card, IDE controller, SCSI controller, USB device, mouse, keyboard, floppy disk, CD or DVD, serial/com ports and parallel ports [26].

VMware has several virtualisation components for various purposes. VMware hypervisor is a software that works with the physical computer's underlying resources for virtualising the physical computer. VMware Workstation virtualises desktop operating systems. VMware Horizon is a virtual desktop integration tool. It enables troubleshooting and configuration of the desktop OS without the need for onsite visit or using remote access. VMware vSphere is a platform for enterprise virtualisation. VMware vCentre enables the management of deployment of virtual machines over a vast amount of host servers. VMware vSAN, VMware NSX, VMware HCX and VMware Cloud are for the virtualisation of the remainder of the data centre [20]. These are just a few of the VMware tools that are available for various virtualisation objectives.

There are several benefits that come with using VMware products. One is that it enhances return on investment since these tools allow the maximisation of the use of resources of a physical computer. Another is it enables space and energy efficiency by running on fewer physical servers and multiple applications. Also, VMware is the leading developer of virtualisation tools. VMware has a 75,000-strong partner network that support its more than 500,000 customer bases [20].

7.2 Docker Technology

Docker is known as the most common container tool as well as the most widely reported Containerisation technology in scientific literature [27]. Docker is distinct from virtualisation based on two aspects: first is the absence of an overhead that arises due to the emulation of a complete virtual machine-specific OS and the second is that a similar kind of OS can be run on both the container and the host. However, containers like Docker and virtual machines deal with a similar challenge which is isolating and controlling the application components [31]. An open-source platform, Docker that execute applications while making the development and distribution of this process easier. Under this technology, a container, which is a standard form, is created by packaging the auxiliary dependencies together. Containers are run on top of the kernel of the OS while isolated. This allows Docker to

provide a lightweight and rapid environment for the efficient running of code while offering a conducive space for the pre-production testing of code from the computer [32].

There are four main components that comprise Docker. The first component is the client and server. The Docker client communicates with the Docker daemon or server; the latter performs all the function. The Docker client or daemon can be run on a similar host or it can also be run on a separate host by linking the Docker client to its remote daemon. The second component are the Docker images. In the Docker environment, images serve as the building blocks. Containers are launched from the images. In the life cycle of Docker, images constitute the build part. Union file systems are used by Docker images in a layered format and are gradually constructed using a sequence of instructions [30].

The third component of Docker are registries. Registries are where the built images are stored by Docker. Private and public registries are the two kinds of registries. The Docker Hub is a public registry for images that is run by the Docker company. In the Docker Hub, users can make an account for storing and sharing images. The fourth and final component of Docker are containers. A container is an execution environment that is also a range of standard operations and an image format. Every container holds a cargo in the form of a software image. A software image enables the performance of a set of operations. A Docker container can either be an application server, a database or a web server [30].

Applications of Docker are numerous. It can help expedite, streamline and make more effective the build workflow and local development. Docker can also enable application and service stand-alone operation over different settings. Another use of Docker is running continuous integration tests in isolated instances. Docker can likewise be used to build and test, pre-deployment, architectures and intricate applications on a local host. It has also been applied for Platform-as-a-Service infrastructure construction as well as for Software-as-a-Service applications and other processes [30].

7.3 IDS as VMware or Docker

Network traffic has been increasing exponentially in the data centre. This has made IDS a suitable layer of security to the data centre. VMware has been applied as an IDS strategy through the intrinsic security framework. Intrinsic security is one that is integrated into infrastructure and spans the IT environment. VMware NSX is a prime example of this framework. NSX enables this function in two approaches: through east-west security and network virtualisation. East-west security allows the implementation of security policies in the hypervisor for the data centre. On the other hand, network virtualisation separates the physical network from the traffic flow management. These two approaches combine to provide intrinsic security while conferring flexibility to the network design of the data centre [33].

NSX constructs security into the infrastructure of the network virtualisation where this feature always exists and need not to be deployed. Given that NSX has a distributed architecture, security capacity is not limited by a centralised control. This also gives NSX the ability to see all every workload and application, thus enabling it to preset security policy regulation and to monitor workload lifecycle [33].

The engines of NSX Distributed IDS/IPS is based on Suricata; the former using the latter to create a runtime environment to the IDS/IPS engines as well as management functionality. NSX embeds within the firewall the functionality of IDS/IPS. The firewall allows passing of all traffic first before the configuration-dependent inspection by IDS/IPS of the traffic occurs. Since the firewall and IDS/IPS functionality are co-located, the security policies of the network can be simplified in terms of enforcement and expression. In the hypervisor's kernel, the firewall module is connected to the engines of NSX Distributed IDS/IPS which are in the user space. Traffic is inspected in the firewall where the hypervisor receives traffic, facilitating communication between applications. The traffic is then sent to the user space-located IDS/IPS module [33].

Virtualisation of IDS such as using VMware provides a range of benefits. The first advantage of virtualising IDS is individual scalability. This is because different components of IDS require varying levels of resources. Thus, virtualised IDS can offer flexibility and cost-efficiency in running security. The second benefit is it allows customisation which is achieved by laying down different policies. The third advantage of virtualising IDS is the ability of allowing reusability or shareability. This makes IDS save on resources and make maintenance and management of the system easier [34].

For Docker, the advantages of this tool in relation to security measures such as IDS can be derived from its various mechanisms such as the running of isolated containers. Isolation mechanisms can be implemented in a variety of ways based on kernel features of the OS. Examples of these features include control groups, namespaces and capabilities. To bolster the isolation steps specific tools such as App-Armor, Security-Enhanced Linux and Linux Security Modules can be utilised. For reducing system calls, the Secure Computing code of the Linux kernel can be applied [35].

Namespaces are created by Docker when running a container to serve as isolated workspaces. Mounting file systems, networking and isolation of processes are the functions of namespaces. This prohibits an internal process from visualising files, processes and others in the container's external environment. Control groups function by limiting resources available to the container to work in the system. Capabilities enables the running of container processes based on specific privileges [35].

7.4 Taxonomy on the use of IDS as VMware or Docker

The use of IDS and its effectiveness can be tested in a simulation after the internal network has already been accessed by an attacker and has made a list of targets followed using the IP protocol to conduct internal network scans. The attacker uses nmap for this latter task. Nmap is port-scanning tool that is open source, providing information on ports of a device, their services and the operating system running. When a virtual machine like that that runs VMware Workstation has an application that has been compromised by an attack, only the OS of the virtual machine is affected. This allows for a

sandboxed environment that is more secure [36]. Docker, on the other hand, improves the stability and portability of the various device software. The running software can be isolated from other processes when container images are converted into new containers. Therefore, when the application has been compromised, Docker enhances the security by limiting the access of the attacker to the affected container only. However, containers can be compromised by attackers to infect other containers when not properly configured [36].

It has been noted that the efficiency of IDS tools is dependent on the availability of resources of virtual machine such as VMware. More specifically, that the amount of RAM and CPU core allotted for the virtual machine is correlated with the outcome of IDS performance have been observed when a single Suricata instance was evaluated with varying virtual machine resources [37]. When multiple instances of Suricata was used in multiple container scenario, the study found an increase in alerts with proportional Suricata instance, accompanied by increasing CPU usage for every instance of Suricata. Deploying a number of security instances enables the application to increase its data rate management capability. These results indicate that multiple IDS can be feasible when launched parallel as dockerised instances. When several instances are underway, Suricata is found able to distribute load as a response [37]. VMware Workstation 6.5 has been used by Day and Burns in 2011 to test the function of Suricata and Snort as a visualisation platform [38]. This choice of platforms was due to VMware Workstation 6.5's better disk and IO performance, according to the authors, when compared to Virtual PC and Virtual Box. This virtualisation tool was coupled with the operating system Ubuntu 10.4 TLS 32 bit. Suricata and Snort were setup to execute similar rule-sets. The classification of Snort was different from that of Suricata in that the former uses 174 pre-processors and 134 decoder rules. AcidBase, MYSQL and Barnyard were all used for both Suricata version v1.0.2 and Snort version v2.8.4.2 [38].

A strategy for applying IDS in virtual machine such as VMware is exemplified by virtual machine introspection (VMI). VMI is the concept that supports the intrusion detection that is referred

to as out-of-the-box. Inspection of the virtual machine state is performed by relocating outside the virtual machine the inspection module. In order to detect any intrusion, the guest system's software that is running is analysed externally. This enables the continuous detection of malware event in the event of an intrusion which is an advantage for this technique [39]. An example of the virtual machine-based intrusion detection is the Xen Access which is a virtual memory introspection technique. Virtual disk monitoring and virtual memory inspection is provided for by the monitoring library of Xen Access. This method does need any modification on the OS or virtual machine. Xen Access can visualise another virtual machine's memory which is used to determine OS data from the target [40].

7.5 Comparison and Analysis of Previous Studies

Prior studies have explored the potential of using open-source IDS with virtualisation and Containerisation tools such as VMware and Docker, respectively. For example, the study of Day and Burns in 2011 analysed the accuracy and performance of Suricata and Snort on four cores using virtual machines run by VMware [38]. The authors were able to assess the performance of the two IDS technologies on the limited number of cores. However, it was noted that there was a need to conduct the same performance tests using significantly more numerous cores to properly evaluate the Suricata and Snort accuracy and performance [38].

White, Fitzimmons and Matthews [41] made an apples-to-apples comparative analysis of Suricata and Snort as reported by Day and Burns in 2011 [38]. This time, White et al. performed the tests using cores beyond four. The two systems' default performance as well as advanced configurations were compared. In addition to raising the CPU cores, they also tested the effect of multi-instance Snort configuration. They also modified the used rulesets as well as the processed workloads. The authors demonstrated that this change resulted in a significant enhancement in Suricata's performance; in fact, they claimed improvements reaching 20X.

Another study related to that of Day and Burns [38] and of White et al. [41] is the study by Albin [42]. To compare the performance of Suricata and Snort, Albin conducted three experiments involving testing ruleset, static pcap files and live network traffic with the use of Pytbull. For the virtual machine host, Albin used VMware ESXi. The author assessed the detection accuracy, memory requirements and speed of the systems in the different experiments. Albin concluded that larger traffic volumes can be handled by Suricata than Snort given the same level of accuracy.

Unlike the study by White et al. [41], Albin's work involved changing component and unquantified background traffic. This difference could have affected the reproducibility of Albin's results. Albin demonstrated that traffic volume and traffic type were not consistent across the experiments as shown by the variability in the network traffic during Suricata and Snort comparison runs. Also, according to White et al. [41], the use of Pytbull by Albin while linked to a running network could have rendered the traffic to a condition of variability as observed in the network [41].

There have been more studies relating to the security of virtual machines through IDS as evidenced by publications mentioned above. However, on the part of container technologies such as Docker, there have been limited research. This is even though the introduction of IDS in Dockerised and similar systems is not dependent on any background knowledge regarding the application within the container. It also does not need modification of the host kernel or the container [43].

The above observation has also been echoed by Hickman in 2018 who noted that conducting IDS and analysis in Dockerised world has been limited in the literature [44]. The author, however, contends that some reports are available regarding the intrusion detection methods, container security application and determination of build area. He adds several references are available pertaining to traditional systems and methods that can be utilised when making comparisons with novel container network techniques [44, 45].

According to Chandrasekaran [46], although Docker is a widely used and publicised container technology, it has its limitations in relation to IDS security aspects. Although the author was able to

create from applications and libraries new Docker images and found Docker a convenient tool for building, deploying and working among containers, he claims that the container technology was not appropriate for their system due to potential security issues associated with isolation using Docker containers.

7.6 Using Open-Source IDS as VMware or Docker

This current research exploring the potential of using open-source IDS as VMware or Docker can provide significant contributions to the field of security in virtual machine and containerised systems. As the aforementioned studies indicate, there are still large gaps when it comes to network security, especially IDS with its usability challenges [47], as it involves these fast-evolving technologies such as VMware and Docker.

The studies of Day and Burns [38], White et al. [41] and Albin [42] has not only demonstrated the potential of applying IDS tools such as Suricata and Snort with virtual machines and containers such as VMware and Docker. These three studies have also successfully evaluated the accuracy and performance of these IDS tools in the virtual machine and container environments. Day and Burns [38] highlighted the need to conduct performance tests using more numerous cores to properly evaluate the Suricata and Snort accuracy and performance. White et al. [41] thus increased the CPU cores beyond four and modified the used rulesets as well as the processed workloads, thereby greatly enhancing Suricata's performance. Meanwhile, Albin was able to assess the detection accuracy, memory requirements and speed of the systems in the different experiments comparing the performance of Suricata and Snort. Albin concluded that larger traffic volumes can be handled by Suricata than Snort given the same level of accuracy [42]. This current research can build on the results of these previous studies. Increasing the CPU cores higher than that what was formerly done can reveal more significant data for this present study. Also, modifying the rulesets and varying the workload processes should provide more useful insight.

As has been discussed in the studies of Abed [43], Hickman [44] and Chandrasekaran [46], research on open-source IDS applications in Dockerised environments has been limited. Therefore, this is where this current research can contribute considerably. Addressing the security challenges inherent in Docker systems need to be prioritised with an emphasis on IDS tools Suricata and Snort. This means testing and evaluating the various configurations and rulesets, as well as workloads, are essential in determining the potential solutions to issues that have been previously floated. For instance, the interactions between containers and the importance of container isolation during IDS operation can be a point of focus or start in this current research. Findings in such experiments can provide invaluable information for present development and aid future research as well.

CHAPTER 8 – SUMMARY & CONCLUSION

The use of Intrusion Prevention System (IPS) and Intrusion Detection Systems (IDS) is widely applied in various businesses and organisations of all sizes. The applications are majorly employed to analyse and monitor network security against any violation of security policies for unwanted traffic. This paper aimed at establishing an understanding of how the Intrusion Detection Systems (IDS) could be deployed as a Docker container and Virtual Machine (VM) to analyse and monitor network security of small and medium enterprises. The study mainly focused on evaluation and deployment of some well-established open-source IDS systems using the new emerging trend that is Docker container technology in comparison to classic VM approach. The key concept in the research was to establish whether containerized IDS would provide better protection to the network security against DoS and reconnaissance attacks comparing to a similar experiment using the VM approach.

As an emerging trend, the Docker technology has revolutionized the information technology industry, the open-source container areas provide developers with a comprehensive flexibility while testing, running and deploying their applications in an isolated environment. Due to ease in management, flexibility and lightweight across different operating systems, for the past five years the

Docker has become one of the fast-growing trends in information technology. The system has greatly advanced capability of running multiple Suricata/Snort IDS in parallel in a properly isolated environment to help solve issues related to engine becoming over-loaded and packet been dropped.

This research was based on the IDS performance in a Paytrail company's systems. The company mainly deals with payments through e-commerce and internet services thus exposing it to potential threats from hackers with malicious intents. The thesis of the project focused on integration, trustworthy of the open-source and performance expected of the company's systems. Thus, mainly the methodology was to establish solutions and threats on the company by comparing IDS performance and different solutions based on the highlighted three factors above. Some of the challenges expected while performing this project include Docker complexity and heavy Linux coding to configure the Suricata and Snort containers to perform a parallel analysis to the network traffics. However, looking at it broadly, it will go a long way in developing the researcher's skills on handling various things.

The thesis gives a comparison among three Intrusion Detection Systems that are ranked based on their importance in facilitating Small and medium enterprises. Due to very rapid technological advancements, some of the tools that are applied in threat elimination must be developed at very fast and significant rate to help curb the threats. It was noted that various firms that take security precautions would prefer implementation of IDS in their systems. Thus, this project tasked various components of the IDS to evaluate its performance based on its performance. In monitoring and evaluation of the threats various cases were sampled to assess them based on the impacts they had on system. This was imported from a progressive basis from solving an existing company's threats.

CHAPTER 9 – RECOMMENDATIONS

The application of virtual machines is prone in various organizations worldwide. This is because VM is known to particularly execute very complex tasks. However, there is a likelihood for the users of VM to shift its application to launching of small applications which in essence does not need the virtual machines thus making its system inefficient. Thus, there is need for use of launchers that are more suitable for such tasks, the Docker is an example of such lightweight application that can be used to efficiently launch such small tasks applications more efficiently and very fast. Thus, institutions need to invest more on the Docker technologies with an aim of improving their efficiency as detailed by the architecture of the Docker Container.

In performance evaluation of both virtual machines and Docker engine particular benchmarking tools should be used. These benchmarking tools include Sysbench apache benchmarking and Phoronix. Benchmarking is important because essentially it helps in evaluating the performance of the machine based on its output. It is recommended that the systems need to be done for comprehensive benchmarking one in a while to ensure that it is performing optimally and has no threats. Benchmarking essentially refers to comparing the performance of the machine to other simulation processes that need to map its output in a particular manner.

Docker machines are the best because essentially in every test that was carried out while comparing virtual machines and Docker machines it was realized that Docker performed better in all times. This could be attributed to a layer of QEMU in the virtual machine that made it inefficient than the Docker containers. Thus, organizations should essentially go for Docker containers over virtual machines because of its comprehensive mechanisms on handling launching of both small and large applications with ease.

The rise in demand of Docker containers puts it in a position of threat especially from external persons with malicious intents. More security programs need to be imbedded into the systems to ensure

that the container is work properly. More simulates and benchmarking need to be done to ensure any variants are eliminated before finding ways into the system. The application is widely used by companies worldwide, especially these companies that hand cash transfers or very important information. The rise in advancement of technology implies that the threat levels within these companies are categorically easier to manage though may also pose to be a challenge in the long run. Organizations worldwide need to ensure they have the latest emergence in technology to address such challenges.

Docker containers tend to have some security constraints that may particularly interfere with their performance; thus, it is vital that in the future more work be done in highlighting the security constraints within the containers that may in the end hinder progressive performance of the Docker containers. It is equally important to do more research on virtual machines on how to make them better, maybe through elimination of QEMU the inefficiencies witnessed in the machines could largely be reduced.

REFERENCES

- [1] M. Bharati and S. Tamane, "Defending against bruteforc attack using open source - SNORT," Proc. Int. Conf. Inven. Comput. Informatics, ICICI 2017, no. Icici, pp. 903–907, 2018.
- [2] J. T. Rodfos, "UNIVERSITY OF OSLO Department of Mathematics," Thesis, no. may, 2005.
- [3] J. Timofte, "Intrusion Detection using Open Source Tools," Inform. Econ. J., vol. 12, no. 2, pp. 75–79, 2008.
- [4] B. Q. Musawi, "Mitigating DoS / DDoS attacks using iptables," no. January 2012, 2016.
- [5] F. Hock and P. Kortiř, "Commercial and open-source based Intrusion Detection System and Intrusion Prevention System (IDS/IPS) design for an IP networks," ICETA 2015 - 13th IEEE Int. Conf. Emerg. eLearning Technol. Appl. Proc., pp. 1–4, 2016.
- [6] R. F. Erbacher and R. Pino, "Open Source Software Tools for Anomaly Detection Analysis," no. April, 2014.
- [7] O. B. Lawal, A. Ibitola, and O. B. Longe, "Analysis and Evaluation of Network-Based Intrusion Detectionork Using Snort Freeware," African J. Comput. ICT African J. Comput. ICT Ref. Format O.B. Lawal, vol. 6, no. 2, pp. 169–184, 2013.
- [8] H. N. Tran, "A Dynamic Scalable Parallel Network-based Intrusion Detection System using Intelligent Rule Ordering," 2017.
- [9] Y. Al-Dhuraibi, F. Paraiso, N. Djarallah, and P. Merle, "Autonomic Vertical Elasticity of Docker Containers with ELASTICDOCKER," IEEE Int. Conf. Cloud Comput. CLOUD, vol. 2017–June, pp. 472–479, 2017.
- [10] M. Hänninen, "Open source intrusion detection systems evaluation for small and medium-

sized enterprise environments,” 2019.

- [11] V. Nagadevara, “Evaluation of Intrusion Detection Systems under Denial of Service Attack in virtual Environment,” 2017.
- [12] M. Saunders, P. Lewis, and A. Thornbill, *Research Methods for Business Studies*. 2000.
- [13] by R. Burke Johnson and A. Onwuegbuzie, “Mixed methods research: A research paradigm whose time has come,” *Educ. Res.*, vol. 33, no. 7, pp. 14–26, 2004.
- [14] H. Berger and A. Jones, “Cyber security & ethical hacking for SMEs,” *ACM Int. Conf. Proceeding Ser.*, vol. Part F130520, 2016.
- [15] J. Clonts, “The concept of reliability as it pertains to data from qualitative studies,” *Annu. Meet. South West Educ. Res. Assoc. Houston, TX.* , 1992.
- [16] T. Linda, “The strengths and weaknesses of quantitative and qualitative research: what method for nursing?,” 1994.
- [17] K. Hinkelmann Hans Friedrich Witschel, “How to choose a research methodology,” pp. 1–49, 2015.
- [18] H. Y. Ong, C. Wang, and N. Zainon, “Integrated earned value Gantt chart (EV-Gantt) tool for project portfolio planning and monitoring optimization,” *EMJ - Eng. Manag. J.*, vol. 28, no. 1, pp. 39–53, 2016.
- [19] R. A. Lundin, “Ethical considerations in doing and disseminating project management research,” *Int. J. Manag. Proj. Bus.*, vol. 4, no. 1, pp. 150–156, 2011.
- [20] IBM, “VMware,” IBM Cloud Education, available at:
<https://www.ibm.com/cloud/learn/vmware>, accessed: 22 January 2021.
- [21] Ward, B., “The book of VMware - The complete guide to VMware Workstation,” Canada,

William Pollock, 2002.

- [22] Ali, I. and N. Meghanathan, "Virtual machines and networks - Installation, performance, study, advantages and virtualization options," International Journal of Network Security and Its Applications, vol. 3, no. 1, 2011.
- [23] VMware, "Network throughput in a virtual infrastructure," VMware White Paper, available at: <http://www.cse.psu.edu/~buu1/teaching/spring06/papers/vmware-tput.pdf>, accessed: 22 January 2021.
- [24] Lee, H., "Virtualisation basics: Understanding techniques and fundamentals," School of Informatics and Computing Indiana University 815 E 10th St. Bloomington IN 47408 available at: <http://dsc.soic.indiana.edu/publications/virtualisation.pdf>, accessed: 22 January 2021.
- [25] Bouali, L., Abd-Elrahman, E., Bouzefrane, S. and H. Afifi, "Virtualization techniques: Challenges and opportunities," International Conference on Mobile, Secure and Programmable Networking Project, DOI: 10.1007/978-3-319-50463-6_5, June 2016.
- [26] VMware, "Building the virtualized enterprise with VMware infrastructure," VMware White Paper, available at: https://www.thespecialist.com/documents/wp_vmware_enterprise.pdf, accessed: 22 January 2021.
- [27] Vase, T., "Advantages of Docker," University of Jyväskylä, available at: <https://jyx.jyu.fi/bitstream/handle/123456789/48029/1/URN%3ANBN%3Afi%3Aju-201512093942.pdf>, accessed: 23 January 2021.
- [28] Casalicchio, E. and S. Iannucci, "The state-of-the-art in container technologies: Application, orchestration and security," Concurr. Comput. Pract. Exp., <https://doi.org/10.1002/cpe.5668>, 2020.

- [29] Pahl, C., Brogi, A., Soldani, J. and P. Jamshidi, "Cloud container technologies: A state-of-the-art review," IEEE Transactions on Cloud Computing, May 2017.
- [30] Turnbull, J., "The Docker Book," available at:
<http://lsi.vc.edu.es/pablogn/docencia/manuales/The%20Docker%20Book.pdf>, accessed: 23 January 2021.
- [31] Bigstep, "What is Docker and why use it?" Bistep, available at: <http://www.odbms.org/wp-content/uploads/2014/11/What-is-Docker.pdf>, accessed: 22 January 2021.
- [32] Rad, B.B., Bhatti, H.J. and M. Ahmadi, "An introduction to Docker and analysis of its performance," IJCSNS, vol. 17, no.3, 2017.
- [33] VMware, "VMware NSX Distributed IDS/IPS," VMware White Paper, available at:
<https://www.vmware.com/content/dam/digitalmarketing/vmware/en/pdf/products/nsx/vmw-nsx-distributed-ids-ips-tech-white-paper-dec.pdf>, accessed: 23 January 2021.
- [34] Li, H., Zhang, N., Hu, H. and Y. Park, "Towards effective virtualization of intrusion detection systems," SDN-NFV Sec'17, 22 March 2017, Arizona, USA.
- [35] Abdelmassih, C. "Container orchestration in security demanding environments at the Swedish Police Authority," KTH Royal Institute of Technology, available at: <https://www.diva-portal.org/smash/get/diva2:1231856/FULLTEXT02.pdf>, accessed: 23 January 2021.
- [36] Madison, Z.D., "Honeyhive - A network intrusion detection system framework utilizing distributed Internet of Things Honeypot sensors," Theses and Dissertations, 3179, available at:
<https://scholar.afit.edu/etd/3179>, accessed: 23 January 2021.
- [37] Ranaweera, P., Imrith, V.N., Liyanage, M. and A.D. Jurcut, "Security as a Service platform leveraging multi-access edge computing infrastructure provisions," ICC 2020 - 2020 IEEE International Conference on Communications (ICC), Dublin, Ireland, 2020, pp. 1-6, doi:

10.1109/ICC40277.2020.9148660.

- [38] Day, D.J. and B.M. Burns, "A performance analysis of Snort and Suricata network intrusion detection and prevention engines," The Fifth International Conference on Digital Society, February 2011.
- [39] Fesl, J., Gokhale, V., Dolezalova, M., Cihak, D. and J. Janecek, "Cloud infrastructures protection technique based on virtual machines live migration," ACIT 2018, 1-3 June 2018, Czech Republic.
- [40] Mahmood, I., "A survey paper on hypervisor-based cloud intrusion detection system (IDS)," Acta Scientific Computer Sciences, vol. 2, issue 2, 2020.
- [41] White, J.S., Fitzimmons, T.T. and J.N. Matthews, "Quantitative analysis of intrusion detection systems: Snort and Suricata, Proceedings of SPIE - The International Society for Optical Engineering, 2013.
- [42] Albin, E., "A comparative analysis of the Snort and Suricata intrusion-detection systems, Calhoun: The NPS Institutional Archive, available at:
<https://core.ac.uk/download/pdf/36699186.pdf>, accessed: 23 January 2021.
- [43] Abed, A.S.O., "Securing cloud containers through intrusion detection and remediation," Virginia Polytechnic Institute, available at:
https://vtechworks.lib.vt.edu/bitstream/handle/10919/87730/Abed_AS_D_2017.pdf?isAllowed=y&sequence=1, accessed: 24 January 2021.
- [44] Hickman, A., "Container intrusions: Assessing the efficacy of intrusion detection and analysis methods for Linux container environments," SANS Institute, available at:
<https://www.sans.org/reading-room/whitepapers/detection/container-intrusions-assessing-efficacy-intrusion-detection-analysis-methods-linux-container-environments-38245>, accessed:

24 January 2021.

- [45] Durairaju, S.S., “Intrusion detection in containerized environments,” San Jose State University, available at:
https://scholarworks.sjsu.edu/cgi/viewcontent.cgi?article=1638&context=etd_projects,
 accessed: 24 January 2021.
- [46] Chandrasekaran, S., “Towards a low-memory-footprint, container-based IoT security gateway,” Carnegie Mellon University, available at: <http://www.andrew.cmu.edu/user/sanjayc/main.pdf>,
 accessed: 23 January 2021.
- [47] Welinger, R., Hawkey, K., Muldner, K., Jaferian, P. and K. Beznosov, “The challenges of using an intrusion detection system: Is it worth the effort?” Symposium On Usable Privacy and Security (SOUPS) 2008, July 23-25,2008, Pittsburgh, PA, USA
- [48] Ouzahra Nabil, UWTSD MSc Project Proposal, 2020.

List of Figures

Figure 1: The research onion by Saunders.....	18
Figure 2: Project Gantt Chart.....	23
Figure 3: The project experimentation design	24
Figure 4: Inside Docker IDS Server.....	25

Index Terms:

IDS: Intrusion Detection System

IPS: Intrusion Prevention System

NIDS: Network Intrusion Detection System

HIDS: Host Intrusion Detection System

DoS: Denial of Service (attacks)

NBA: Network Behaviour Analysis

Docker: New emerging trend technology (Set of platforms as a service (PaaS))

VM: Virtual Machine

VMware: Virtualisation Software

Snort: Open-source IDS

Suricata: Open-source IDS

OSSEC: open-source IDS

VRT: Vulnerability Research Team (rules set update)

CPU: Central Processing Unit

RAM: Random Access Memory

VMware vSphere: Cloud computing virtualisation Platform also known as VMware Infrastructure.

VMware vCentre: Centralised management tools for the VMware.

VMware vSAN: Virtualised SAN (Storage Area Network)

VMware NSX: Network Virtualisation VMware

VMware HCX: Mobility Application to simplify migration and optimise workloads

Sysbench Benchmarking: Benchmarking suit for Linux

Phoronix Benchmarking: Open-Source automated benchmarking tool

QEMU: Quick Emulator Open-Source tool for Hardware Virtualisation

Nabil Ouzahra MSc Project

GRADEMARK REPORT

FINAL GRADE

/100

GENERAL COMMENTS

Instructor

PAGE 1

PAGE 2

PAGE 3

PAGE 4

PAGE 5

PAGE 6

PAGE 7

PAGE 8

PAGE 9

PAGE 10

PAGE 11

PAGE 12

PAGE 13

PAGE 14

PAGE 15

PAGE 16

PAGE 17

PAGE 18

PAGE 19

PAGE 20

PAGE 21

PAGE 22

PAGE 23

PAGE 24

PAGE 25

PAGE 26

PAGE 27

PAGE 28

PAGE 29

PAGE 30

PAGE 31

PAGE 32

PAGE 33

PAGE 34

PAGE 35

PAGE 36

PAGE 37

PAGE 38

PAGE 39

PAGE 40

PAGE 41

PAGE 42

PAGE 43

PAGE 44

PAGE 45

PAGE 46

