

**OPTIMIZATION OF NETWORK DEVICE HARDENING IN A
MULTIVENDOR ENVIRONMENT**

BY

ALI BELLO IMOUKHUEDE

A Thesis Presented to the
DEANSHIP OF GRADUATE STUDIES

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS

DHAHRAN, SAUDI ARABIA

In Partial Fulfillment of the
Requirements for the Degree of

MASTER OF SCIENCE

In

COMPUTER ENGINEERING

DECEMBER 2023

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS
DHAHRAN 31261, SAUDI ARABIA

DEANSHIP OF GRADUATE STUDIES

This thesis, written by **ALI BELLO IMOUKHUEDE** under the direction of his thesis adviser and approved by his thesis committee, has been presented to and accepted by the Dean of Graduate Studies, in partial fulfillment of the requirements for the degree of **MASTER OF SCIENCE IN COMPUTER ENGINEERING**.

Thesis Committee



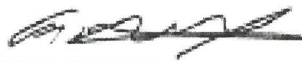
Prof. Tarek Rahel Sheltami (Adviser)



Dr. Ashraf S. Hasan Mahmoud (Member)



Dr. Abdulaziz Yagoub Barnawi (Member)



Prof. Aiman Helmi El-Maleh
Department Chairman



Prof. Suliman Saleh Al-Homidan
Dean of Graduate Studies

02 - 01 - 2024

Date



©Ali Bello Imoukhuede
2023

Dedication

This work is dedicated to my family and friends, whose unwavering support and encouragement have been the driving force behind this research endeavor. Your belief in me has been my greatest motivation.

ACKNOWLEDGMENTS

First off, I would like to start by thanking Allah the almighty, who in his infinite mercy provided me with the chance to secure the KFUPM graduate scholarship, to begin this wonderful journey. Indeed, I was lucky enough to have been tutored by excellent instructors at KFUPM and having Professor Tarek Rehil Sheltami among them as my thesis advisor is something I could not have asked for better. Further, I was fortunate to have Dr. Ashraf S. Hasan Mahmoud and Dr. Abdulaziz Yagoub Barnawi as part of my thesis committee. To my friends, colleagues, and relatives whose list seems to be infinite, I appreciate your support, prayers, and well wishes in all aspects of my life. My acknowledgment section will be incomplete without making a mention of the support and consistent prayers that I have received from my parents and siblings throughout all the academic and non-academic adventures that I have been in so far in my life.

TABLE OF CONTENTS

ACKNOWLEDGMENTS	v
LIST OF TABLES	viii
LIST OF FIGURES	ix
ABSTRACT (ENGLISH)	x
ABSTRACT (ARABIC)	xii
CHAPTER 1 INTRODUCTION	1
1.1 Overview	1
1.2 Research Objectives	2
1.3 Research Contribution	3
CHAPTER 2 BACKGROUND	5
2.1 Hardening	5
2.2 Center for Internet Security (CIS)	6
2.3 Network Automation	8
2.3.1 Common Network Automation Tools and Frameworks	9
2.4 The Nornir Framework	10
CHAPTER 3 LITERATURE REVIEW	12
3.1 Overview	12

CHAPTER 4	METHODOLOGY	15
4.1	Automation Framework Setup	15
4.2	Simulation	17
CHAPTER 5	RESULTS AND DISCUSSION	20
5.1	TEST 1: Hardening of a Cisco router	20
5.2	TEST 2: Hardening of 6 Cisco routers	24
5.3	TEST 3: Hardening a Fortigate firewall	27
5.4	TEST 2: Hardening a Juniper router	32
5.5	TEST 5: Hardening a multivendor network	35
CHAPTER 6	CONCLUSION AND FUTURE WORK	38
APPENDIX A	CIS PROFILES	42
A.1	LEVEL 1	42
A.1.1	Management Plane	42
A.1.2	Control Plane	43
A.1.3	Data Plane	43
A.2	LEVEL 2	43
A.2.1	Management Plane	43
A.2.2	Control Plane	44
A.2.3	Data Plane	44
REFERENCES		44
VITAE		57

LIST OF TABLES

4.1	Hardware Specifications	19
4.2	Software Specifications	19
5.1	Fortigate Hardening Categories and Elapsed Time	31
5.2	Juniper Hardening Categories and Elapsed Time	34

LIST OF FIGURES

4.1	Nornir workflow	16
4.2	SSH Connection to Network Device	17
4.3	Test Scenarios with Nornir Automation Framework	18
5.1	Network Setup For Basic Router Configuration On GNS 3	20
5.2	Management, Control, and Data Plane	22
5.3	Successful Ping Test Between Workstation and Router On GNS 3	23
5.4	Elapsed Time for Hardening the Three Router Planes	24
5.5	Elapsed Time of Level 1 and Level 2 Profile Hardening	25
5.6	Hardening of Devices with number of workers set to 1	26
5.7	Topology Setup For Hardening Six Cisco Routers	26
5.8	Execution Time Comparims	27
5.9	Network Topology For FortiGate Hardening	28
5.10	Port1 Interface with IP Address and SSH Enabled	30
5.11	Elapsed Time Of Fortigate Hardening Categories	31
5.12	Elapsed Time of Level 1 and Level 2 Profile Hardening	31
5.13	Network Topology For FortiGate Hardening	32
5.14	em2 Interface Configuration	32
5.15	Elapsed Time Of Juniper Hardening Categories	34
5.16	Elapsed Time of Level 1 and Level 2 Profile Hardening	35
5.17	Multivendor Network Topology	36
5.18	Elapsed Time to Configure 18 Devices	37

THESIS ABSTRACT

NAME: Ali Bello Imoukhuede

TITLE OF STUDY: Optimization of Network Device Hardening in a Multi-vendor Environment

MAJOR FIELD: Computer Engineering

DATE OF DEGREE: December 2023

This research paper explores the optimization of network device hardening through network automation. Unlike previous works, it goes beyond the confines of a single vendor's environment. This provides insight regarding using network automation to harden a multivendor network topology, assessing the effectiveness and efficiency through a series of test scenarios. In our study, we accomplished comparatively lesser device hardening times, showcasing the agility and adaptability of our Python-based Nornir automation framework. A reduction of 82.18% was archived when compared with previous work. In comparison to the traditional network configuration techniques, it achieved a whopping 99.99% reduction in execution time. These results show that using a Python-based automation framework like Nornir can greatly simplify the process of hardening network devices while improving security and reducing the possibility

of configuration inconsistencies. The significance of our findings will be of great importance to network security professionals. By achieving such substantial reductions in device hardening times, our research underscores the potential for organizations to improve their network security swiftly and efficiently.

ملخص الرسالة

الاسم: علي بيلو إيموكويدي

عنوان الدراسة: تحسين تقوية أجهزة الشبكة في بيئة متعددة البائعين

التخصص: هندسة الكمبيوتر

تاريخ الدرجة العلمية: ديسمبر 2023

تستكشف هذه الورقة البحثية تحسين تقوية أجهزة الشبكة من خلال أتمتة الشبكة. وعلى عكس الأعمال السابقة، فهو يتجاوز حدود بيئة بائع واحد. يوفر هذا نظرة ثاقبة فيما يتعلق باستخدام أتمتة الشبكة لتقوية طوبولوجيا الشبكة متعددة البائعين، وتقييم الفعالية والكفاءة من خلال سلسلة من سيناريوهات الاختبار. في دراستنا، أنجزنا أوقات صلابة أقل نسبياً للأجهزة، مما أظهر سرعة الحركة والقدرة على التكيف لإطار عمل التشغيل الآلي *Nornir* القائم على *Python*. تم أرشفة تخفيض قدره 82.18% بالمقارنة مع العمل السابق. بالمقارنة مع تقنيات تكوين الشبكة التقليدية، فقد حققت انخفاضاً هائلاً بنسبة 99.99% في وقت التنفيذ. تظهر هذه النتائج أن استخدام إطار عمل التشغيل الآلي المستند إلى *Python* مثل *Nornir* يمكن أن ييسر إلى حد كبير عملية تقوية أجهزة الشبكة مع تحسين الأمان وتقليل احتمالية عدم تناسق التكوين. ستكون أهمية النتائج التي توصلنا إليها ذات أهمية كبيرة لمحترفي أمن الشبكات. ومن خلال تحقيق مثل هذه التخفيضات الكبيرة في أوقات تصلب الأجهزة، يؤكد بحثنا على إمكانية قيام المؤسسات بتحسين أمان شبكاتها بسرعة وكفاءة.

CHAPTER 1

INTRODUCTION

1.1 Overview

In the dynamic landscape of modern networking, the continuous evolution of technology and the exponential growth of network complexity have necessitated innovative approaches to manage and operate network infrastructures efficiently [1] [2]. Manual management and configuration of network devices have become increasingly time-consuming, error-prone, and inefficient [3] [4]. Attackers have also improved their ability to infiltrate network devices by combining numerous types of attacks against devices with weak and insecure passwords, configuration errors, or security flaws [5] [6]. The creation and availability of various hacking tools have also further simplified penetrating a network architecture. Considering the expansive nature of enterprise networks and the varied potential states they may assume, the scope of the attack surface can extend to hundreds of devices, particularly in a multivendor environment. Implementing rigorous device hardening practices facilitates the identification and resolution of vulnerabilities within the system [7]. This ensures proactively address-

ing the security loopholes before they are exploited by a threat actor. To secure the infrastructure and the data within the infrastructure, strict and consistent security policies need to be implemented [8] [9]. One approach to solve these problems is by automating the hardening process[10]. Network automation is a cost-effective strategy that configures devices rapidly, minimizes human errors, reduces costs, enhances security, and improves productivity, flexibility, and resilience [11]. This is majorly achieved by the automation of repetitive tasks and by fostering programmability and orchestration capabilities [12]. Hence, network automation is a perfect approach to implementing adequate security policies and configurations in network devices[13].

1.2 Research Objectives

The overall objective of the study is to increase network security using network automation thereby increasing speed and efficiency. The network security implementations will follow the guidelines of CIS. They will be implemented on devices across various vendors to serve as proof of concept that the automation framework can be used in a multi-vendor network architecture. To determine the performance, the results will also be compared with those of earlier studies.

Listed below are the specific objectives:

- **Objective 1:**

To automate the process of level 1 and level 2 device hardening as recommended by the CIS Benchmark

- **Objective 2:**

To optimize the automation process by exploiting multithreading.

- **Objective 3:**

To carry out hardening of various network devices (routers and firewalls) from different vendors (Cisco, Juniper, and Fortinet) all on the same network.

- **Objective 4:**

Obtain a faster performance metric for hardening devices as compared to previous research.

1.3 Research Contribution

The research makes several significant contributions to the field of network security through the optimization of network device hardening using a Python-based Nornir automation framework. These contributions are as follows:

- **Multi-Vendor Network Automation:**

The research extends beyond the limitations of a single vendor's environment, addressing the critical need for network automation solutions that can operate seamlessly in multi-vendor network topologies. This marks a departure from previous works focused on vendor-specific approaches.

- **Substantial Reduction in Hardening Times:**

One of the primary contributions is the achievement of a substantial reduction in device hardening times. The research records a reduction when compared to previous methodologies, highlighting the efficiency gains introduced by the

Nornir automation framework which supports multithreading.

- **Comprehensive Test Scenarios:**

The research contributes to the field by conducting experiments across various test cases involving Cisco routers, FortiGate firewalls, Juniper routers, and a multi-vendor network. The time metrics of implementing security level 1 and level 2 for all vendors under the scope of this research were also brought to the limelight. This comprehensive approach ensures a thorough evaluation of the Nornir framework's effectiveness in diverse network scenarios, providing valuable insights for network administrators and security professionals.

The remaining part of this paper is as follows: It begins by providing additional background information in section 1.1. In section 2, the literature of the most recent work relevant to the research topic is outlined. Section 3 presents the proposed framework and simulation set up which will be used for various test scenarios. Section 4 presents the various simulations and examines the results and performance of the proposed framework. Finally, section 5 provides the conclusions and future work.

CHAPTER 2

BACKGROUND

2.1 Hardening

To strengthen the security levels in various network devices, mobile devices, applications, operating systems, server software, desktop software, cloud providers, etc., a procedure known as "hardening" is used [14]. The possibility of unwanted access to a network's infrastructure is decreased by hardening network equipment [15] [16]. A malicious cyber actor might take advantage of flaws in network device management and configurations to establish a presence and maintain persistence within a network [17]. Adversaries are increasingly focusing on targeting specialized and embedded devices, such as routers and switches, rather than only standard endpoints [18]. They accomplish this by altering routing protocols, exploiting configuration flaws, and introducing malware into the operating systems [19]. To mitigate these attacks and infiltrations, network devices must be configured according to globally recommended standards to reduce the possibility of a successful attack [20]. These rules and guidelines which have been recommended by experts from various parts of the world form

a collective procedural network security approach to ensure that all network devices are protected.

2.2 Center for Internet Security (CIS)

The CIS is a nonprofit organization centered on enhancing cybersecurity preparation and response in the public and private sectors. It utilizes the collective support of the international IT community to defend public and private entities against threats from the internet [21]. The CIS helps to concentrate efforts on the most sensitive devices by taking the knowledge and experience of security specialists from around the world. This makes it easy to identify the specific settings that need to be changed thereby making network devices hardened. Hence, CIS helps to evaluate and maintain a security baseline for IT infrastructures [22] [23]. To protect systems against the current, growing cyber threats, CIS has created Benchmarks, a set of over 100 configuration requirements spanning 25+ vendor product families. The research focuses on three vendors which are Cisco, Juniper, and Fortinet benchmarks. CIS configuration profiles are of two levels:

Level 1:

- **Basic Security Posture:**

Level 1 profiles are designed to establish foundational cybersecurity practices.

They typically include fundamental security measures that organizations should implement to improve their overall security posture. Level 1 controls are often

considered essential for all organizations as a starting point for cybersecurity.

- **Risk Reduction:**

Level 1 profiles aim to reduce the organization's risk of common cyber threats and attacks. Controls at this level are usually basic and widely applicable to different industries and organizations.

- **Minimum Security Standards:**

Organizations implementing Level 1 profiles are expected to meet minimum security standards to address prevalent cybersecurity challenges.

Level 2:

- **Enhanced Security Measures:**

Level 2 profiles build upon Level 1 by introducing additional, more advanced security measures. Controls at this level are designed to provide enhanced protection against a broader range of cyber threats.

- **Increased Complexity:**

Controls in Level 2 profiles may be more complex and require a higher level of technical expertise to implement and manage effectively.

- **Tailored to Organization's Specifics:**

Level 2 profiles may allow for more customization to address the specific needs and risks of individual organizations. Controls may be adapted based on the organization's industry, size, and unique cybersecurity requirements.

- **Continuous Improvement:**

Organizations implementing Level 2 profiles are expected to demonstrate a commitment to continuous improvement in their cybersecurity practices. Controls at this level may involve more robust monitoring, detection, and response capabilities.

2.3 Network Automation

Network Automation, often considered a fusion of programming and network infrastructure expertise, has evolved significantly, drawing inspiration from Software Defined Networks (SDN). This innovative approach relies on leveraging standard computer languages and scripting techniques to administer and maintain network components with minimal human intervention [24] [25]. This development enhances efficiency and drastically reduces configuration time, making it a compelling choice for implementing repetitive tasks across numerous devices [26] [27]. Managing the ever-increasing number of network devices is one notable application of network automation [28] [29] [30]. As the number of devices in a network continues to grow, the need for efficient and scalable management becomes paramount. In these scenarios, the adoption of network automation stands out as a favored choice owing to its ease of implementation and inherent flexibility [31]. This flexibility makes it possible for automation frameworks to strengthen network security by imposing standardized and consistent configurations across the entire infrastructure. This approach significantly reduces the likelihood of human errors, a crucial factor that could otherwise

expose vulnerabilities within the network architecture. By automating the configuration process, organizations can establish a uniform and secure baseline, ensuring that each network component adheres to predefined security protocols, rules, and guidelines. This not only enhances the overall performance of the network but also enables organizations to scale their infrastructure seamlessly.

2.3.1 Common Network Automation Tools and Frameworks

- **Ansible:**

Ansible is an open-source configuration management tool that uses YAML as its configuration format, making it easily readable by humans[32]. It uses the concept of playbooks which enables users to specify the actions, settings, and procedures necessary to automate different processes on target hosts or distant systems [33] [34].

- **Puppet:**

This is an open-source configuration management platform that permits the provisioning and automated management of IT systems and applications [35]. Unlike Ansible in which any computer can be a controller, Puppet uses the concept of “Puppet Master” which ensures a centralized point of control [36].

- **Chef:**

Chef follows an Infrastructure as Code (IaC) approach, which allows users to express their infrastructure configurations in code, making it more scalable, version-controlled, and repeatable [37]. Similar to Puppet, Chef uses the concept

of “Chet Server” or “Chef Master” as a centralized location to store cookbooks, recipes, attributes, roles, and other configuration data [38].

- **SaltStack:**

Often referred to simply as “Salt”, is written in Python and uses a client-server architecture to automate the deployment, configuration, and management of systems [39]. It is a configuration management tool designed to install and manage software on existing servers/end stations [40].

- **Netmiko:**

This is a Python library that makes utilizing SSH connections to automate network device interactions less complex. Unlike its counterpart, Paramiko [41], it provides a unified interface to manage and control various network devices irrespective of their vendor or platform [42]. This makes it a practical option for automation purposes [43].

The success and effectiveness of task automation are closely related to the chosen platform. The automation platform that has been selected for this research is Nornir.

2.4 The Nornir Framework

Nornir is an automation framework that is centered around the concept of “inventory”. The inventory contains one or more host information. This allows the utilization of multiple input sources into the inventory. This means in a multi-vendor network, a task can be run against all Cisco devices excluding Arista devices. Similarly, a task can

also be run against all routers but not switches. Hence, inventory management gives granular control of network automation by allowing the grouping of devices by users. This grants users the ability to easily run tasks against portions of the network. This is called inventory filtering. Nornir also manages job distribution to the devices and supplies a standard foundation to create "plugins"[44]. It provides a few inventory plugins to ease the automation process and configuration. The standard plugin is SimpleInventory, which works with YAML files. Tasks that are defined may include custom plugins, community-shared plugins, or Python code. The framework also has data management, parallelization, and error-tracking functionalities. Nornir uses data sets in its execution. It performs operations on this data and manages all threads. This often means that data of each node in a network environment is contained in the inventory. This makes Nornir flexible and suitable for several automation use cases.

CHAPTER 3

LITERATURE REVIEW

3.1 Overview

The authors of [45] used Ansible to configure EIGRP routing and some advanced configurations leveraging the concept of automation. Network automation aids network managers in automating processes using scripts, which reduces setup time and limits human error. Their data demonstrate that Ansible successfully and error-free deployed the configuration to the routers. Additionally, it was used to put EIGRP authentication into place, boosting network security. The scope of the research was on the implementation of a routing protocol without any device hardening configuration. Our research will focus on network security configurations which are crucial in ensuring network availability, performance, and reliability. In [46], a multilayer switch was used as a case study for network hardening, to identify vulnerabilities linked to getting unauthorized access to the device while it was in use. Vulnerabilities were discovered using the Nessus vulnerability scanner, the Metasploit project, and Nmap. The findings of the vulnerability scans reveal several flaws in the system that could

help an attacker infiltrate the operations of the device. Consequently, more tools and methods are required to increase the security of network devices. The research was centered around finding vulnerabilities and not providing practical approaches to harden the network devices. Hence, the research being proposed in this paper can serve as a consolidation by providing an automated methodology to seal the loopholes and vulnerabilities in network devices. In the work [47], A solution was put out to automate router security enhancements and make certain that they are updated and hardened with little to no configuration modifications and human involvement. Initial configuration, vulnerability patching, compliance audits, and rollback are the solution's four main tasks. All these were achieved successfully in a minimal amount of time as compared to manual implementation. The research only covered the scope of hardening Cisco routers. It didn't offer a solution that included other network devices from different vendors. The automation libraries used are Netmiko and Paramiko. These are Python libraries and do not support automation functionality like inventory management and multithreading for flexible configuration. The paper [48] outlines a task-based automation methodology that enables a campus network's hardening percentage to be increased to achieve high availability and reduce the likelihood of various attack vectors occurring. The Ansible automation tool was used to implement CIS Benchmarks. Their implementation indicated an increase in hardening which was achieved within a reduced amount of time as opposed to its implementation using traditional methods. Ansible uses the concept of playbooks written in YAML. YAML is highly human-readable but was not designed to handle sophisticated logic. Hence when a network scales, handling network automation becomes very complex. On the

other hand, a Python automation framework has full access to the entire Python ecosystem, making it more flexible and able to handle complex operations at a large scale. A Python automation framework also has better performance. The main issue is that Ansible serializes and deserializes JSON data internally within the core and between every task, which becomes a huge drawback when configuring a large number of devices. The authors of [49] proposed a strategy to limit the data an attacker might obtain once they infiltrate a network. Their approach is preventing lateral movement, diverting an attacker, or preventing access to a crucial device. Their linear time algorithms showed that even with incomplete information about the attacker and their actions, the defender can still reduce their danger. The research was conducted from a conceptual viewpoint without practical implementation. It is also a manual approach, rather than automated, hence not suitable for a large-scale network. In [50], the authors explored the use of Jinja template engine. It was aimed to give network administrators the information and abilities they need to successfully traverse the intricacies of network management. It started by going through the application stack and explaining the roles and relationships between each layer. The Ninja template engine is then described and used alongside Nornir to configure BGP, VTP, OSPF, and EIGRP. It was also used to show facts about the device. As compared to our research, Nornir is used to configure a multi-vendor network to harden the security posture of the devices based on CIS benchmarks.

CHAPTER 4

METHODOLOGY

4.1 Automation Framework Setup

There are 4 files associated with setting up the Nornir automation framework. They are:

- **Host file:**

Metadata about the network devices to be configured, like IP addresses, FQDNs, credentials, etc., were specified in the "hosts.yaml" file.

- **Group file:**

The groups listed here corresponded to the ones listed in our "hosts.yaml" file. They are needed for inventory filtering, which is the ability to run a task against a specific set of devices.

- **Defaults file:**

The default values, for example, username and password were specified in the "default.yaml" file

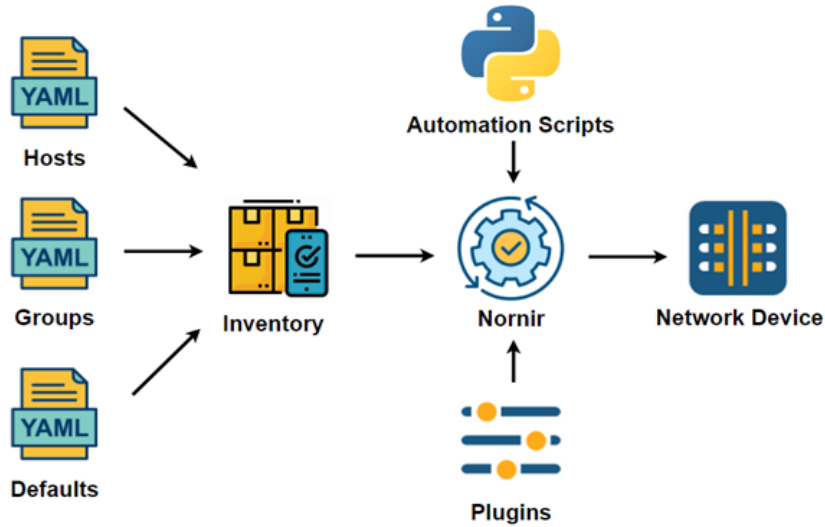


Figure 4.1: Nornir workflow

- **Config file:**

The content of the host file, group file, and default file were aggregated and mapped together by the “config.yaml” file. The thread plugin was imported and the “num_workers” varied depending on the number of devices. These configurations ensure that multithreading is used when setting up multiple network devices.

The final file needed is a python.py file. This file was used to import all the necessary dependencies needed to configure the network devices. The transport mechanism used to communicate with the devices was Netmiko. The complete workflow of Nornir is illustrated in Figure 4.1

Secure Socket Shell (SSH) provides a secure channel for accessing and managing network devices, servers, and other systems. SSH is widely used for remote administration, file transfers, and secure communication between devices [51]. It listens on port 22 of the Transmission Control Protocol (TCP) by default[52]. Nornir Netmiko

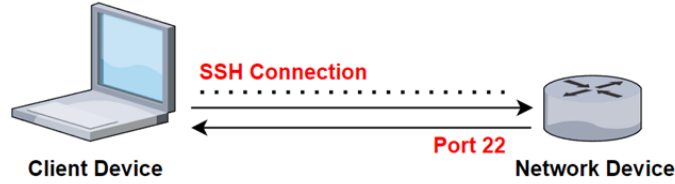


Figure 4.2: SSH Connection to Network Device

Plugin uses SSH which provides users with a secure way to access a network device over an insecure network. A typical SSH connection session is illustrated in Figure 4.2.

4.2 Simulation

In the field of networking, network simulation tools are essential because they offer a virtual environment for creating, simulating, and evaluating intricate network infrastructures [53]. Researchers and network professionals are among the many who benefit from these technologies, which provide an adaptable platform for learning and experimenting without being limited by physical gear [54]. The simulations in this research were done on Graphical Network Simulation-3 (GNS 3). It supports the use of both actual and virtual devices to model complicated networks. The images of the devices from various vendors when imported into GNS 3 simulate a real-world scenario [55]. This enables proper network analysis in terms of latency, throughput, and other network parameters [56]. GNS 3 also has Wireshark integrated into it. Wireshark is the most prominent and extensively used network protocol analyzer in the world [57]. It is the most frequently used standard across many commercial and non-profit firms, governmental organizations, and educational institutions because it enables you to monitor all network and traffic activities [58] [59]. To achieve the objectives of this

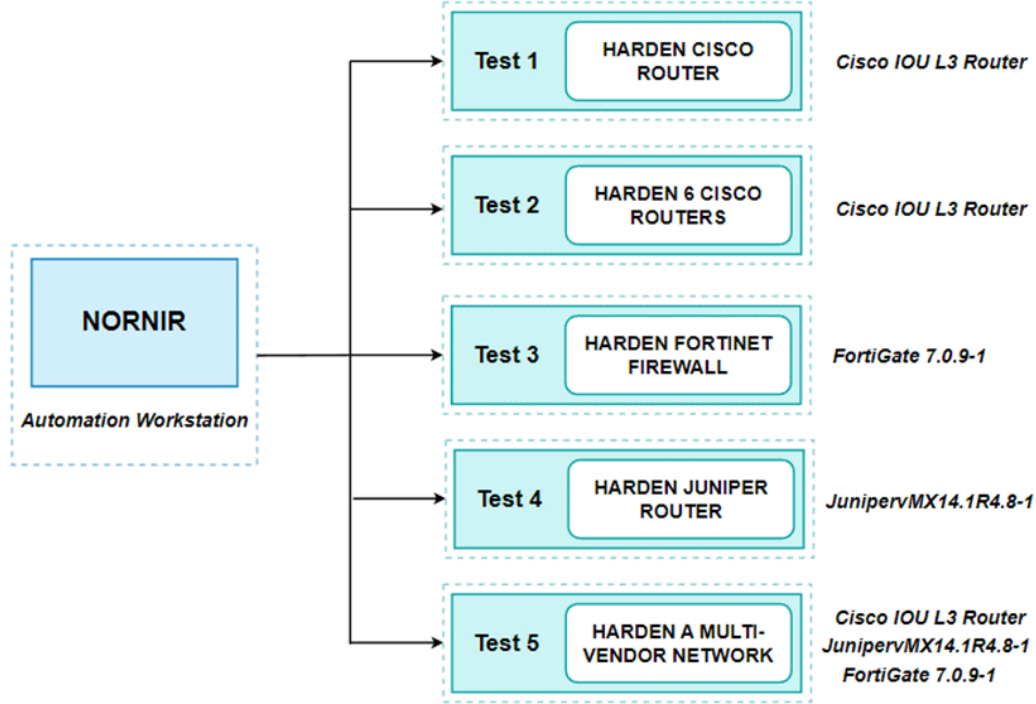


Figure 4.3: Test Scenarios with Nornir Automation Framework

research, 5 different simulations were carried out. The different test cases are illustrated in Figure 4.3. The pseudocode used for all the implementations is presented in Figure 3 below. The elapsed time of level 1 and level 2 for test cases 1, 3, and 4 were recorded. This will better inform network professionals on the time implication of implementing the security profiles based on the needs of their organizations and the sensitivity of their operations. From a general viewpoint, security profile level 1 had a higher elapsed time in all test cases. This is because it serves as the foundation of network device security and contains all the basic security implementations.

For the implementation and evaluation of the simulation, a code was written on Visual Studio Code IDE. Table 4.2 displays the hardware specifications used to perform the simulations and Table 4.2 displays the software specifications used to perform the simulations and capture performance metrics for analysis in the next section.

Table 4.1: Hardware Specifications

Name	Description
CPU	Intel® Xeon® Silver 4216 CPU @ 2.10GHz
OS	64-bit Ubuntu 20.04.4 LTS
Memory	128 GB

Table 4.2: Software Specifications

Name	Version	Purpose
GNS 3	2.2.36	Emulate a real network
GNS 3 Virtual Machine	2.2.36	Build QEMU and IOU images
Visual Studio Code	1.75.1	Build automation scripts
Cisco IOU L3	i86bi	Network device to be configured
FortiGate	7.0.9-1	Network device to be configured
Juniper vMX	14.1R4.8-1	Network device to be configured
VMware Workstation Pro	16.2.4	Virtual machine to run GNS 3
Nornir	3.1.1	Automation Framework

Cisco IOU is an application that runs like any other Cisco IOS device because it is created as a native Solaris image. IOU supports all cross-platform features and protocols. This makes it a suitable choice for simulating networks in a virtual environment. The Cisco IOU L3, JunipervMX, and FortiGate images were imported into GNS 3 and installed on the GNS 3 virtual machine which has been integrated with the VMware workstation Pro software.

CHAPTER 5

RESULTS AND DISCUSSION

5.1 TEST 1: Hardening of a Cisco router

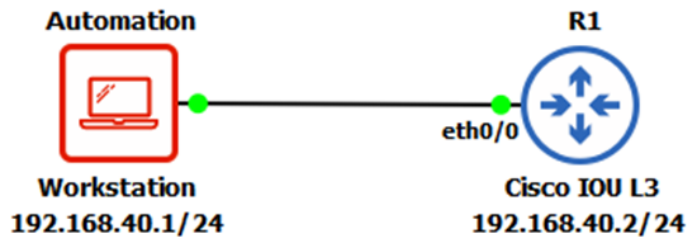


Figure 5.1: Network Setup For Basic Router Configuration On GNS 3

The hardening rules for a Cisco router have the following categories:

- **Management Plane**

CIS guidelines for the management plane aim to prevent unauthorized access and ensure that only authorized administrators can access and configure the device. This involves practices such as:

- Enabling strong authentication mechanisms
- Restricting access to management interfaces to specific IP addresses or

networks.

- Disabling unnecessary services and features not required for management tasks.
- Regularly updating and patching the device’s firmware and software.

- **Control Plane**

The control plane is responsible for routing and maintaining the device’s routing table. Hardening the control plane ensures the integrity and stability of the routing functions. The guidelines for the control plane include:

- Implementing robust routing protocols and securing them with authentication mechanisms.
- Applying route filtering and access control lists (ACLs) to control the flow of routing updates and prevent route manipulation by unauthorized entities.
- Configuring route aggregation and summarization to reduce the size of routing tables and limit unnecessary exposure of internal network details.

- **Data Plane**

Securing the data plane involves protecting the device and network from unauthorized data traffic and potential threats. CIS guidelines for the data plane include:

- Applying access control lists (ACLs) to filter traffic based on allowed sources, destinations, and protocols.

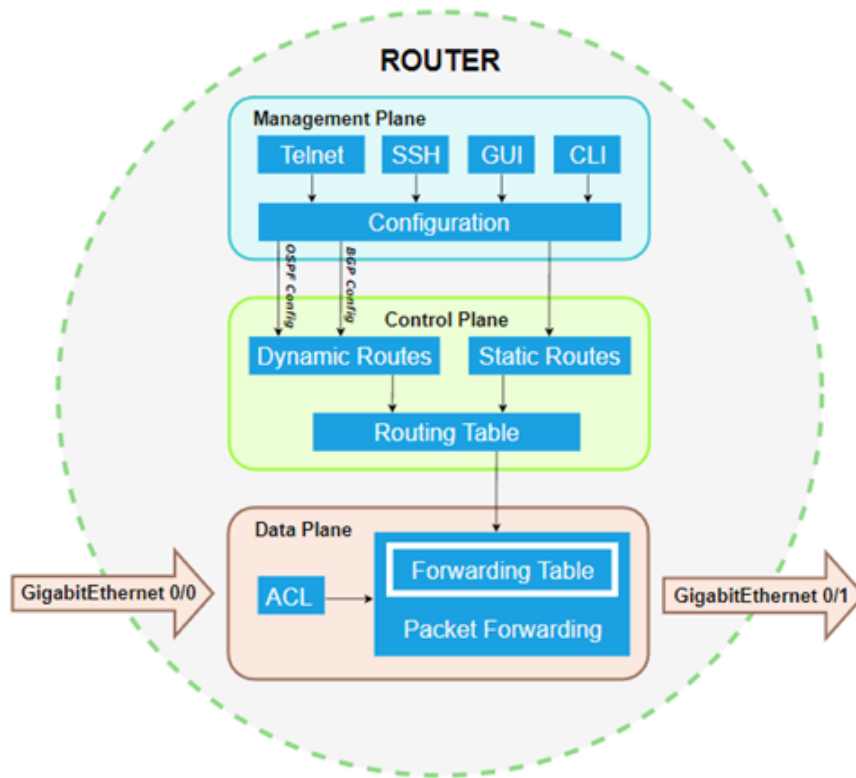


Figure 5.2: Management, Control, and Data Plane

- Enabling features like Unicast Reverse Path Forwarding (uRPF) to prevent IP address spoofing and protect against certain types of denial-of-service (DoS) attacks.
- Implementing encryption protocols like IPsec to secure data transmission across public or untrusted networks.

To harden the Cisco router, the router is first connected to the automation workstation which has PyCharm and Nornir installed. The network setup is shown in Figure 6 above. The automation workstation was assigned an IP address of 192.168.40.1/24 and the router interface eth0/0 was assigned an IP address of 192.168.40.2/24. These are both private IP addresses needed to establish connectivity between both devices.

```
osboxes@osboxes:~$ ping 192.168.40.2 -c 5
PING 192.168.40.2 (192.168.40.2) 56(84) bytes of data.
64 bytes from 192.168.40.2: icmp_seq=1 ttl=128 time=4.18 ms
64 bytes from 192.168.40.2: icmp_seq=2 ttl=128 time=3.66 ms
64 bytes from 192.168.40.2: icmp_seq=3 ttl=128 time=3.33 ms
64 bytes from 192.168.40.2: icmp_seq=4 ttl=128 time=3.71 ms
64 bytes from 192.168.40.2: icmp_seq=5 ttl=128 time=6.10 ms
```

Figure 5.3: Successful Ping Test Between Workstation and Router On GNS 3

A ping test was done to confirm layer 3 connectivity between both devices. The output is shown in Figure 5.3

Next, the host, config, group, and default YAML files were configured accordingly. Finally, the automation scripts were written to import the transport mechanism and integrate all the YAML files. Following the CIS benchmarks, the process of hardening a Cisco router is structured into three distinct planes: the Management Plane, the Control Plane, and the Data Plane. Specific configurations included AAA rules, access rules, banner rules, password rules, SNMP rules, global service, logging rules, NTP rules, loopback rules, routing rules, border routing filters, and OSPF authentication. The execution time was obtained by running the code 5 times for each plane. The average was then calculated and recorded. The duration to harden the various device planes is illustrated in Figure 5.4

As anticipated, the hardening process for the Management Plane consumed a noteworthy 10.1499 seconds of execution time. This elongated duration can be attributed to the comprehensive nature of the Management Plane's responsibilities, encompassing critical administrative functions such as device configuration, monitoring, and remote access. It also involves securing authentication methods, implementing access controls, and configuring strong passwords [60]. These security measures collectively

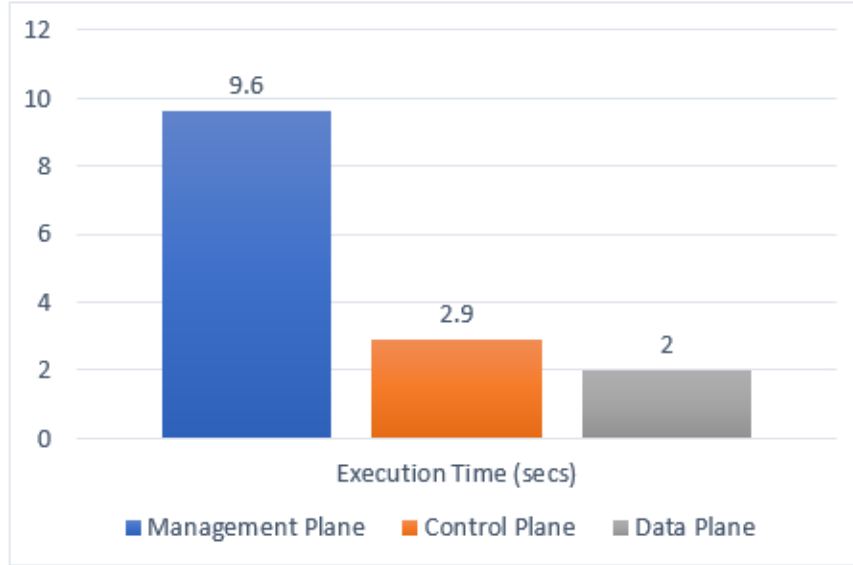


Figure 5.4: Elapsed Time for Hardening the Three Router Planes

contribute to the safeguarding of the router, guaranteeing that access and management privileges are exclusively granted to authorized personnel. The execution time of Level 1 and Level 2 profiles for Cisco router hardening were also recorded. The results are illustrated in Figure 5.5

The total execution time required to implement the entirety of the CIS recommended configuration, encompassing the Management, Control, and Data Plane configurations on the Cisco router, amounted to 11.4 seconds. Using a confidence level of 95%, the upper and lower confidence interval can be expressed as $CI = [10.7, 12]$

5.2 TEST 2: Hardening of 6 Cisco routers

Nornir distinguishes itself from other automation frameworks through its multithreading capabilities which enhance efficiency and performance. This feature is facilitated through the utilization of the config.yaml file. This file allows for the specification of

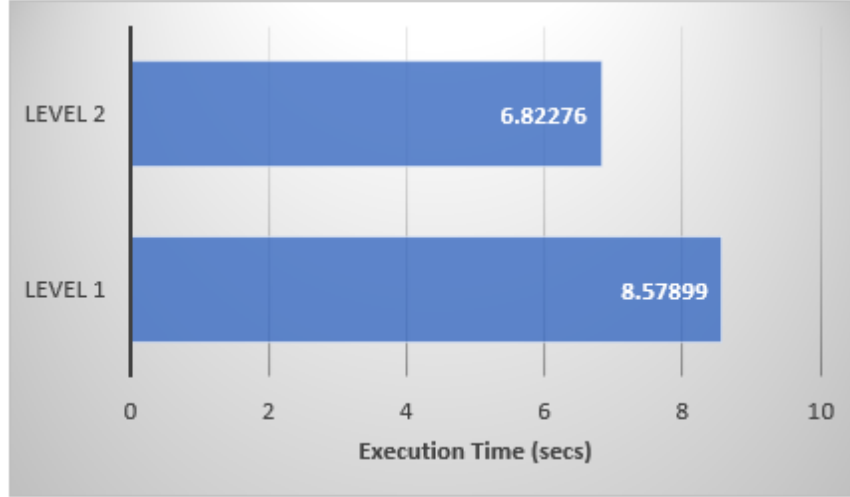


Figure 5.5: Elapsed Time of Level 1 and Level 2 Profile Hardening

the number of workers, a parameter that dictates the extent to which multithreading is employed for the orchestration and configuration of network devices. To test this feature, the number of workers was set to 1 and the number of devices to be configured was increased from 1 to 3. The results obtained are shown in Figure 5.6.

The test results demonstrate that configuring the number of workers to a value of 1 resulted in the sequential execution of hardening tasks. This observation shows the singular processing nature of task execution, where each task is meticulously carried out one after the other. Part of the objective of this research is to compare the performance of our proposed framework with previous studies, hence a topology of 6 devices was set up similar to the one used in previous studies. The topology is shown in Figure 5.7

In determining the optimal value for the number of workers in the configuration process of 6 Cisco devices, a systematic approach was employed. The number of workers was incrementally increased, and the corresponding execution times

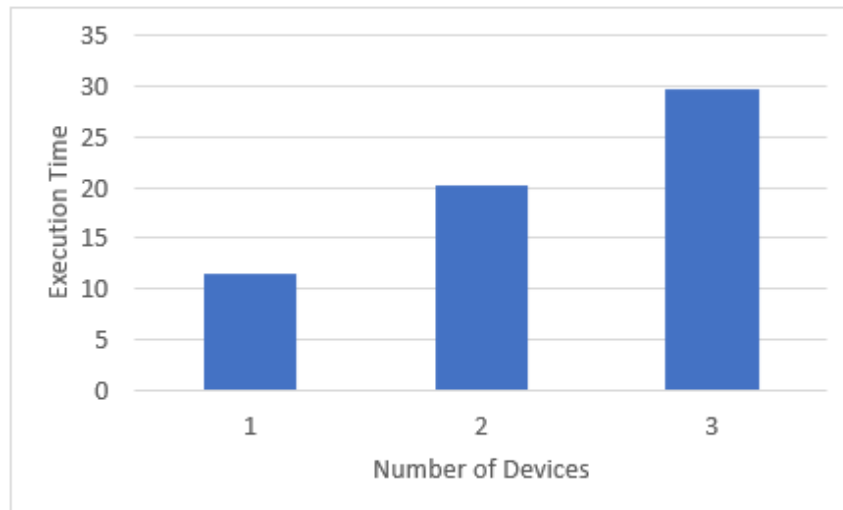


Figure 5.6: Hardening of Devices with number of workers set to 1

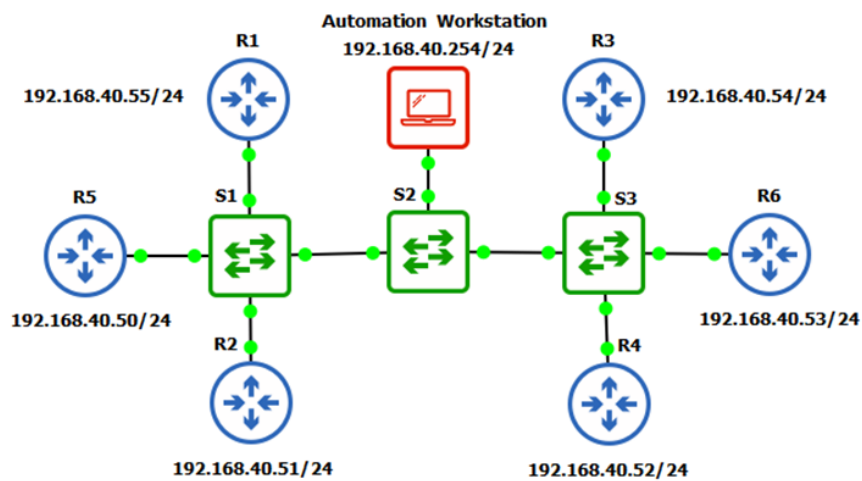


Figure 5.7: Topology Setup For Hardening Six Cisco Routers

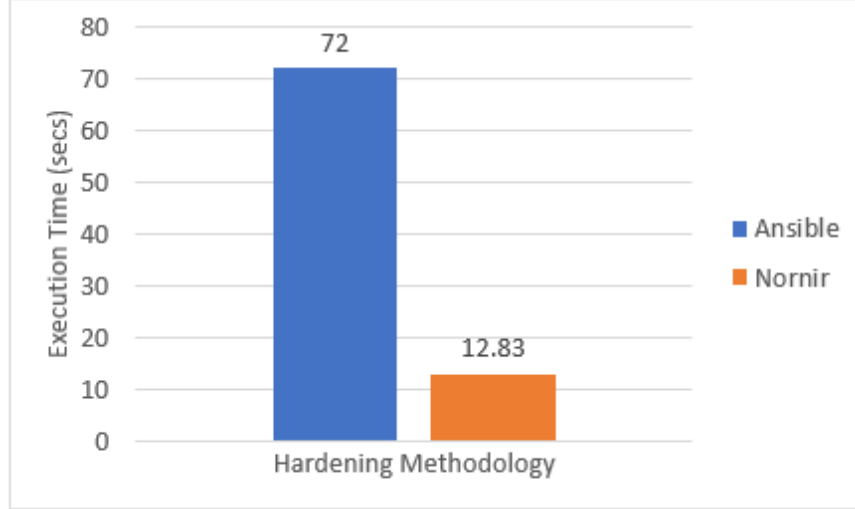


Figure 5.8: Execution Time Comparims

were recorded. The lowest execution time of 12.83 seconds was obtained when $\text{num_worker} \geq 6$. Hence, subsequent hardening implementations were done with $\text{num_worker} \geq \text{number of devices}$. To compare the elapsed time of our device hardening process with the findings reported in a previous research paper, Ansible was used to harden 6 Cisco devices. The execution time obtained was 72 seconds. This presents an 82.18% decrease in execution time. Our proposed automation framework is approximately 5.61 times faster mostly due to its multithreading ability. Figure 5.8 shows the elapsed time using both methodologies.

5.3 TEST 3: Hardening a Fortigate firewall

In this experiment, a hardening process was conducted on a FortiGate firewall while adhering to the Center for Internet Security (CIS) benchmarks. A FortiGate 7.0.9-1 image was used for the simulation. It was connected to the automation workstation via port 1. The network topology is shown in Figure 5.9

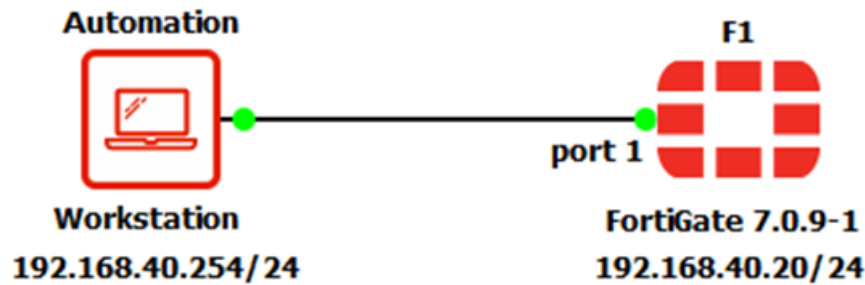


Figure 5.9: Network Topology For FortiGate Hardening

The hardening measures covered the following key categories:

- **Network Settings**

- Restricting access to management interfaces and services.
- Configuring secure protocols for remote management access.
- Enabling logging and monitoring for network events.

- **System Settings**

- Applying the principle of least privilege for administrative access.
- Disabling unnecessary services and protocols.
- Enforcing strong password policies and user authentication mechanisms.

- **Policy and Object**

- Reviewing and optimizing firewall policies.
- Limiting traffic to only essential services and applications.
- Defining and organizing objects for efficient policy management.

- **Security Profiles**

- Creating custom security profiles for traffic inspection.
- Configuring antivirus, intrusion prevention, and web filtering profiles.
- Enabling application control to manage application usage.
- **Security Fabric**
 - Integrating the FortiGate firewall with other security components.
 - Setting up automated threat intelligence sharing for enhanced threat detection.
- **VPN**
 - Establishing secure Virtual Private Network (VPN) connections.
 - Enabling encryption protocols for data privacy and integrity.
- **Users and Authentication**
 - Implementing secure authentication methods, such as two-factor authentication (2FA).
 - Integrating with existing authentication systems like LDAP or Active Directory.
- **Logs and Reports**
 - Configuring logging to capture security events and anomalies.
 - Generating regular reports for monitoring and analysis.

```
KFUPM-FIREWALL # show system interface port1
config system interface
    edit "port1"
        set vdom "root"
        set ip 192.168.40.20 255.255.255.0
        set allowaccess ssh
        set type physical
        set snmp-index 1
    next
end
```

Figure 5.10: Port1 Interface with IP Address and SSH Enabled

The first step was to configure the port1 interface and enable SSH for remote access from the automation workstation. This can be verified by the “show system interface port1” command output as shown in Figure 5.10

Following the completion of the hardening process, the elapsed time of each category was computed and is presented in Table 5.1 and Figure 5.11. The system Settings involve configurations related to the fundamental behavior and operation of the firewall, and hence, has the longest execution time. It also includes settings related to system resources, administrative access, and core functionalities. The complexity of these configurations contributes to the longer time needed for hardening. The elapsed time for Level 1 and Level 2 profiles is also provided in Figure 5.12. The process of implementing the full CIS recommended configuration for FortiGate, spanning all the categories took 13.93 seconds to complete.

Table 5.1: Fortigate Hardening Categories and Elapsed Time

SN	Name of Fortigate Area	Execution Time (secs)
1	Network Settings	12.477
2	System Settings	13.119
3	Policy and Objects	12.442
4	Security Profiles	12.498
5	Security Fabric	12.672
6	VPN	12.468
7	Users and Authentication	12.458
8	Logs and Reports	12.429

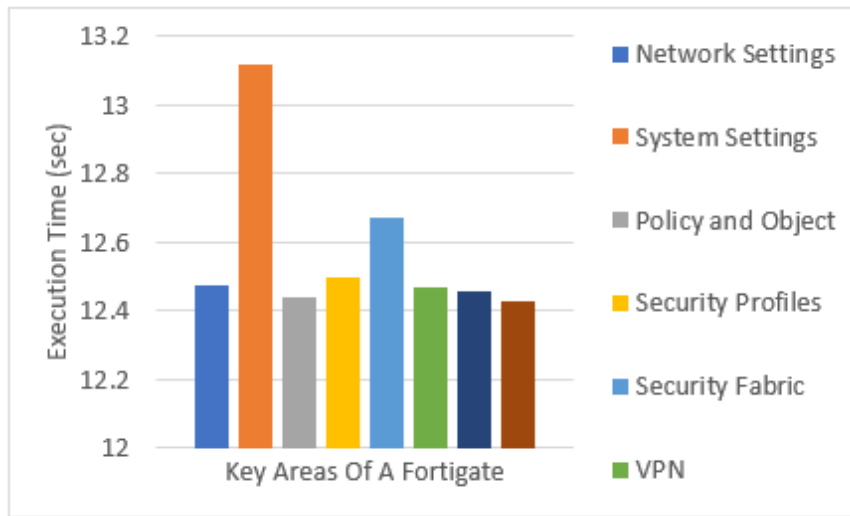


Figure 5.11: Elapsed Time Of Fortigate Hardening Categories

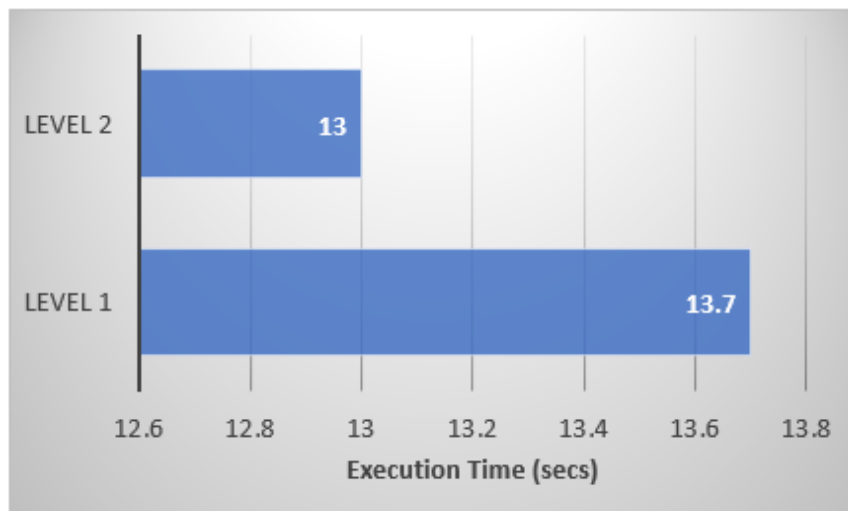


Figure 5.12: Elapsed Time of Level 1 and Level 2 Profile Hardening

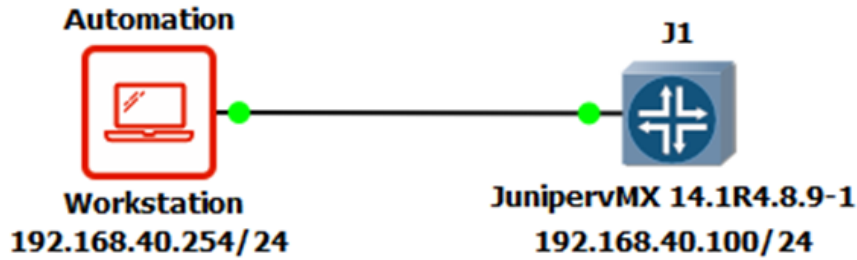


Figure 5.13: Network Topology For FortiGate Hardening

```
admin@KFUPM-JUNOS> show configuration interfaces em2
unit 0 {
    family inet {
        address 192.168.40.100/24;
    }
}
```

Figure 5.14: em2 Interface Configuration

5.4 TEST 2: Hardening a Juniper router

An investigative procedure was also undertaken to assess the security enhancement measures applied to a Juniper router utilizing the Juniper vMX 14.1R4.8.9-1 image. These security enhancements align with the stringent guidelines prescribed by the Center for Internet Security (CIS). To enable layer 3 connectivity with the workstation, the juniper router was assigned an IP Address of 192.168.40.100/24 as depicted in the illustrative network topology and the “show configuration interface em2” output presented in Figure 5.13 and 5.14.

The security enhancements encompassed the essential categories listed below:

- **Firewall**
 - JUNOS Devices offer Stateless, per-packet Firewall functionality for IPv4, IPv6, and MPLS traffic. The filters can also be employed to safeguard the

Junos device itself.

- The Filters are set up within the `[edit firewall]` hierarchy.

- **Interface**

- This section offers guidance for secure configuration about Interface-specific parameters and options.
- These settings are found within the `[edit interfaces <interface name>]` hierarchy.

- **Protocols**

- This hierarchy is instrumental in device hardening, accommodating a broad spectrum of protocols, which encompasses crucial elements like routing protocols, MPLS, and PIM.
- Its parameters are contained under the `[edit protocols]` hierarchy.

- **SNMP**

- The Simple Network Management Protocol (SNMP) offers a standardized interface for the management and monitoring of network devices.
- This section offers essential guidance for configuring SNMP parameters securely, which are contained under the `[edit snmp]` hierarchy.

- **System**

- In the context of device hardening, this section delves into the configurations that pertain directly to the JUNOS System, encompassing aspects

Table 5.2: Juniper Hardening Categories and Elapsed Time

SN	Name of Juniper Area	Execution Time (secs)
1	Firewall	1.421
2	Interface	1.360
3	Protocols	2.534
4	SNMP	1.079
5	System	3.928

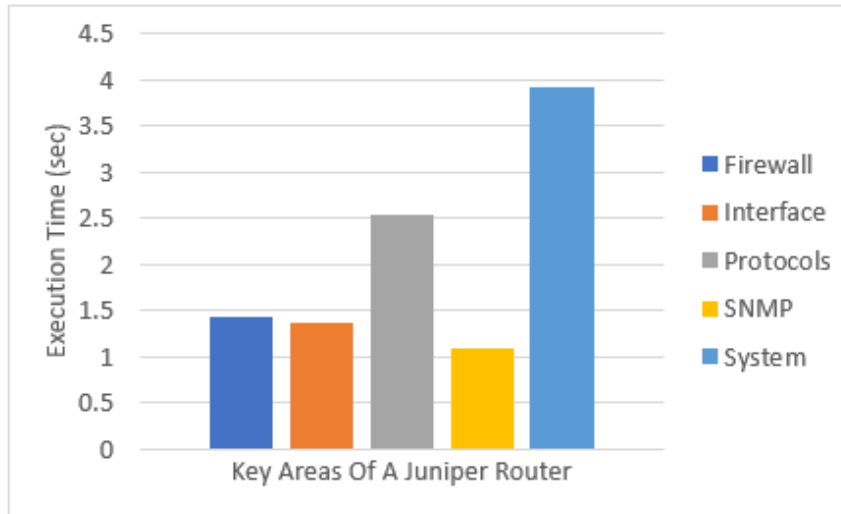


Figure 5.15: Elapsed Time Of Juniper Hardening Categories

like DNS Servers, Hostname, Configuration Archiving, and User management.

- All of these configurations are implemented within the `[edit system]` hierarchy.

Upon the successful implementation of the hardening process, analysis was conducted to measure the time required for each distinct category. As experienced in the hardening process of the FortiGate firewall, the system category also takes the longest time based on similar reasons. The findings of this comprehensive assessment are provided in both Table 5.2 and Figure 5.15, presented subsequently.

Furthermore, to offer a more granular perspective on the hardening process, the

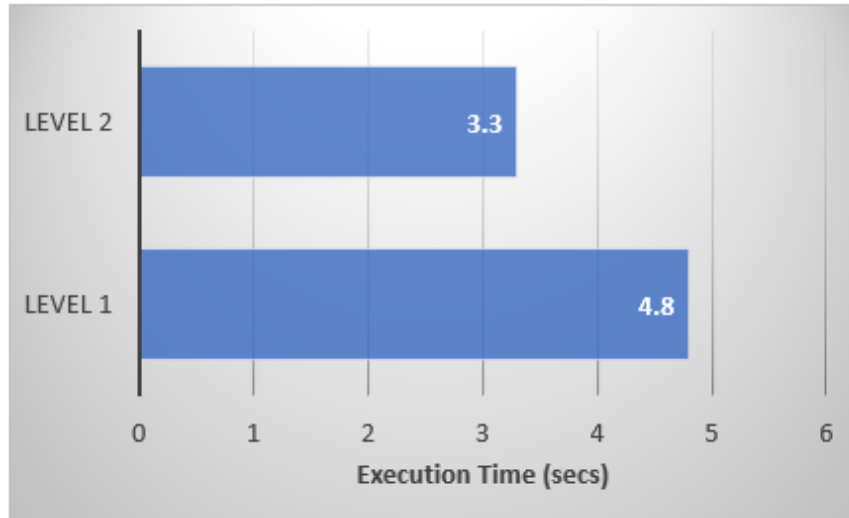


Figure 5.16: Elapsed Time of Level 1 and Level 2 Profile Hardening

elapsed time specifically associated with the Level 1 and Level 2 security profiles is detailed in Figure 5.16

5.5 TEST 5: Hardening a multivendor network

The network topology employed for assessing the device hardening in a multi-vendor environment is depicted in Figure 5.17. The devices selected for the hardening process include 8 Cisco routers, 7 Juniper routers, and 3 Fortinet firewalls, making a total of 18 devices. The simulation was executed using an incremental approach, introducing devices one after the other to evaluate how the elapsed time increases as the number of devices increases. Figure 5.18 shows that the highest time interval during the entire process was VT1 and VT2 which were 7.7 seconds and 12.3 respectively. VT1 denotes the transition from the last Cisco device to the first Juniper device while VT2 denotes the transition time between the last Juniper device to the first Fortigate device. The elapsed time for the entire process was 44 seconds. Using a confidence level of 95%,

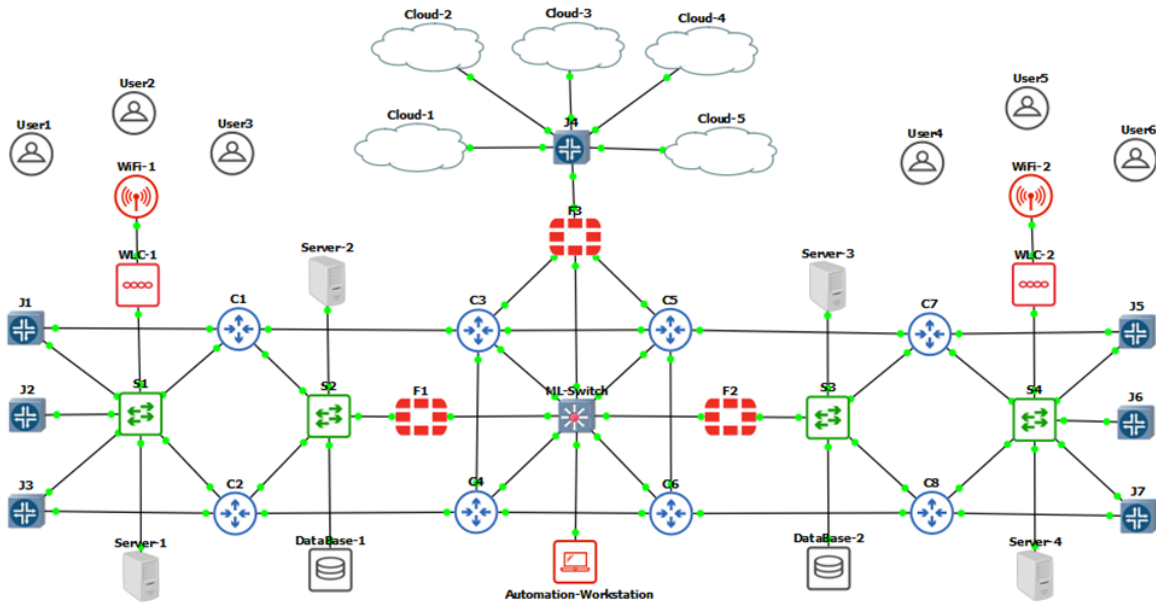


Figure 5.17: Multivendor Network Topology

the upper and lower confidence interval can be expressed as $CI = [43.8, 44.1]$

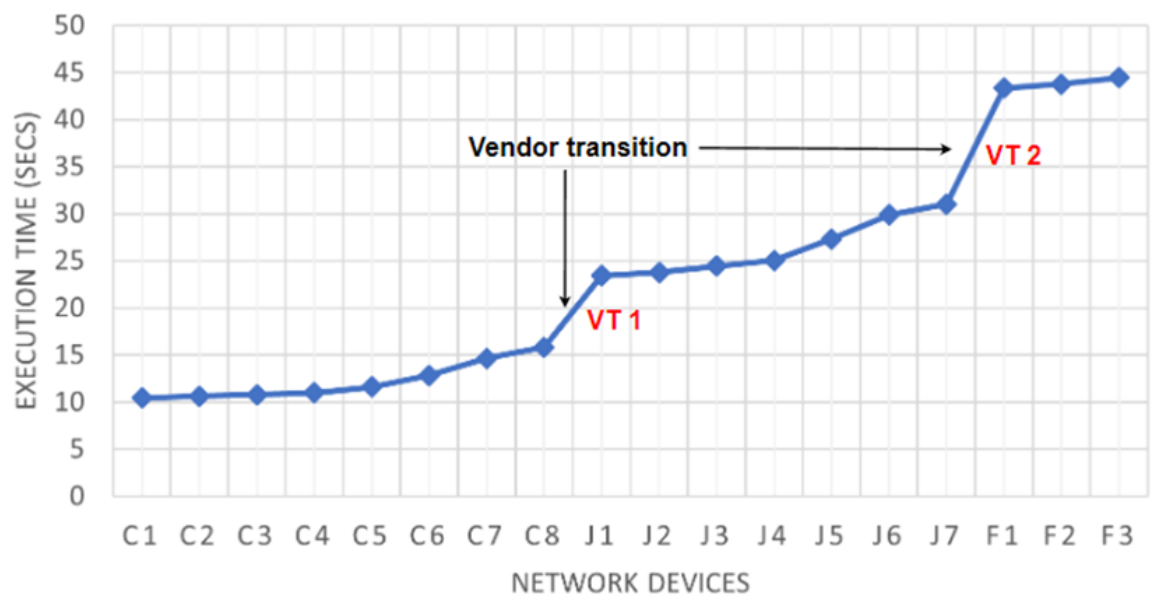


Figure 5.18: Elapsed Time to Configure 18 Devices

CHAPTER 6

CONCLUSION AND FUTURE WORK

The central objective of this research was to automate the security hardening procedures for network devices, specifically routers and firewalls. Traditionally, these hardening processes have been executed manually, a method known for its inefficiency and potential inadequacy, resulting in vulnerabilities within an organization's network security. The solution encompasses the hardening of network devices from diverse vendors, including industry leaders such as Cisco, Juniper, and Fortigate using Cisco IOU, Juniper vMX, and Fortigate images in a virtual environment as proof of concepts. The experiments carried out in this research were categorized into various test cases. In the first test case, an automation workstation was set up in the virtual environment and interfaced with a Cisco router. The hardening procedure was implemented to cover the management, control, and data planes of the Cisco router. The second test case compared the results obtained from hardening 6 Cisco

routers with previous work. Significantly less time was recorded which demonstrates the multithreading abilities of the Nornir automation framework. The third test case involved the hardening of a FortiGate firewall using the initial setup workstation. System settings and security fabric had the longest hardening time due to the numerous configurations that need to be implemented on a firewall device. This ensures that the entire internal fabric of a network architecture is secure, with the firewall device at the edge of the network. The fourth test case involved hardening a juniper router. For a juniper router to be fully hardened, a network administrator needs to pay more attention to the protocols and system settings categories. This is due to the loopholes that an attacker can use to infiltrate the router if it has loose protocol configurations. For test cases 1,3 and 4, the system/management category took the longest time to harden. This is attributed to the fact that the "System" category involves configurations related to the overall system behavior, including core settings, system resources, and global parameters. Configuring these settings is more complex and involves a larger number of parameters compared to other aspects. The last test case implemented the hardening procedure on a multi-vendor network involving Cisco routers, Juniper routers, and FortiGate firewalls. The approach used is user-friendly and can be used by individuals with limited prior knowledge of routers, firewalls, and in-depth networking concepts. The conclusive findings affirm that employing an automation framework featuring multi-vendor and multi-threading capabilities can yield substantial enhancements in the performance of the automation process, thereby fortifying the security posture of network devices. The impact of the research findings extends beyond mere efficiency gains. These findings are of paramount importance

to network security professionals, as our approach not only accelerates the hardening process but also minimizes the likelihood of configuration inconsistencies. As we look ahead, several avenues for future work that can build upon the foundations laid in this study are as follows:

- **Comparison with Existing Tools in the Market:** By conducting a comparative analysis, the unique strengths of Nornir and possible weaknesses can be better understood which will pave the way for improvements.
- **Automate Real Devices:** The Nornir framework can be used to automate the hardening configurations on real devices. Automating real devices will bring about a comprehensive and reliable understanding of the framework's impact on diverse network environments, reinforcing its applicability in real-world scenarios.
- **Machine Learning for Threat Detection:** Explore the integration of machine learning algorithms to enhance threat detection capabilities. By leveraging historical data and patterns, the automation framework could intelligently adapt security settings to proactively mitigate emerging threats.
- **Compliance Monitoring and Reporting:** Incorporate features for continuous compliance monitoring. Develop reporting mechanisms that provide insights into the adherence of network devices to security policies and industry standards over time, facilitating regulatory compliance and auditing processes.
- **Threat Intelligence Integration:** Integrate external threat intelligence feeds

into the automation framework to enhance its ability to adapt security configurations based on the latest threat intelligence. This could involve collaboration with threat intelligence providers and the development of mechanisms to dynamically update security policies.

- **User Behavior Analytics:** Incorporate user behavior analytics into the framework to identify abnormal patterns or deviations in user activity. This could contribute to the early detection of insider threats and the automatic adjustment of security policies based on anomalous behavior.

APPENDIX A

CIS PROFILES

A.1 LEVEL 1

A.1.1 Management Plane

- Local Authentication, Authorization, and Accounting (AAA) Rules
 - Require AAA Service
 - Require AAA Authentication for Local Console and VTY Lines
- Access Rules
 - Require Local User and Encrypted Password
 - Require SSH for Remote Device Access
 - Require VTY Transport SSH
 - Forbid Auxiliary Port

A.1.2 Control Plane

- Clock Rules
 - Require Clock Timezone - UTC
- Global Service Rules
 - Forbid CDP Run Globally
 - Forbid Finger Service
 - Forbid IP BOOTP server
 - Forbid Identification Service
 - Require TCP keepalives-in Services

A.1.3 Data Plane

- Routing Rules
 - Forbid Directed Broadcast
 - Forbid IP source-route

A.2 LEVEL 2

A.2.1 Management Plane

- Authentication, Authorization, and Accounting Rules
 - Require AAA Authentication Enable

- Require AAA Authentication Login
- Require AAA Accounting Commands
- Require AAA Accounting Connection
- Require AAA Accounting Exec
- Require AAA Accounting System

A.2.2 Control Plane

- Loopback Rules
 - Require Binding NTP Service to Loopback Interface
 - Require Binding TFTP Service to Loopback Interface
 - Require Loopback Interface

A.2.3 Data Plane

- Border Router Filtering
 - Forbid Private Source Addresses from External Networks
 - Forbid External Source Addresses on Outbound Traffic
- Routing Rules
 - Require Unicast Reverse-Path Forwarding
 - Forbid IP Proxy ARP
 - Forbid Tunnel Interfaces

REFERENCES

- [1] D. B. Rawat and S. R. Reddy, “Software Defined Networking Architecture, Security and Energy Efficiency: A Survey,” *IEEE Commun. Surv. Tutor.*, vol. 19, no. 1, pp. 325–346, 2017, doi: 10.1109/COMST.2016.2618874.
- [2] A. Aguado, V. Lopez, J. Martinez-Mateo, M. Peev, D. Lopez, and V. Martin, “Virtual network function deployment and service automation to provide end-to-end quantum encryption,” *J. Opt. Commun. Netw.*, vol. 10, no. 4, pp. 421–430, Apr. 2018, doi: 10.1364/JOCN.10.000421.
- [3] B. A. A. Nunes, M. Mendonca, X.-N. Nguyen, K. Obraczka, and T. Turletti, “A Survey of Software-Defined Networking: Past, Present, and Future of Programmable Networks,” *IEEE Commun. Surv. Tutor.*, vol. 16, no. 3, pp. 1617–1634, 2014, doi: 10.1109/SURV.2014.012214.00180.
- [4] K. Jambunatha, “Design and implement Automated Procedure to upgrade remote network devices using Python,” in 2015 IEEE International Advance Computing Conference (IACC), Jun. 2015, pp. 217–221. doi: 10.1109/IADCC.2015.7154701.

- [5] P. Kumari and A. K. Jain, “A comprehensive study of DDoS attacks over IoT network and their countermeasures,” *Comput. Secur.*, vol. 127, p. 103096, Apr. 2023, doi: 10.1016/j.cose.2023.103096.
- [6] A. N. Alhaj and N. Dutta, “Analysis of Security Attacks in SDN Network: A Comprehensive Survey,” in *Contemporary Issues in Communication, Cloud and Big Data Analytics*, H. K. D. Sarma, V. E. Balas, B. Bhuyan, and N. Dutta, Eds., in Lecture Notes in Networks and Systems. Singapore: Springer, 2022, pp. 27–37. doi: 10.1007/978-981-16-4244-9_3.
- [7] A.-D. Tudosi, D. G. Balan, and A. D. Potorac, “Secure network architecture based on distributed firewalls,” in 2022 International Conference on Development and Application Systems (DAS), May 2022, pp. 85–90. doi: 10.1109/DAS54948.2022.9786092.
- [8] Q. Burke et al., “Enforcing Multilevel Security Policies in Unstable Networks,” *IEEE Trans. Netw. Serv. Manag.*, vol. 19, no. 3, pp. 2349–2365, Sep. 2022, doi: 10.1109/TNSM.2022.3176820.
- [9] A. Chowdhary, A. Sabur, N. Vadhane, and D. Huang, “Intent-Driven Security Policy Management for Software-Defined Systems,” *IEEE Trans. Netw. Serv. Manag.*, vol. 19, no. 4, pp. 5208–5223, Dec. 2022, doi: 10.1109/TNSM.2022.3183591.
- [10] D. Reinelt and M. Wolfram, “Security in virtual automation networks,” in 2008 IEEE International Conference on Emerging Technologies and Factory

Automation, Sep. 2008, pp. 480–483. doi: 10.1109/ETF.2008.4638441.

[11] S. Arzo, C. Serugunda, F. Granelli, R. Bassoli, M. Devetsikiotis, and F. Fitzek, “A Theoretical Discussion and Survey of Network Automation for IoT: Challenges and Opportunity,” *IEEE Internet Things J.*, vol. 8, pp. 12021–12045, Aug. 2021, doi: 10.1109/JIOT.2021.3075901.

[12] M. F. Mohd Fuzi, K. Abdullah, I. H. Abd Halim, and R. Ruslan, “Network Automation using Ansible for EIGRP Network,” *J. Comput. Res. Innov.*, vol. 6, no. 4, pp. 59–69, Sep. 2021, doi: 10.24191/jcrinn.v6i4.237.

[13] “A Survey of Software-Defined Networking: Past, Present, and Future of Programmable Networks | IEEE Journals & Magazine | IEEE Xplore.” Accessed: Nov. 25, 2022. [Online]. Available: <https://ieeexplore.ieee.org/document/6739370>

[14] A. J. Bustamante, N. Ghimire, P. R. Sanghavi, S. Sharma, and D. Maheshwari, “An empirical evaluation of Machine Learning for hardening security devices in Data Networks,” in 2021 IEEE CHILEAN Conference on Electrical, Electronics Engineering, Information and Communication Technologies (CHILECON), Dec. 2021, pp. 1–5. doi: 10.1109/CHILECON54041.2021.9703085.

[15] K. Zenitani, “A multi-objective cost–benefit optimization algorithm for network hardening,” *Int. J. Inf. Secur.*, vol. 21, no. 4, pp. 813–832, Aug. 2022, doi: 10.1007/s10207-022-00586-7.

- [16] V. Sobeslav, L. Balik, O. Hornig, J. Horalek, and O. Krejcar, “Endpoint firewall for local security hardening in academic research environment,” *J. Intell. Fuzzy Syst.*, vol. 32, no. 2, pp. 1475–1484, Jan. 2017, doi: 10.3233/JIFS-169143.
- [17] U. Tariq, I. Ahmed, A. K. Bashir, and K. Shaukat, “A Critical Cybersecurity Analysis and Future Research Directions for the Internet of Things: A Comprehensive Review,” *Sensors*, vol. 23, no. 8, Art. no. 8, Jan. 2023, doi: 10.3390/s23084117.
- [18] Ö. Aslan, S. Aktug, M. Ozkan Okay, A. Yilmaz, and E. Akin, “A Comprehensive Review of Cyber Security Vulnerabilities, Threats, Attacks, and Solutions,” *Electronics*, vol. 12, pp. 1–42, Mar. 2023, doi: 10.3390/electronics12061333.
- [19] A. Gupta and A. Anand, “Ethical Hacking and Hacking Attacks,” *Int. J. Eng. Comput. Sci.*, vol. 610, pp. 2319–7242, May 2017, doi: 10.18535/ijecs/v6i4.42.
- [20] S. Y. Enoch, J. Mendonça, J. B. Hong, M. Ge, and D. S. Kim, “An integrated security hardening optimization for dynamic networks using security and availability modeling with multi-objective algorithm,” *Comput. Netw.*, vol. 208, p. 108864, May 2022, doi: 10.1016/j.comnet.2022.108864.
- [21] R. Sasidharan, “A Case Study to Implement Windows System Hardening using CIS Controls,” *Int. J. Comput. Trends Technol.*, vol. 70, pp. 1–7, Jul. 2022, doi: 10.14445/22312803/IJCTT-V70I7P101.

- [22] H. Winarno, F. Yasin, M. A. Prasetyo, F. Rohman, M. R. Shihab, and B. Ranti, “IT Infrastructure Security Risk Assessment using the Center for Internet Security Critical Security Control Framework: A Case Study at Insurance Company,” in 2020 3rd International Conference on Computer and Informatics Engineering (IC2IE), Sep. 2020, pp. 404–409. doi: 10.1109/IC2IE50715.2020.9274594.
- [23] W. K. Sedano and M. Salman, “Auditing Linux Operating System with Center for Internet Security (CIS) Standard,” in 2021 International Conference on Information Technology (ICIT), Jul. 2021, pp. 466–471. doi: 10.1109/ICIT52682.2021.9491663.
- [24] P. Mihăilă, T. Balan, R. Curpen, and F. Sandu, “Network Automation and Abstraction using Python Programming Methods,” *MACRo 2015*, vol. 2, Oct. 2017, doi: 10.1515/macro-2017-0011.
- [25] S. Lee, T. Wong, and H. S. Kim, “To Automate or Not to Automate: On the Complexity of Network Configuration,” in 2008 IEEE International Conference on Communications, Beijing, China: IEEE, 2008, pp. 5726–5731. doi: 10.1109/ICC.2008.1072.
- [26] M.-A. Ilie and C. I. Rîncu, “Convergence and security improvements by using automation in DMVPN networks,” in 2022 14th International Conference on Communications (COMM), Jun. 2022, pp. 1–5. doi: 10.1109/COMM54429.2022.9817308.

- [27] A. Babaei, P. M. Kebria, and S. Nahavandi, “A survey on Automation Technologies used in Network Control and Management,” in 2022 15th International Conference on Human System Interaction (HSI), Jul. 2022, pp. 1–6. doi: 10.1109/HSI55341.2022.9869444.
- [28] O. W. J. Altalebi and A. A. Ibrahim, “Optimization of Elapsed Time of Automation for Large-Scale Traditional Networks and Proposing New Automation Scripts,” in 2022 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA), Jun. 2022, pp. 1–10. doi: 10.1109/HORA55278.2022.9799873.
- [29] “(PDF) Implementation of Network Automation Using Ansible to Configure Routing Protocol in Cisco and Mikrotik Router with Raspberry PI.” Accessed: Jul. 19, 2023. [Online]. Available: https://www.researchgate.net/publication/345158050_Implementation_of_Network_Automation_Using_Ansible_to_Configure_Routing_Protocol_in_Cisco_and_Mikrotik_Router_with_Raspberry_PI
- [30] G. S. Santyadiputra, I. M. E. Listartha, and G. Saskara, “The effectiveness of Automatic Network Administration (ANA) in network automation simulation at Universitas Pendidikan Ganesha,” *J. Phys. Conf. Ser.*, vol. 1810, p. 012028, Mar. 2021, doi: 10.1088/1742-6596/1810/1/012028.
- [31] S. Yadav, “Research Paper on Network Automation,” *Int. J. Res. Appl. Sci. Eng. Technol.*, vol. 7, no. 4, pp. 1446–1450, Apr. 2019, doi:

10.22214/ijraset.2019.4261.

[32] S. Mysari and V. Bejgam, “Continuous Integration and Continuous Deployment Pipeline Automation Using Jenkins Ansible,” in 2020 International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE), Feb. 2020, pp. 1–4. doi: 10.1109/ic-ETITE47903.2020.239.

[33] E. Horton and C. Parnin, “Dozer: Migrating Shell Commands to Ansible Modules via Execution Profiling and Synthesis,” in 2022 IEEE/ACM 44th International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP), May 2022, pp. 147–148. doi: 10.1145/3510457.3513060.

[34] V. Shvetcova, O. Borisenko, and M. Polischuk, “Using Ansible as Part of TOSCA Orchestrator,” in 2020 Ivannikov Ispras Open Conference (ISPRAS), Dec. 2020, pp. 109–114. doi: 10.1109/ISPRAS51486.2020.00023.

[35] “Puppet Infrastructure & IT Automation at Scale | Puppet by Perforce.” Accessed: Jul. 19, 2023. [Online]. Available: <https://www.puppet.com/>

[36] J. A. Shah and D. Dubaria, “NetDevOps: A New Era Towards Networking & DevOps,” in 2019 IEEE 10th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON), Oct. 2019, pp. 0775–0779. doi: 10.1109/UEMCON47517.2019.8992969.

[37] “Chef Software DevOps Automation Solutions | Chef.” Chef Software. Accessed: Jul. 19, 2023. [Online]. Available: <https://www.chef.io/>

- [38] R. Kostromin, “Survey of software configuration management tools of nodes in heterogeneous distributed computing environment,” Jul. 2020. doi: 10.47350/ICCS-DE.2020.15.
- [39] “Saltproject.io.” Accessed: Jul. 19, 2023. [Online]. Available: <https://saltproject.io/index.html>
- [40] P. Masek, M. Stusek, J. Krejci, K. Zeman, J. Pokorny, and M. Kudlacek, “Unleashing Full Potential of Ansible Framework: University Labs Administration,” in 2018 22nd Conference of Open Innovations Association (FRUCT), May 2018, pp. 144–150. doi: 10.23919/FRUCT.2018.8468270.
- [41] “Welcome to Paramiko! — Paramiko documentation.” Accessed: Jul. 19, 2023. [Online]. Available: <https://www.paramiko.org/>
- [42] “Performance Comparison of SSH Libraries,” Journal of University of Shanghai for Science and Technology. Accessed: Jul. 19, 2023. [Online]. Available: <https://jusst.org/performance-comparison-of-ssh-libraries/>
- [43] Z. Li, B. Zhou, W.-G. Zhou, T.-Q. Xu, and X.-F. Zhang, “Application Research on an Automated Batch Network Reinforcement Method Based on SSH Protocol,” 2022, pp. 101–109. doi: 10.1007/978-981-19-0390-8_13.
- [44] “Welcome to nornir’s documentation! — nornir 3.1.1 documentation.” Accessed: Jan. 24, 2023. [Online]. Available: <https://nornir.readthedocs.io/en/latest/index.html>

- [45] M. F. Mohd Fuzi, K. Abdullah, I. H. Abd Halim, and R. Ruslan, “Network Automation using Ansible for EIGRP Network,” *J. Comput. Res. Innov.*, vol. 6, no. 4, pp. 59–69, Sep. 2021, doi: 10.24191/jcrinn.v6i4.237.
- [46] D. Soderman and P. Pekkanen, “NETWORK HARDENING OF MULTI-LAYER SWITCH,” p. 22.
- [47] H. M. D. G. V. Perera, K. M. Samarasekara, I. U. K. Hewamanna, D. N. W. Kasthuriarachchi, K. Y. Abeywardena, and K. Yapa, “NetBot - An Automated Router Hardening Solution for Small to Medium Enterprises,” in *2021 IEEE 12th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*, Vancouver, BC, Canada: IEEE, Oct. 2021, pp. 0015–0021. doi: 10.1109/IEMCON53756.2021.9623186.
- [48] I. Ortiz-Garces, A. Echeverria, and R. O. Andrade, “Automation Tasks Model for Improving Hardening Levels on Campus Networks,” in *2021 Fifth World Conference on Smart Trends in Systems Security and Sustainability (WorldS4)*, London, United Kingdom: IEEE, Jul. 2021, pp. 30–35. doi: 10.1109/WorldS451998.2021.9514030.
- [49] K. Palani and D. M. Nicol, “Hardening Critical Infrastructure Networks Against Attacker Reconnaissance,” in *Quantitative Evaluation of Systems*, M. Gribaudo, D. N. Jansen, and A. Remke, Eds., in *Lecture Notes in Computer Science*. Cham: Springer International Publishing, 2020, pp. 258–275. doi: 10.1007/978-3-030-59854-9_19.

- [50] “Network Automation | European Journal of Technology.” Accessed: Oct. 24, 2023. [Online]. Available: <https://ajpojournals.org/journals/index.php/EJT/article/view/1547>
- [51] D. Sikeridis, P. Kampanakis, and M. Devetsikiotis, “Assessing the overhead of post-quantum cryptography in TLS 1.3 and SSH,” in Proceedings of the 16th International Conference on emerging Networking EXperiments and Technologies, in CoNEXT ’20. New York, NY, USA: Association for Computing Machinery, Nov. 2020, pp. 149–156. doi: 10.1145/3386367.3431305.
- [52] S. González, Á. Herrero, J. Sedano, U. Zurutuza, and E. Corchado, “Different approaches for the detection of SSH anomalous connections,” Log. J. IGPL, vol. 24, no. 1, pp. 104–114, Feb. 2016, doi: 10.1093/jigpal/jzv047.
- [53] C. Smera and J. Sandeep, “Networks Simulation: Research Based Implementation using Tools and Approaches,” in 2022 IEEE 3rd Global Conference for Advancement in Technology (GCAT), Oct. 2022, pp. 1–7. doi: 10.1109/GCAT55367.2022.9972119.
- [54] L. Campanile, M. Gribaudo, M. Iacono, F. Marulli, and M. Mastroianni, “Computer Network Simulation with ns-3: A Systematic Literature Review,” Electronics, vol. 9, no. 2, Art. no. 2, Feb. 2020, doi: 10.3390/electronics9020272.
- [55] A. Alrashide, M. S. Abdelrahman, I. Kharchouf, and O. A. Mohammed, “GNS3 Communication Network Emulation for Substation GOOSE Based Pro-

tection Schemes,” in 2022 IEEE International Conference on Environment and Electrical Engineering and 2022 IEEE Industrial and Commercial Power Systems Europe (EEEIC / I&CPS Europe), Jun. 2022, pp. 1–6. doi: 10.1109/EEEIC/ICPSEurope54979.2022.9854689.

[56] E. Tego, V. Attanasio, and F. Matera, “GNS-3 Emulation Platform to Study Wide Area Network Performance in Contexts Close to Reality,” in 2022 AEIT International Annual Conference (AEIT), Oct. 2022, pp. 1–6. doi: 10.23919/AEIT56783.2022.9951844.

[57] X. Hu and Y. Zhou, “Wireshark Packet Dissector for DL/T 860 Protocol,” in 2022 4th International Conference on Electrical Engineering and Control Technologies (CEEECT), Dec. 2022, pp. 30–34. doi: 10.1109/CEEECT55960.2022.10030402.

[58] R. K. CV and H. Goyal, “IPv4 to IPv6 Migration and Performance Analysis using GNS3 and Wireshark,” in 2019 International Conference on Vision Towards Emerging Trends in Communication and Networking (ViTECoN), Mar. 2019, pp. 1–6. doi: 10.1109/ViTECoN.2019.8899746.

[59] S. U. Masruroh, K. H. P. Widya, A. Fiade, and I. R. Julia, “Performance Evaluation DMVPN Using Routing Protocol RIP, OSPF, And EIGRP,” in 2018 6th International Conference on Cyber and IT Service Management (CITSM), Aug. 2018, pp. 1–6. doi: 10.1109/CITSM.2018.8674051.

[60] F. Waheed and M. Ali, “Hardening CISCO Devices based on Cryptogra-

phy and Security Protocols - Part II: Implementation and Evaluation,” Ann.
Emerg. Technol. Comput., vol. 2, no. 4, pp. 11–27, Oct. 2018, doi:
10.33166/AETiC.2018.04.002.

VITAE

Personal Information

- Name: Ali Bello Imoukhuede
- Nationality: Nigeria
- Date of Birth: 10/10/1990
- Email: *alibelloimoukhuede@gmail.com*
- Permanent Address: Building No. 725 Shehu Yakubu Street, Yankaba, Kano -
Nigeria
- Postal Code: 700102

Academic History

- King Fahd University of Petroleum and Minerals (KFUPM)
 - MSc Computer Engineering
- Bayero University Kano (BUK)
 - BSc Computer Engineering

Academic Publication

- “A Systematic Literature Review on Vulnerabilities, Mitigation Techniques, and Attacks in Field-Programmable Gate Arrays” (in-view). Ali Alsuwaiyan, Aliyu Abubakar Habib, **Ali Bello Imoukhuede**, Mohamed Osman Omar, Md. Al Maruf, Mansour Alqarni, Aiman El-Maleh, Abdulaziz Tabbakh, Muhamad Felemban, Akramul Azim, (2023)
- “Assessing Network Simulation Platforms for Network Automation” (manuscript under preparation). **Ali Bello Imoukhuede**, Prof. Tarek R. Sheltami, (2023)

Professional Certifications

- Cisco Certified Network Professional (CCNP)
- Cisco Devnet Associate (DevNet)
- Certified Network Security Specialist (CNSS)