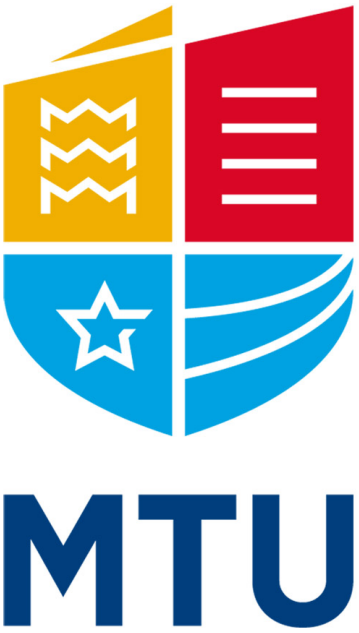

Network Access Control for Endpoint Devices

by

Dipendra Ale

This thesis has been submitted in partial fulfillment for the degree of Bachelor of Science in IT Management

in the
Faculty of Engineering and Science
Department of Computer Science

May 2023

Declaration of Authorship

This report, Network Access Control for Endpoint Devices, is submitted in partial fulfillment of the requirements of Bachelor of Science in IT Management at Munster Technological University Cork. I, Dipendra Ale, declare that this thesis titled, Network Access Control for Endpoint Devices and the work represents substantially the result of my own work except where explicitly indicated in the text. This report may be freely copied and distributed provided the source is explicitly acknowledged. I confirm that:

- This work was done wholly or mainly while in candidature Bachelor of Science in IT Management at Munster Technological University Cork.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at Munster Technological University Cork or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this project report is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed: Dipendra Ale

Date: 02/05/2023

MUNSTER TECHNOLOGICAL UNIVERSITY CORK

Abstract

Faculty of Engineering and Science
Department of Computer Science

Bachelor of Science

by Dipendra Ale

Traditional solutions that operate individually are no longer adequate since attacks continue to grow in complexity, variety, and performance. For us to proactively prevent such threats, we want a more unified security solution and this is where the Network Access Control solution plays its part as a real-world security platform. This project analyses the effectiveness of NAC as a security solution, as well as different vendor solutions, protocol support, and the interoperability of different vendors while pinpointing endpoint devices. While we agree that NAC is a great solution, we aim to provide the best recommendations in this research for future modification of this security solution as well as deployment options for the university network.

Lastly, We will explore and implement an open-source NAC called PacketFence in our own local machine to suit the requirement of a University.

Acknowledgements

I would like to take this opportunity to thank everyone who assisted me through their guidance throughout this project.

Hanmdan Awan - Project Supervisor Semester 1

Roland Katona - Project Supervisor Semester 2

Contents

List of Figures

List of Tables

Abbreviations

NAC	**N**etwork **A**ccess **C**ontrol
NAP	**N**etwork **A**ccess **P**rotection (Microsoft term)
NAC	**N**etwork **A**dmission **C**ontrol (Cisco term)
GNS3	**G**raphical **N**etwork **S**imulator-3
AD	**A**ctive **D**irectory
OU	**O**rganizational **U**nit
MAC	**M**edia **A**ccess **C**ontrol
GUI	**G**raphical **U**ser **I**nterface
VLAN	**V**irtual **L**ocal **A**rea **N**etwork
DHCP	**D**ynamic **H**ost **C**onfiguration **P**rotocol
DNS	**D**omain **N**ame **S**ystem
NAT	**N**etwork **A**ddress **T**ranslation
AAA	**A**uthenticaion **A**uthorisation **A**ccounting
ACL	**A**ccess **C**ontrol **L**ist
TCP	**T**ransmission **C**ontrol **P**rotocol
DDoS	**D**istributed **D**enial **O**f **S**ervice
IDS	**I**ntrusion **D**etection **S**ystem
IPS	**I**ntrustion **P**rotection **S**ystem
HTTPS	**S**ecured **H**ypertext **T**ransmission **P**rotocol
RAM	**R**andom **A**ccess **M**emory
DDoS	**D**istributed **D**enial **O**f **S**ervice
API	**A**pplication **P**rogramming **I**nterface
WMI	**W**indows **M**anagement **I**nstrumentation
OSI	**O**pen **S**ystems **I**nterconnection
RADIUS	**R**emote **A**uthentiation **D**ial **I**n **U**ser **S**ervice

For/Dedicated to/To my... Family

Chapter 1

Introduction

1.1 Motivation

During my internship at Boston Scientific as a Deskside Technician, I worked with various endpoint devices such as Laptops, PCs, printers, phones, etc. Managing networks within the site was also a big part of the work involved where I had to configure and manage network ports on the switches to activate and assign the device to the appropriate VLAN group. I always wondered what kind of security measures were put in place to secure these systems and devices in such a large organization. Even though I didn't get much experience in the Cybersecurity field during my time in the company, I wanted to explore what solutions are available and how they can be implemented into a corporate network.

In today's businesses, networks store a lot of data and information. These data include everything from client details to crucial company data. This constantly turns organizations into targets and exposes them to a variety of security risks, such as unauthorized access, which always has serious repercussions and deteriorating impacts on organizational networks. Endpoint systems significantly increase the amount of threat to organizational networks. It is highly likely that the entry of viruses, worms, backdoors, and spyware into corporate networks may pose a threat to the entire network infrastructure given the rise in users connecting to the network using laptops and other portable devices like internet phones, PDAs, and home entertainment equipment.

The networks are open systems without well-defined security perimeters where users move their laptops and mobile devices into and out of the workplace. Contractors, vendors, and site visitors may require access to the physical network to perform the necessary work. Even employees in the office with access to the network are even subject

to threats through internet access, e-mail, instant messages, etc. Remote access users can connect from both their homes and public spaces. This is especially evident during the past two years of the Covid-19 phase when people were encouraged to work from home and access companies' networks through the use of VPN, SD-WAN (Software-Defined WAN), or SASE (Secure Access Service Edge) remotely. Remote work is becoming more and more popular by the day, and it is defiantly here to say in the long term and hackers will look to exploit this area. The traditional security tools and products such as intrusion detection and prevention (IDS/IPS), antivirus, and firewalls are not sufficient as the network is too diverse to solely rely on these solutions.

Wireless communication networks have expanded significantly in recent years, because of their advantages over conventional networks including Mobility and comfort. A quick click of a button and being able to connect to an organization's network remotely seems very practical and easy but this also brings various security issues such as being vulnerable to malicious programs. There have recently been several concerns that seek to address how the network can be safeguarded against external endpoint connecting devices like laptops, portable devices, and more recently, mobile phones, which are constantly connected to the corporate network.

How should we approach tracking the security compliance of these devices?

Is the security solution resilient enough to secure these endpoint devices?

There have been security solutions invented in the past such as the introduction of WEP (Wired equivalent privacy) protocol published by IEEE in 1999 as a security component of 802.11b addressed some levels of security such as Privacy, Data Integrity, and Authentication but showed major security weaknesses. Since then, there are various WEP cracking software tools such as WEPAttack and WEPCrack. After several years, Wi-Fi Protected Access (WAP) protocol was introduced as a more secure alternative which was published by the Wi-Fi Alliance in 2003 [1].

Network Access Control (NAC) is covered in this project with a particular focus on securing endpoint devices. The purpose of this thesis is to investigate various NAC Implementations and their significance in today's world. Securing a network from unauthorized users is becoming more crucial than ever as the world becomes increasingly linked and digitalized. Giving a hacker easy access to the network is the most straightforward approach to creating data leaks. Because of this, network access control and cyber security must advance over time.

1.2 Contribution

The contribution aspect of this thesis is derived from the modules covered in the IT Management degree whereby we will look to showcase our learnings from various modules and integrate them into the thesis. Cybersecurity and Networking are the two major areas taught and covered in my degree which is the foundation of my thesis topic. We have been introduced to various modules that cover these areas since 1st year such as Scaling and Managing Networks, Networking Fundamentals, Network Security, and Security Monitoring. Our plan is to unitize our knowledge from these modules as well as not limit ourselves to only relevant modules to our thesis but use modules such as Requirement Engineering to visually create user scenarios. Another key source of knowledge is the learnings from my internship where I worked in Datacenters, end-point devices, and key infrastructure alongside the IT team.

The key contribution structure is listed as follows:

- Using Visual Diagram tool to showcase diagrams related to NAC
- NAC solution selection process and comparative analysis
- Implementing Open-Source NAC solution PacketFence to suit university standards using Virtual Machines
- Setup an Active Directory to integrate with PacketFence NAC
- Analysis of functionality of NAC with a focus on endpoint devices using PacketFence

1.3 Structure of This Document

The document is structured as follows:

- Chapter 1 contains an introduction to our project which elaborates on motivation, executive summary, and contributions to the project.
- Chapter 2 contains the background of our project which elaborates on the thematic area within the Computer Science of the project.
- Chapter 3 contains the Problem which elaborates on the problem definition, objectives, and functional and non-functional requirement sections of the project.

- Chapter 4 contains the implementation approach to our project which elaborates on the architecture, risk assessment, methodology, implementation plan schedule, evaluation, and prototype of the project.
- Chapter 5 contains the Conclusion of the research phase of our project which elaborates on the discussion, conclusion, and future work of the project.

Chapter 2

Background

2.1 Thematic Area within Computer Science

The two main thematic areas that are covered within Computer Science are Networking and Cybersecurity. The core subject of this project is an in-depth analysis of Network Access Control (NAC) as a security solution focused on end-point devices. Any device that connects to a network from outside the firewall of an organization is considered a network endpoint. Endpoint examples include the following:

- Laptops
- PCs
- Tablets
- Mobile devices
- IoT devices
- Printers
- Scanners
- Medical devices

The basic goal of network security is to make sure that users can freely utilize the network without worrying about losing or compromising our valuable resources and data. Therefore, network security for any organization must safeguard network computers and protect data throughout both their transmission over the public internet and their storage on computer systems and storage devices.

Some common attacks on the networks include [2]:

- Unauthorized access – This refers to when attackers gain access to a network without authorization. Weak passwords, a lack of social engineering defense, previously compromised accounts, and insider threats are a few of the factors that contribute to unauthorized access attempts.
- Distributed Denial-of-Service (DDoS) - The main objective of this attack is to attempt to make a system resource unavailable to its intended users. Attackers build and use botnets, which are a large number of compromised devices that are then used to direct unwanted traffic to the network. DDoS can happen at the application or network level, for instance by running complex SQL queries that completely overwhelm a database or sending massive amounts of SYN/ACC packets that overwhelm a server.
- Man In The Middle Attack – A man-in-the-middle attack entails attackers intercepting traffic, either within our network or between our network and external sites. Attackers can steal data that is being communicated to gain user credentials or even take over users' sessions if communication protocols are not secured.
- SQL Injection Attack – Numerous websites accept user input without validating or sanitizing it. After that, attackers can submit a form or perform an API call by passing malicious code in place of the desired data values. The code is run on the server, giving attackers the ability to compromise it.
- Privilege escalation – After breaking into our network, an attacker can utilize privilege escalation to increase their influence. Attackers can escalate their privileges for the same systems either horizontally or vertically. Horizontal privilege escalation entails acquiring access to additional, nearby systems.

Common attacks on endpoint devices and end users [2]:

- Malicious software – These programs or software possess the ability to self-inflict and harm systems. This is also commonly referred to as malware. Some other types of Malicious software are viruses, Trojan horses, worms, backdoors, spyware, etc.
- Ransomware – This is a form of malware that uses encryption to keep a victim's data hostage for a fee. Critical user information is encrypted and locked behind ransomware, which remains locked until the victim complies with the cybercriminal's demands. Due to its rapid spread capabilities, an entire organization might be quickly destroyed.

- Phishing – This is a type of cybercrime where login credentials are obtained and access private data. This is one of the most common endpoint security threats. Phishing schemes are typically started by clicking on email links, which can range from being obvious to well-organized and well-planned out.
- Identity Spoofing – An attacker can pose as the legitimate user without needing that user's login credentials. Common identity spoofing techniques include network spoofing, message replay, software exploitation assaults, and man-in-the-middle attacks.
- Password Sniffing – Sniffing passwords is a term used to describe a group of software applications that are essentially used to log in remotely by capturing usernames and passwords. SMTP, FTP, and Telnet are a few examples of network applications. For authentication, users might provide their username and password therefore this information may be intercepted by password sniffers.
- Eavesdropping – This is one of the oldest techniques for stealing information in network and computer communication. It enables an attacker to intercept electronic data from network traffic using sniffer software e.g., TCPdump and Wireshark.

Here are a few figures to support the idea that endpoint security will be crucial for every company in 2022 and beyond [3]:

- In 2021, successful ransomware attacks affected 53 percent of firms, and 77 percent of those were struck more than once.
- In 2019, endpoint attacks affected 68 percent of enterprises.
- In comparison to web-based attacks, which account for 23 percent of cases, and Office documents, which were 45 percent of cases, email accounts for 94 percent of malware delivery methods.
- 91 percent of cyberattacks begin with phishing emails, and consumers are most likely to fall for them out of curiosity, fear, and urgency.
- Only about 15 percent of companies are utilizing endpoint security solutions.
- According to a recent FBI report, up to 4,000 complaints about cyberattacks are sent to its Cyber Division each day which was a 400 percent increase above what they were observing before the coronavirus.

There was a study conducted on the state of an endpoint security risk by Ponemon Institute [4] demonstrates that organizations are failing to reduce their endpoint security risk, particularly against new and unidentified threats. In fact, 68 percent of the

respondents in the year's study which increased from 54 percent in 2017 reported that their business has been a victim of one or more endpoint attacks that have been able to compromise data or IT infrastructure over the past 12 months.

2.2 A Review of Network Access Control

There are a wide variety of current solutions and best practices to combat these security risks such as Antivirus, Firewall, IDS/IPS, VPN, SSL, and HTTPS. However, my research enabled me to recommend NAC as the unified solution while also integrating other security solutions as listed above. NAC limits access to network resources according to outlined or defined policies, hence enforcing network security. NAC unifies a variety of network security systems, including Identity Management, Firewalls, IDS, and Anti-Virus software. NAC tracks who have access to the network at any given time, the devices that are used to control access, employees, visitors, and requests related to Wi-Fi access, as well as remote users and clients who can be blocked on the off chance that they are unintentionally or maliciously manipulating the network.

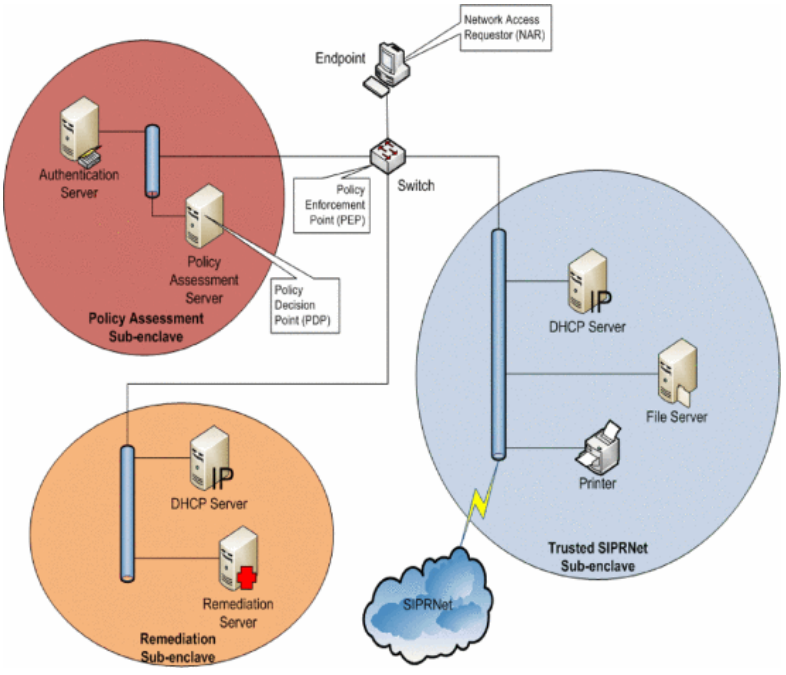

FIGURE 2.1: NAC Architecture[5]

Figure 2.1 above shows how NAC functionality is frequently provided by establishing sub-enclaves, each with a distinct purpose. One is a secure network, another is where broken clients are repaired, and a third (the entrance point) may be thought of as an enclave where authentication and integrity assessments take place. NAC solutions are based on NAC policies that are defined on a central policy server and implemented by

network infrastructure components (switches, routers, firewalls, and so on). Authentication, authorization, and accounting may also be handled by a separate server or servers. Many commercial NAC systems employ the IEEE 802.1x protocol for authentication and enforcement, as well as proprietary software for the policy server and endpoint agent. Early NAC solutions focused mostly on policy management and enforcement. Today's solutions generally expand on that, adding capabilities like endpoint profiling, guest management, visibility, and analytics, as well as enhanced support for BYOD (Bring Your Own Device) scenarios.

Policies provide a variety of key NAC benefits. Instead of manually approving or denying access on a per-device or user basis, a network or system administrator may set the conditions required for access. NAC isn't always all or nothing, and more advanced rules may offer visitors or contractors varying levels of network access than full-time employees. Devices can also be "quarantined", which gives them just enough access to update software or conduct other remedial action without enabling them to access the rest of the internal network.

The policy-based approach utilized in most NAC systems offers a significant level of scalability and flexibility. Admins may add or amend policies at any moment, changing the rules that control access for tens of thousands of devices virtually quickly. This is a key ability for dealing with fast-moving threats like worms or ransomware that may exploit recently publicized vulnerabilities. During high-stakes events like WannaCry or NotPtyta, an organization could dramatically lower its risk by separating unpatched devices from the rest of the network. Endpoints without anti-virus software, intrusion detection and prevention software, operating system, and updates are examples of vulnerable systems. NAC not only checks for the presence of various security software, but also ensures that end systems have the most recent virus definitions, intrusion detection and prevention have the most recent updates, and operating systems have the most recent updates, patches, and hotfixes. The system will verify the endpoints to see whether they match the security baselines and then restrict access to systems that lack any or all of this security software. This is a preventative strategy for the cooperative network since it decreases network vulnerability to assaults and the possibility of worms and malware infecting other systems.

NAC solutions fall into two categories:

- Clientless - There is no requirement for any software to be installed to assist with the NAC process.
- Client-based - There is a requirement for a software component to be pre-installed to assist in the NAC process.

There are several NAC solutions available, each of which may work in a different manner. As a broad rule of thumb, network control may be imposed in two ways:

- Pre-Admission NAC refers to NAC technology that conducts an evaluation using a set of standards before granting access to a network. If these requirements are not satisfied, it will not allow the devices to connect to the network. [3] Pre-Admission NAC can be found in solutions such as Microsoft NAP, Cisco NAC, and Mobile NAC.

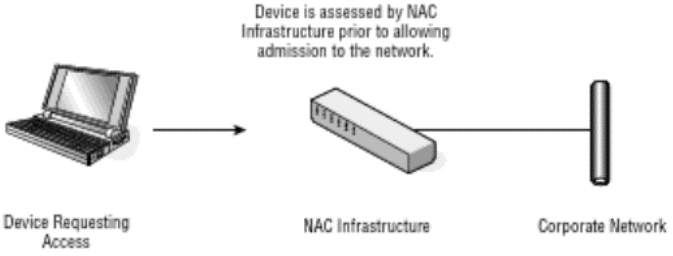

FIGURE 2.2: Pre-Admission NAC[6]

- Post-Admission NAC occurs after a client has been granted access to the network. This functionality is crucial since the device's security posture may change after it has been granted access to the network for the first time. Additionally, restrictions may be necessary based on how the device behaves once connected to the network.

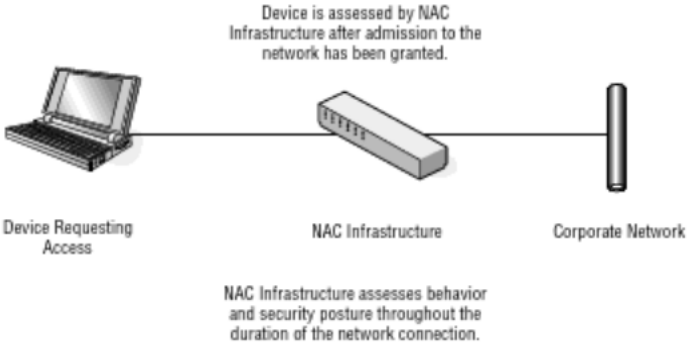

FIGURE 2.3: Post-Admission NAC[6]

NAC can also be based on where the network's decision-making and enforcement processes are located:

- Out-of-band solutions often employ a policy server that is not directly connected to the network. The policy server interfaces with network infrastructure devices

including switches, routers, and wireless access points, which implement NAC policies and allow or reject traffic as needed.

- Inline NAC solutions integrate decision-making and enforcement at a single location inside the usual flow of traffic. For bigger networks, this can demand a significant amount of resources and has the potential to negatively impair network performance if something goes wrong.

2.2.1 Network Access Control Processes and Functions

There are several network access control solutions available today. There is no universal standard of network access control; different institutions have their own setup-level justifications to characterize NAC. NAC functions go through three main stages [6]:

- NAC Awareness
- NAC Standard
- NAC Interoperability

According to various research, NAC is in the second stage of "standards" currently; there is a strong focus on NAC standards at the moment, with individuals from various groups working together to standardize NAC. The following is the key set of features offered by NAC solutions [6]:

- Remediation

  Isolated devices are immediately relocated to the isolated network and may be unable to access the protected network. Remediation allows the isolated to recover from non-compliant health status to a compliant state before being permitted to rejoin the network. Installing and setting new patches, anti-virus software program updates, signature and certificate updates, IDS systems, and firewalls are all part of the process. The system is allowed access after it has received all of the necessary upgrades.

- Authorization

  This refers to the procedures or methods used to identify a person based on their user credentials. After a client/user connects to a secure network, that is after passing the validation and posture evaluation steps and achieving compliant status, NAC certifies every resource accessed by the user within the internal network. The authorization process is carried out by the authentication, authorization, and

accounting systems triple AAA, Remote Authentication Dial-in UserService (RADIUS), and DIAMETER, which operates inside the Internet protocol stack's application layer protocol.

- Policy Implementation

  NAC enforces predefined policies on an end-user computer, granting the triple A (AAA) system access to policies specified for devices connected to a secure network. It employs several techniques to impose policy on systems, including the use of VLAN to establish decisions that limit users' ability to connect to a certain VLAN. Firewalls are also used to enforce policies on end-users and devices by establishing rules such as activating and disabling ports, banning specific URL websites, intrusion detection systems(IDS), and so on. Access control lists, like firewalls, are frequently used to configure network access rules and pass them to a switch, router, or any gateway server to enforce the access list policy on end-point devices.

- Posture Evaluation

  This is a unique aspect of NAC that refers to using a set of predefined criteria to determine network device compliance or to determine a device's degree of trust before granting access. Posture evaluation techniques entail running a number of tests on an end-point device along with perceptions and measurements, then forwarding the data to the NAC policy server to access the machine's consistency or compliance status to check for digital certificates that have been approved or validated creates signature files for intrusion prevention systems and all trusted application program lists.

- Validation

  NAC servers are set to authenticate or validate all users who enter the protected network; the following are standard current authentication or validation procedures in NAC:

  - Hypertext Transfer Protocol (HTTP)
  - Point-to-Point Tunneling Protocol (PPP)
  - IPSec, often known as IP security
  - Secure Socket Layer and Transport Layer Security (SSL/TLS)
  - Virtual Private Network (VPN)
  - Dynamic Host Control Protocol (DHCP)
  - IEEE 802.1X

- Node Identification

  This refers to devices that acquire access to a secure, protected network. The process and function involved are critical to NAC because they must be aware of each node that connects to the local network in order to extend NAC capabilities and processes such as post-affirmation control, validation, authorization, policy implementation, isolation, remediation, and posture evaluation. Node identification is accomplished in a variety of methods to identify nodes accessing the network, as well as on several tires, depending on the manner of access. Virtual private networks(VPN), wireless local area networks(WLAN), cable or wired local area networks, and dial-up connections are examples of typical modes of access.

- Isolation

  This is a modern feature incorporated into NAC with the main goal of isolating problematic devices from the secured and protected network so that the network is safe from unprotected compliance devices. Configuring VLAN to a single distinct network route communications to a specific network achieves isolation.

- Post-Admission Control [7]

  This is equivalent to threat mitigation. When a device is confirmed compliant and connects to the private network - users, nodes, and their sessions are monitored for any malware activity or policy breaches. In the event of such activities, the user's access can be restricted by quarantining or terminating the session. Post-admission control functions similarly to the capabilities of Intrusion Prevention Systems(IPS).

2.2.2 Why implement NAC?

Many companies approach network security in sections such as installing a firewall here and an anti-virus solution there. Using fully different systems for controlling access rights, on the other hand, leads to massive disarray and a significant amount of administration overhead. This is where Network Access Control comes in to enable a fully centralized approach to network security. This is achieved by leveraging your Active Directory (or other directory systems/multi-auth) and allowing you to build a set of rules (or policies) that are enforced on any device that seeks to connect to the network.

Security rules were rigorous and easy to maintain in the days of cable connections and computer laboratories where you knew what devices you needed to support and where they were since you bought them.

However, as IoT and BYOD devices have grown in popularity, so have the security and cyber threats. The visual representation of statistics on the total number of device connections can be seen in diagram 2.2 below:

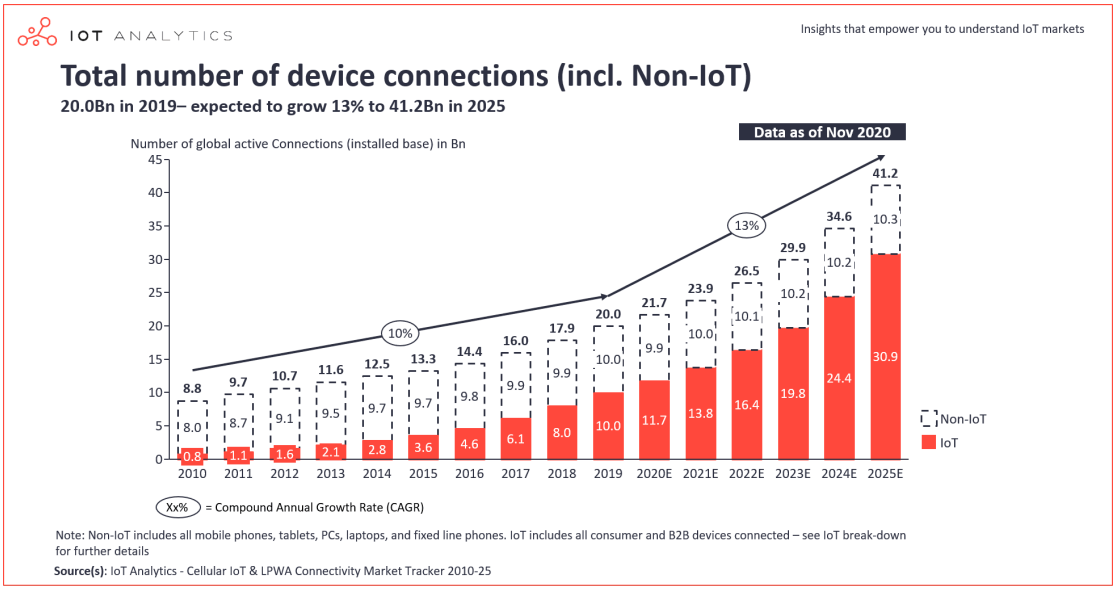

FIGURE 2.4: [8]

With immense technological power comes huge network responsibility. IT professionals are now faced with a major challenge: "How can I monitor network traffic for malicious activity while yet providing scalable network access?". NAC is the solution to this and should be at the top of everyone's security must-have list. Here are the main three reasons to roll out network access control solutions:

- Improve network visibility

  Managing network access begins with having adequate visibility. It will be impossible to provide a stable and secure network experience if you cannot identify who or what is using your network and where they are attempting to go. We can use NAC to determine these questions of who, what, where, when, and how an end-use or endpoint device is accessing your network.

- Minimize cyber threats

  According to Purplesec's 2021 trend report [9], cybercrime has increased 600 percent as a result of the COVID pandemic. Any network security flaws that exist might be exploited by an attacker. Network access control reduces cyber dangers in two ways: role-based access control and malware scope limitation. Role-based access control (also known as RBAC) is a security approach in which users are

assigned roles depending on their position within the organization. RBAC is identical to parental controls for monitoring a child's internal but on a broader and more finite scale.

Role-based access control, with pre-defined policies and assigned roles, enables you to design the ideal wireless experience for any type of end-user/device accessing your network. When it comes to minimizing cyber attacks, NAC comprises two crucial steps: Authentication and Authorization [8].

Authentication occurs when the system checks the user's credentials, whereas authorization occurs when the system permits or refuses access depending on the policies in place. In order to gain network access, a user must complete both procedures. The method of assigning and implementing security policies based on those roles (also known as endpoint integrity) allows you to manage the behavior of any devices attempting to connect to the network.

The second method that network access control boosts security is by narrowing the scope of malicious activity. Every day, 360,000 new harmful files are unknowingly downloaded by employees, according to Kaspersky [10]. So, even an authorized user may mistakenly run their device rogue by opening a malicious email, which can propagate throughout the network if sufficient access restrictions are not in place.

NAC can safeguard intellectual property and sensitive data against illegal user, capture, or manipulation. In other words, even if the device/user has access to a specific piece of information within the network, if they are barred from "operating inappropriately" with it, the device/user has access to it. As a result, NAC ensures that your end-users are following the terms of service and that viruses are not being propagated unknowingly (or purposely) over the network.

- Enhance network performance

  One benefit of NAC that is frequently overlooked is how it is able to increase network performance. Companies often install numerous SSIDs for staff as a workaround for adopting the NAC solution. It can accomplish the job on a basic level, but each SSID that is broadcast consumes bandwidth. As a result, each time you assign a different password to a different end-user on the network, you drastically hamper everyone's performance.

  With a NAC solution's role-based access control capabilities, you may have all users on the same SSID since policy enforcement is based on their position within the company. Each SSID removed will restore around 10 percent of the bandwidth. In more complicated environments- say, a university - up to 7 or 8 SSIDs may be

used. However, by adopting a NAC solution, we were able to return 40-50 percent of bandwidth back just by removing the overhead caused by the additional SSIDs.

You also could even construct job-specific bandwidth contracts that limit or reserve rates based on your role. This is used to prioritize particular programs or users above others in the traffic, such as an employee over a guest. It is evident that NAC not only improves security but also network performance.

Chapter 3

Problem - Network Access Control

3.1 Problem Definition

Connectivity is quite broadly extended in today's network setups. In order to communicate with servers and other network resources, users can connect their devices to any access point or outlet for connection. Over time, the majority of these endpoint devices may turn out to be prime targets for penetration, endangering the security of the organization. These dangers can vary from unwanted to catastrophic occurrences that could possibly bring down the whole network of an organization. We could look at our own University as a prime example. Most students connect to the campus network using their own laptops, phones, etc. This increases the danger of the network being compromised as these devices may have been in various locations before connecting to the campus network as it introduces vulnerabilities to the network.

3.2 Objectives

The main output of this project is to address the shortcoming of the networks through the implementation of NAC solutions. The project can be broken down into three phases which are investigation, deployment, and configuration. The first phase of the investigation is to research different types of NAC solutions provided by various vendors and perform an in-depth comparative analysis. In this thesis, we have chosen Cisco NAC Appliance (Cisco Clean Access) and Microsoft Network Access Protection (NAP). We plan to utilize a conceptual diagram to visually demonstrate the operation and use

case of these solutions using the Visual Paradigm application. The detailed analysis will provide a good understanding of each solution and help them to consider these solutions according to their requirements. The analysis is carried out in the following sections:

- Deployment scenarios and topologies along with their architecture
- Components of each solution
- Security Posture
- Policy and Access Control
- Interoperability
- Cross-platform Support
- Reporting Mechanism

Once we have analyzed the solutions, we proceed to incorporate our findings to suit universities' networks and provide a suitable recommendation. The last phase will involve implementing an open-source NAC solution called PacketFence and configuring it to suit university network standards. We will be utilizing various tools such as Virtual Application, Ubuntu 18.04, SSH, and RDP (Remote Desktop Protocol).

- Configuration setup
- LDAP configuration for Staff/Student Login
- Configuring Guest Network Access
- Analysing issues in the network

3.3 Functional Requirements

In order to implement NAC, it is critical to prepare for the complete installation of your NAC solution from beginning to end. You don't want to install a NAC tool without first understanding the procedure and what information you need to acquire ahead of time. The following are the key phases in deploying a NAC solution:

- Understanding your endpoints Great plans are always the result of good research and planning. The first step in implementing a NAC solution is to survey every computer, phone, IoT device, and server that is a network endpoint. Depending

on the size of the company, this may be time-consuming on the front end, but without this information, your NAC system cannot be completely effective until it has a list of the devices that you know are accessing your network.

- Check update your Directory System As me I mentioned previously in the background section, a NAC solution must determine how much authorization each individual should have. As a result, we must select how we will handle the various roles in our organization. We can accomplish this by confirming the user IDs that can be discovered in the current directory system- typically, Microsoft's AD- and then determining how the roles should be established.

- Establishing and implementing permissions Permissions will vary depending on industry and companies' preferences, but the Principle of Least Privilege (PoPL) is a strategy that many security professionals purpose. PoPL requires that access be restricted to the most basic level feasible, which means that just the access required for an individual to do their task is granted. The integration of our NAC tool with the existing directory should be straightforward. Simply ensure that each employee is enrolled as a user in the NAC tool so that their network behavior can be tracked.

- Maintain constant updates The most important aspect of every company's network strategy is keeping things fresh and updated. As you encounter staff turnover or expansion, make changes to everything from Active Directory to the authorization policies of the NAC solution to ensure that the NAC tool remains successful.

Two of the main questions asked regarding the implementation of NAC are: What NAC features should I look for, and which NAC solution is appropriate for my company? There are a few factors to consider that might affect the NAC solution that you pick. These are as follows:

- Your present security strategy's maturity.
- The granularity of enforcement policies
- The goals for using NAC in your network
- Preference for CAPEX or OPEX
- Organization's budget

When it comes time to invest in a NAC solution and a list of questions below should be brought up to the vendor.

- What level of network visibility does the solution provide for my company?
- Is it compatible with my current infrastructure?
- How adaptable is it to changes in infrastructure in the future?
- How strictly does it enforce?
- Does it meet our compliance requirements?
- How difficult is the deployment procedure?
- Is the solution providing genuine assistance rather than merely community-based assistance?
- What is the total cost of the NAC solution?

3.4 Non-Functional Requirements

This report should act as a guide to the foundation of NAC and should be available to users as a knowledge base document. This can be achieved through an in-depth analysis of the top viable solutions available in the current market.

While there are several comparative research papers available online, there is a need for building on existing research and contributing through our computer science knowledge. The goal of this project is not to provide already existing research analysis.

The implementation approach of Open-Source NAC PacketFence should be fully functional to suit a University. The users created should be deemed as fictional to match the requirement of our project needs.

Snapshots of Virtual Machines should be created regularly in the event of a failure or system issue. This allows us to revert a VM back to the point before the failure. This should not be treated as a backup solution.

Chapter 4

Implementation Approach

This project's implementation phase concludes in two sections:

- Comparative analysis of Cisco NAC vs Microsoft NAP
- Implementation of Open-Source PacketFence NAC

4.1 Architecture

PacketFence is an open-source NAC solution that offers extensive feature which includes a captive portal for registration and remediation, centralized wired and wireless management, support for 802.1X, layer-2 isolation for problematic devices, integration with the Snort/Suricata IDS, and Nessus vulnerability scanner, PacketFence can be used to effectively secure networks of all sizes, including very large networks[11].

Features include[11]:

- Out of band (VLAN Enforcement) – When VLAN enforcement is used, PacketFence's operations are conducted outside the band, allowing the system to scale globally and have greater failure resistance.
- In-Band (Inline Enforcement) – In-band configuration of PacketFence is also an option especially if your network access points or switches are unmanageable. Additionally, compatibility with Maximum scalability and security are achieved by activating both VLAN and Inline enforcement's capacity to safeguard older systems.

- Hybrid support (Web Auth Enforcement) – If you have an accessible device that supports an external captive portal (like Cisco WLC or Aruba IAP), PacketFence can also be set up as a hotspot.

- Voice over IP (VoIP) support – VoIP, also known as IP Telephony (IPT), is widely supported by numerous switch manufacturers (Cisco, Nortel, Edge-Core, HP, LinkSys networks).

- 802.1X – A FreeRADIUS module supports both wired and wireless 802.1X.

- Wireless integration – A FreeRADIUS module enables seamless PacketFence integration with wireless networks. This enables you to secure your wired and wireless networks using the same user database and captive gateway, resulting in a consistent user experience.

- Registration – As with "captive portal" systems, PacketFence includes an optional registration procedure. In contrast to the majority of captive portal systems, PacketFence remembers users who have already registered and will instantly grant them access without further authentication. This is configurable, of course. Users cannot activate network access without first agreeing to an Acceptable User Policy, which can be set.

- Detection of abnormal network activities – Using local and remote Snort or Suricata sensors, abnormal network behaviors (computer viruses, worms, spyware, traffic barred by setup policy, etc.) can be discovered. PacketFence adds its own alerting and suppression mechanisms on top of each warning type, going beyond mere detection. Administrators are able to choose from a variety of adjustable actions for each infraction.

- Proactive vulnerability scans – Vulnerability scans using Nessus or OpenVAS can be run upon registration, on a regular, or on an ad hoc basis. PacketFence associates the vulnerability IDs of each scan engine with the violation configuration, returning content-specific web pages describing vulnerabilities the host may have.

- Isolation of problematic devices – PacketFence supports many switch vendors' isolation approaches, including VLAN isolation with VoIP support.

- Guest Access – PacketFence includes support for a specific guest VLAN. You configure your network so that the Guest VLAN only connects to the Internet, while the registration VLAN and captive portal are used to show the guest how to register for access and how his access works.

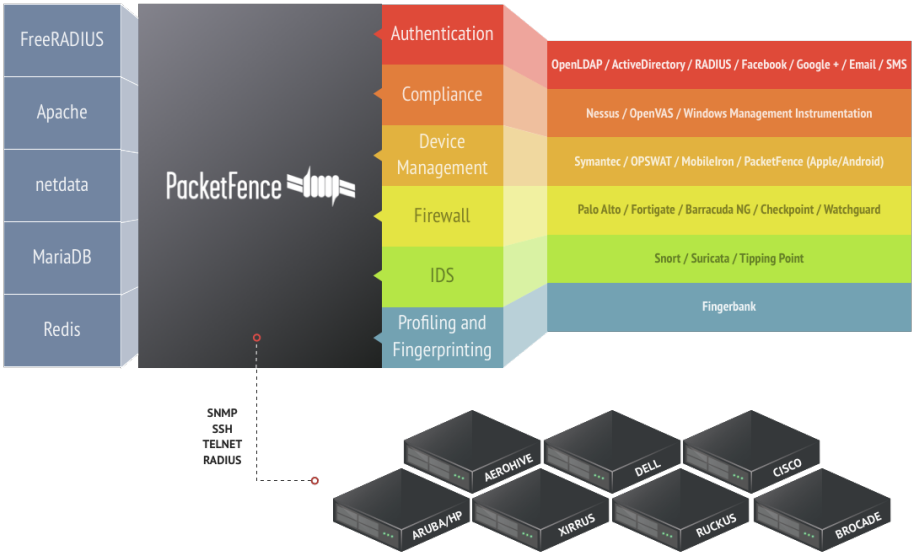

FIGURE 4.1: Overview of PacketFence Architecture[6]

1) Environment

- Network Integration

  The diagram below illustrates VLAN enforcement. Inline enforcement should be viewed as a basic flat network in which PacketFence serves as a firewall/gateway.

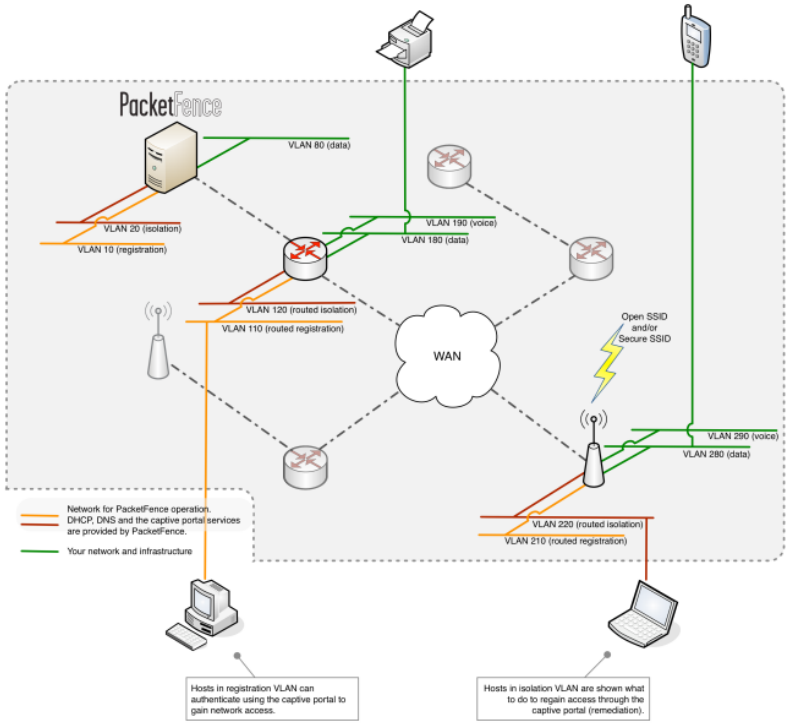

FIGURE 4.2: PacketFence network Integration[11]

- PacketFence Components

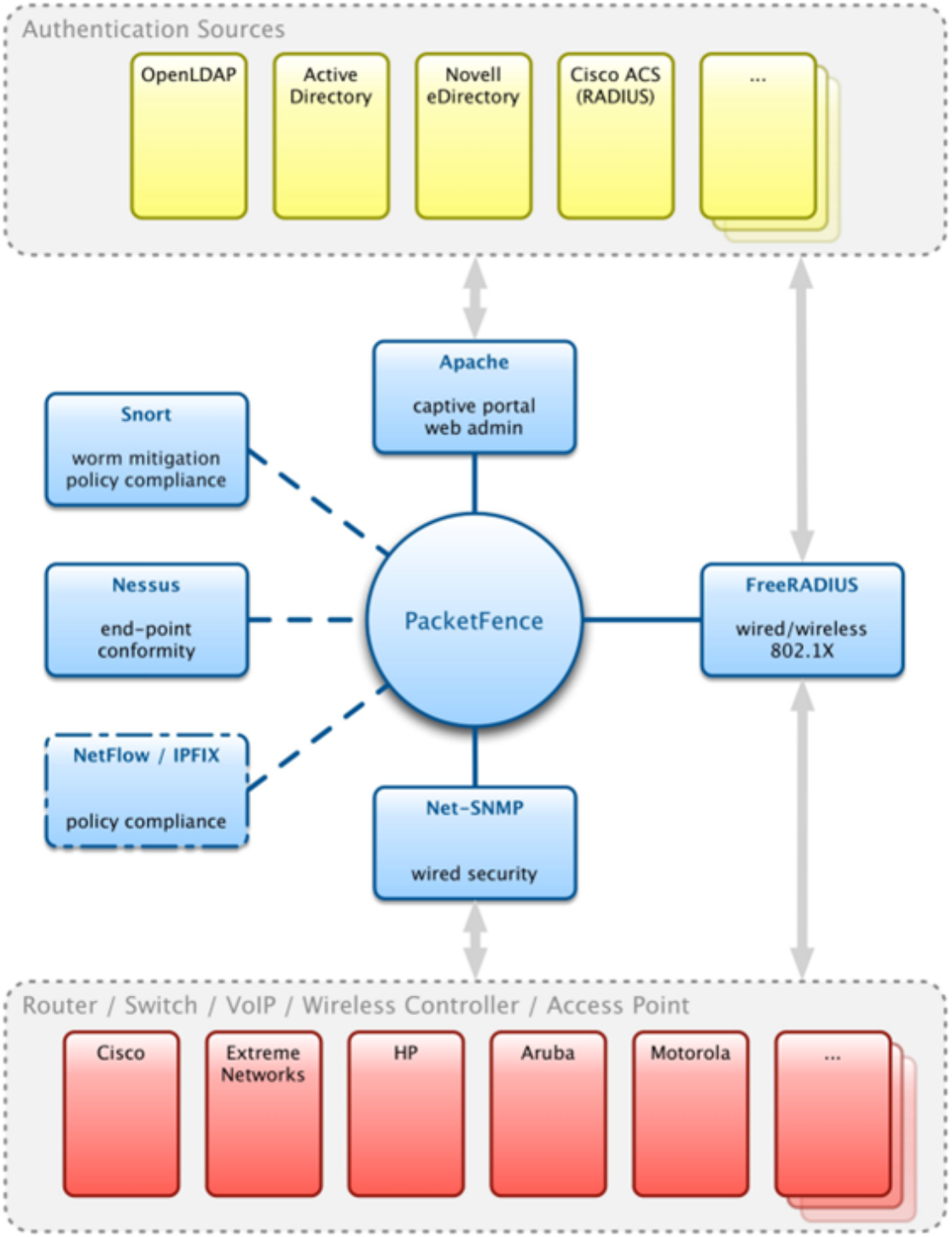

FIGURE 4.3: PacketFence Components[11]

- System Requirements

  PacketFence reuses several infrastructure components. As a result, we require the following:

  - Web server (Apache)
  - Database server (MySQL or MariaDB)

  Depending on the configuration, we may need to install other components such as:

  - DHCP server (ISC DHCP)
  - RADIUS server (FreeRADIUS)
  - NIDS (Snort/Suricata)

  In our case, we are going to assume that all of these components are operating on the same server (i.e., "localhost" or "127.0.0.1") on which PacketFence will be deployed. Installing PacketFence requires a solid grasp of the underlying components as well as GNU/Linux. The table below contains recommended components, along with their version numbers:

  - MySQL server MySQL 5.1
  - Web server Apache 2.2
  - DHCP server DHCP 4.1
  - RADIUS server FreeRADIUS 2.2.0
  - Snort Snort 2.9.1
  - Suricata Suricata 1.4.1

- Minimum Hardware Requirements

  A list of server hardware recommendations is provided below:

  - 3 GHz Intel or ADM CPU
  - 4 GB RAM
  - 100 GB disk space (RAID-1 recommended)
  - 1 Network Card (2 Recommended)

- Operating System Requirements

  On the x86-64 architecture, PacketFence supports operating systems such as Debian 11.x (Bullseye), Red Hat Enterprise Linux 8.x Server and Ubuntu 18.04[11].

2) Installation of required tools

- Host

  The ideal approach is to use two separate machines to host the Virtual Machines required to perform this work. While we may not require access to all of the VMs at the same time, we will take extra precautions and prepare another machine which will be our home Windows 10 PC. We will be utilizing our 14" Acer school laptop for the most part.

- Virtual Machine

  A CPU that supports long mode is required to operate the PacketFence virtual appliance successfully. In other words, our host must have a 64-bit capable CPU. PacketFence ZEN comes with a pre-configured virtual disk (OVF). PacketFence's ZEN (Zero Effort NAC) version enables us to quickly deploy PacketFence in our network environment. It is a virtual appliance that contains a fully installed and configured version of PacketFence. It is compatible with VMware ESX/ESXi, VMware Workstation, Microsoft Hyper-V, and other solutions.

  A separate Windows server 2016 VM will be required to set up and have access to Domain Controllers and Active Directory. This is a focal point of the project as this will assist us in the further configuration of PacketFence.

- Hypervisor

  In this installation process, VMware Workstation Pro 16 is being used as well as 16 GB of RAM devoted to the virtual machine.

3)Deployment

Once the hypervisor is installed, the .ovf template must be deployed. We want to extract the PacketFence ZEN zip file in an organized location. We then connect the ESXi host to the VMware workstation client and load the template.

4)Configuration of PackeFence environment

This section involves six key steps:

- Enforcement

  This initial and most crucial stage in the configuring procedure. This is where we will select the enforcement method: VLAN (out-of-band), INLINE (in-band), or both. The decisions made in this phase will have an impact on the next stage of configuration, where we will have to set up the various networks. Our plan is to

do it in inline enforcement for university cases. For us to build an Inline setup, we need to make sure our Virtual Machine has two network interfaces (1 for the Inline and another to go out). We will require a switch port on the management network for the PacketFence ZEN box (eth0). We will also require a switch port in the inline network for the PacketFence ZEN box (eth1) that must be configured in the access mode and in the same access VLAN as every switch port to which devices will be attached.

- Network configuration

  In this step, we want to manually configure our network interfaces. The web interface will provide a list of all network interfaces that are currently installed on the system. When the network interface is set up (DHCP or manually), an IP address and a netmask will be visible. We are able to change these, add/remove VLANs on physical interfaces, and enable/disable an interface.

- Database configuration

  In this step, we will set up the MySQL server required by PacketFence. The database and schema, as well as the appropriate user for operations, will be established. The PacketFence user account on the MySQL server is also created in this step.

- PacketFence Configuration

  In this step, we will be able to customize our PacketFence installation's general settings. This will enable us to meet the requirement for the university standard.

- Administration

  This is where we establish the administrator who will have access to the PacketFence Administration Web Interface. We should simply be able to enter the appropriate username and password, then create the user.

- Services - Confirmation

  Finally, but not least. In this step, we start the PacketFence server using the configurations from the previous steps. We should be able to proceed to the web management interface if everything has been configured properly. The status of services will allow us to see if everything is going as planned. If not, we will be able to notice which service is having issues, and the log output will assist us to figure out the errors.

4.2 Risk Assessment

- 1) Difficulty of implementation

  Implementing NAC is very ambitious and requires a certified professional to deploy and configure properly. While the installation part is easy, we may require external personnel for their guidance throughout our configuration process.

- 2) Lack of Data of the University for implementation

  Due to the limited time constraints of this implementation, gathering data such as the existing security solutions within the university is difficult. Access to this information must be requested or may not even be granted. This implementation may not particularly incorporate the intended target.

- 3) Real-time testing

  There is a lack of opportunity to test our solution in a real-time environment which may lead to us not being able to identify whether our project is deemed successful or not.

- 4) No requirements

  Implementation of NAC requires a list of requirements set out by the client to an IT professional. In our case, we must identify the requirements on our own which can prove to be challenging.

- 5) What other security solutions are implemented already that will integrate with NAC?

  NAC integrates well with other security solutions such as Identity Management, Firewalls, IDS, and Anti-Virus. We have no information on what other solutions exist within our fictional environment.

- 6) Access to Active Directory Domain

  We will require access to Windows Server for the creation of our own Active Directory for integration into PacketFence. These generally require subscription fees to purchase and use. Without access to a viable Active Directory, we may not be able to configure and deploy PacketFence properly.

- 7) Storage Requirement

  Installing and hosting three Virtual Machines requires a large amount of storage and RAM capacity. In the case that we may not be able to fulfill these requirements, the overall project completion could be at stake or could significantly delay the project.

- 8) Project Backups

  Backup should be considered an important factor for the completion of the project. There are various risks associated while executing this project such as Virtual Machines failures or Hard Disk failures. We must devise a backup plan to ensure we have project redundancy in place.

TABLE 4.1: Initial risk matrix

Frequency/ Consequence	**1-Rare**	**2-Remote**	**3-Occasional**	**4-Probable**	**5-Frequent**
4-Fatal		1			
3-Critical	7	6,8			
2-Major		4,5			
1-Minor		2,3			

4.3 Methodology

This study predominantly involves an approach to the implementation of Network Access Control which requires an in-depth understanding of cybersecurity and network. The various core modules that we have undertaken in the past such as Networking, Cybersecurity, and Internet and Networking Services will assist us with this project. Proper implementation requires an IT professional with years of experience as NAC can be quite complex depending on the configuration requirements. The project's research phase will be key to grasping the foundation knowledge of this topic. We have to evaluate and devise various factors required for the implementation phase:

- Understanding Networking and Security concepts

  This section is the foundation of our project where we must prioritize some time to gain knowledge on these concepts. This can be fulfilled in the background chapter and is highlighted in the architecture part of NAC. These areas include the Authentication Server, Policy Assessment Server, DHCP Server, Remediation Server, File Server, and Policy Enforcement and Decision Point.

- Comparative Analysis of available solutions

  This is the initial part of our implementation phase where we want to get a deeper insight into the different vendors available as a NAC solution. Our approach is to choose two viable solutions such as Cisco NAC and Microsoft NAP and an open-source NAC called PacketFence. This analysis will help us to determine various aspects of NAC such as their capabilities and limitations.

- Gathering of Required Tools and Technologies

  We must outline the required tools and technologies for the implementation of the Open-Source NAC PacketFence. This is a pinnacle part of the project where we must ensure that we have the correct compatible technologies along with their versions. As concluded previously, the main tools that we plan to utilize are VMware Workstation Pro, Ubuntu 20.04 as a host machine, and PacketFence ZEN. This implementation will also require us to learn how to set up an Active Directory for the configuration part of the PacketFence. Fortunately, we have learned to create a working Active Directory in a module called Windows Security in the second year of our degree which will help us achieve this task without many issues. Our plan is to revise and reuse the materials of the Windows Security module. This means we will also have access to Windows Server 2016 which is what will be used to create Active Directory.

- Configuration Requirements As we do not have official requirements, we are going to devise an assumption plan to suit university requirements. Time constraints could impact the completion of the following configurations:

  - Role-based Access Control
  - Adding Authentication Sources
  - Authentication Mechanisms
  - Fingerbank

- Trello Board

  Having used Trello Board previously as a project management tool for group projects, we have decided to use it to visually manage to track our project progress and workflow. We will create a new workspace to clearly lay out our project end goals and tasks with an achievable timeline. We aim to update the board each time we successfully complete a task. This will ultimately help us to be more organized and align with our goals.

4.4 Implementation Plan Schedule

- Week 1 - 2 (23rd Jan - 3rd Feb)

  Comparative analysis of Cisco NAC vs Microsoft NAP

- Week 3 - 4 (6th Feb - 17th Feb)

  Implementation approach with CISCO and Microsoft NAP

- Week 5 - 6 (20th Feb - 3rd Mar)

  Install the required tools for PacketFence implementation

- Week 7 - 8 (6th Mar - 17th Mar)

  Install Windows Server 2016 and setup Active Directory

- Week 9 - 10 (20th Mar - 31st Mar)

  Configure PacketFence via GUI

- Easter Week

  Integrate PacketFence to University Standards with various requirements Role-based Access Control

- Week 11 - 12 (17th April - 28th April)

  Improvements and Modification phase

4.5 Evaluation

4.5.1 Comparative analysis of Cisco NAC vs Microsoft NAP

This first part of the implementation section should provide an overview of the three products along with their functionality and architecture. This will give us a better understanding of what each product offers and help us to evaluate the key differences between open-source solutions versus paid ones. Further comparative analysis will be carried out on various components such as architecture, interoperability, access control, cross-platform support, policies, etc.

4.5.2 Implementation of PacketFence (Open-source NAC solution)

The second part of the implementation will conclude the demonstration of installing PacketFence as a means of NAC solution and make an effort to configure it to integrate with our own university. A sufficient level of knowledge of PacketFence will be required from the analysis phase of implementation and only then we can plan the installation. A list of assumptions and requirements must be filled out before configuring PacketFence to ensure our setup goes according to the initial plan. We will require various software to be installed on our own machines. These include a hypervisor to host our virtual machine Packetfence ZEN. Access to an Active Directory will be required as part of the configuration to enable user interaction. The goal for this implementation is laid out as follows:

- Make sure that users may only "register" a specific amount of devices per user class. For example, we would let staff have as many devices as they wish. Students are allowed to get two devices (Laptop or tablet and smartphone). Using this as a scenario, NAC allows us to "limit" the number of devices registered per user, which is very beneficial.
- Allowing devices to self-register depending on the user class rules.
- Installing Windows Server 2016 to create an Active Directory to which we can connect.
- Maintaining a reasonably adaptable "guest" network.

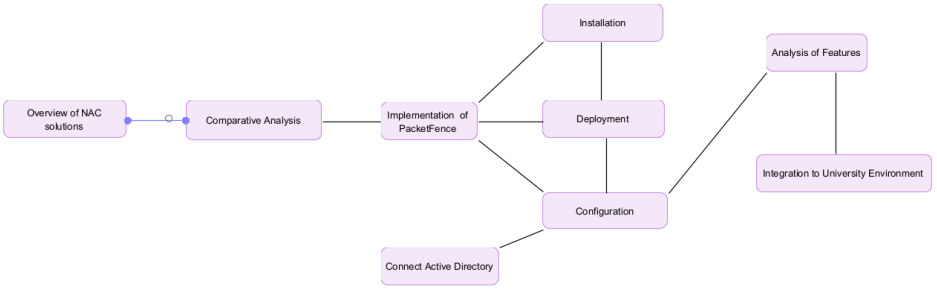

FIGURE 4.4: Roadmap to the success of the implementation

4.6 Prototype

This project is composed of an in-depth comparative analysis of NAC solutions which will also give us insight into PacketFence to help us in the installation and configuration phase. In this section, we performed a rough deployment of PacketFence ZEN Virtual Machine in VMware Workstation as a test to kickstart our implementation journey.

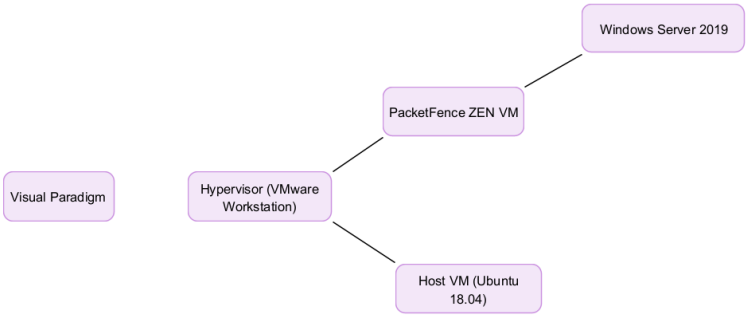

FIGURE 4.5: Roadmap of technologies being used in implementation

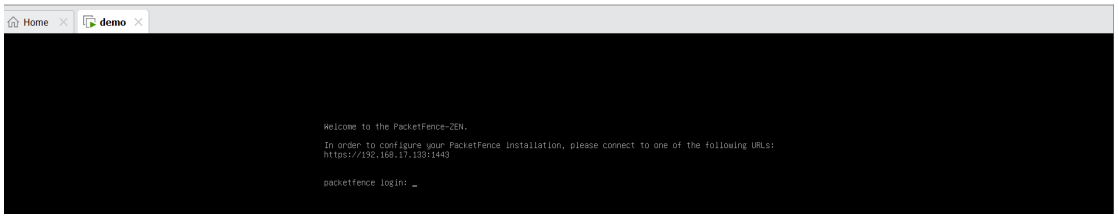

FIGURE 4.6: PacketFence Demo deployment

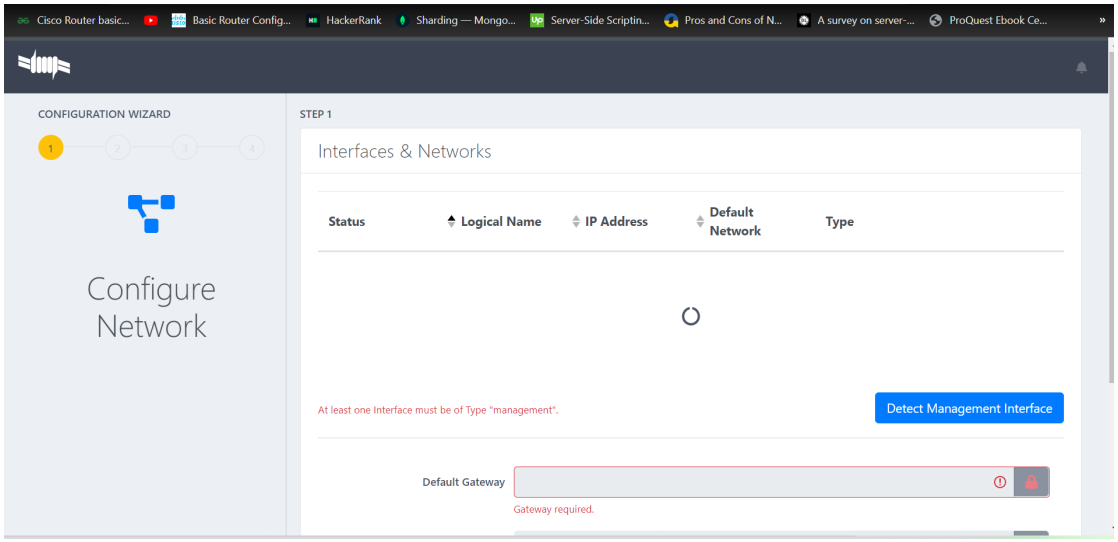

FIGURE 4.7: PacketFence Configuration Steps demo

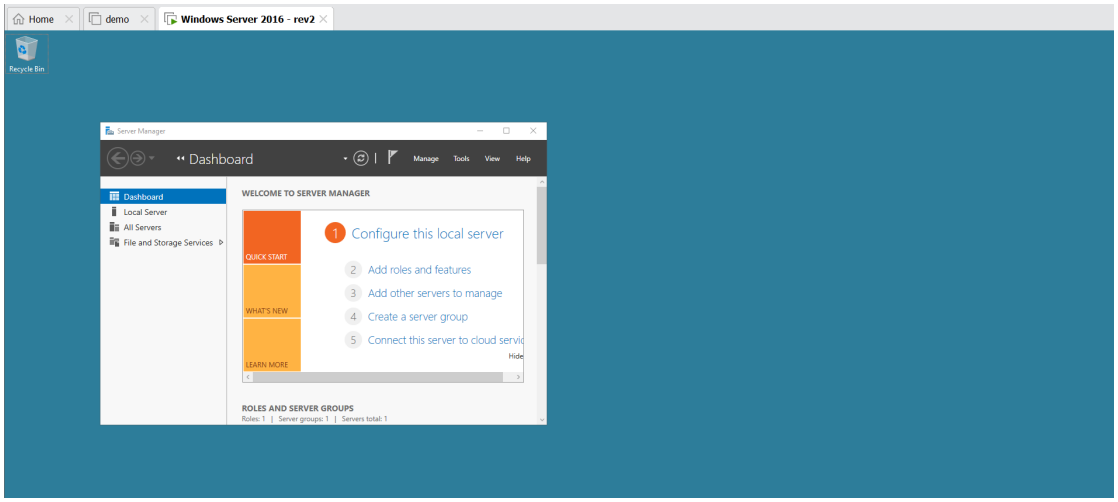

FIGURE 4.8: Active Directory demo

Chapter 5

Implementation

5.1 Actual Solution Approach

5.1.1 Open-Source NAC Implementation

The implementation section of our project involves applying Network Access Control within a small-scale network infrastructure. The primary goal was to experiment and demonstrate the features of Network Access Control through the use of an Open-Source NAC solution. This section is composed of three phases that walk through the setup and configuration process of each component of the project. The two main working components of this project are PacketFence and GNS3. Windows Server 2016 was also installed and configured in phase 2 for User Management and integration with PacketFence but due to an error encountered, an alternative solution approach was carried out. This is discussed in more detail in the Difficulties Encountered section [5.2].

5.1.1.1 Phase 1:

Installation and configuration of PacketFenceZEN

In this phase, we installed and configured PacketFence's ZEN (Zero Effort NAC) which is a condensed, pre-compiled edition of PacketFence where it enabled us to quickly deploy PacketFence in our network environment. PacketFenceZEN is a Linux-based VMware appliance that serves as a small version of the solution. Although it lacks half of the functionalities of the full version, it nevertheless provides an efficient approach to testing and exercising various NAC features [2].

PacketFence was the solution we used to illustrate the concept of Network Access Control within a virtual network environment for us to get a deeper understanding of our study. PacketFence is built on the 802.1x [4.1] port-based deployment mechanism for network access control and includes an array of security enforcement methods and policies. The integration of the PacketFence solution into an existing infrastructure can pose a challenge where the operation can take up to months to implement fully. We aimed to explore its core features such as endpoint device detection, registration, and authentication, as well as network traffic filtering and tracking.

This phase began with downloading the PacketFenceZEN version 12.1.0(OVF) file from the PacketFence website and importing it into a suitable hypervisor. We used VMware Workstation 16 Pro hypervisor to host the PacketFence VM. We dedicated 6 GB of RAM to the PacketFence Virtual Machine to sure we have adequate memory to run the PacketFence server and access its functionalities.

PacketFenceZEN VM settings:

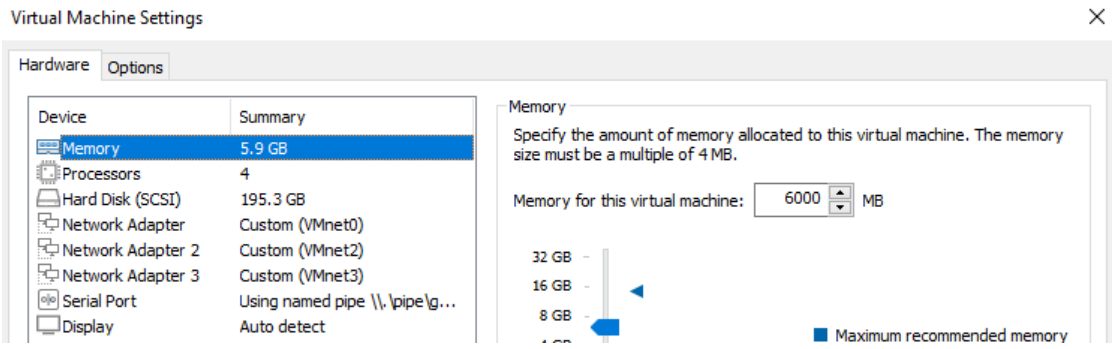

FIGURE 5.1: PacketFenceZEN VM settings

The type of enforcement method that we proceeded with for this installation is Inline enforcement. Inline enforcement is a very convenient way for conducting access control on older network devices that cannot do VLAN enforcement where compatibility issues can occur. In typical Inline enforcement, we require two switch ports: one in the Management network for the eth0 and one in the Inline network for the eth1 which must be set in access mode and in the same access VLAN as every switch port to which devices will be attached. In Inline enforcement, the PacketFence VM sits in the data path and essentially acts as a router to forward data packets to our endpoint devices in the topology which will be created in the later phases of the project implementation. For our Inline enforcement, we dedicated three network interfaces with two Inline connections allocated for eth1 and eth2. The three network adapters that we set for PacketFenceZEN VM also enabled us to have three network interfaces (One for Management and two for Inline connection) to configure in the initial steps of PacketFence installation.

The inline enforcement uses **"ipset"** [5.1.2.1], a framework within the Linux Kernel to classify nodes as registered, unregistered, or isolated based on stored IP and MAC addresses. Once access is granted, the inline enforcement method forwards traffic from the inline network through the management network interface and out to the internet.

VLAN Out-of-Band enforcement method was also attempted which presented as the more effective approach to enforcing NAC and it is one of the key components of PacketFence. This enforcement method operates through VLAN assignment as the name suggests where the PacketFence server assigns the VLAN to the endpoint devices. This VLAN could be our own creation or a unique VLAN where PacketFence displays the captive portal for remediation or authentication. VLAN assignment efficiently isolates the hosts at the OSI layer2, making it the most difficult to bypass and the most adaptable. However, this method would require a supported switch type for the creation and management of specific VLANs which prevented us to proceed with this enforcement which is outlined in the PacketFence guidebook [11]. We ran into issues with VLAN Enforcement configuration commands not being supported in the EtherSwitch(C2691).

PacketFenceZEN VM is assigned an IP address for the eth0 interface through the network's DHCP service which is used as the management interface to access the PacketFence Web-based Configurator GUI. In the case of an IP address not assigned automatically by PacketFenceZEN VM, a static IP address can also be configured by simply logging in as a root user with **"p@ck3tf3nc3"** password and navigating to the **"/etc/network/interfaces"** file.

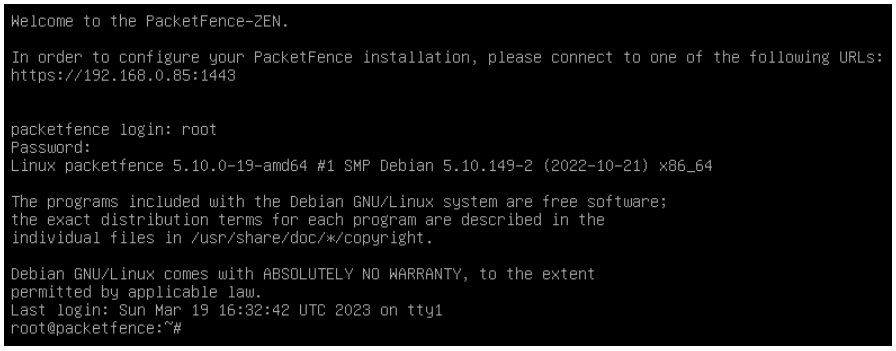

FIGURE 5.2: PacketFence Web server URL

We must also ensure that **[net.ipv4.ip forward]** is enabled on our server. IP forwarding is the capacity of an OS to receive incoming network packets on one interface, identify that they are not intended for the system, but rather for another network, and then forward them appropriately. In order to ensure this we have to edit the **/etc/sysctl.conf** file and change the IP forward value from 0 to 1 as shown in Figure 5.3 below. We then applied this rule change and made it permanent by doing the following: **sysctl -p /etc/sysctl.conf**

FIGURE 5.3: Changing net.ipv4.ipforward value to 1

Now we proceeded to access the PacketFence configurator through the Web-based GUI for further configuration of the PacketFenceZEN solution environment via the URL provided **"https://192.168.0.85:1443"** as shown in Figure 5.2 above.

Step 1: Network Configuration

This Web-based configurator GUI page displays the inventory of all network interfaces that are presently loaded on the VM. The three network interfaces that we created previously in the PacketFence VM settings will be detected and presented in the interfaces and network list. The management interface (eth0) had already been defined as we had already received the IP address **"192.168.0.85"** through the networks' DHCP service allowing us to connect with the server as well as NAT traffic from the inline network interface.

We now need to manually set static IP addresses for our eth1 and eth2 network interfaces which will be used as inline interfaces. For this particular configuration, we allocated the "192.168.1.0" subnet and set the IP address of "192.168.1.244" for eth1. The "192.168.2.0" subnet and "192.168.2.244" IP addresses were then allocated for eth2. In this stage, we would also need to define VLANs on the interfaces in the case of the VLAN enforcement method. This was not feasible for this particular demonstration due to switch compatibility issues as discussed previously hence no VLANs were created. Now, it is crucial to ensure that these three interfaces are enabled and it is also important for us to note that these adjustments take effect immediately. In our case, we enabled the required interfaces (Management and two Inline Connections) for inline enforcement as shown below in Figure 5.4.

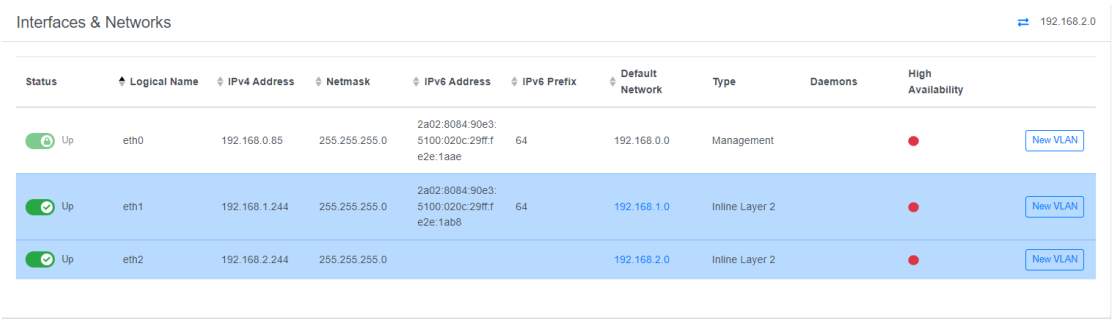

FIGURE 5.4: Network Interfaces for Inline enforcement

PacketFence also provides its own DHCP, Proxy DNS, and NAT services which it provides us with the option to enable on the Network Interface configurator page. We proceeded to enable these services in both of our inline interfaces as it makes our task easier where there will be no need to provide these services to our endpoint devices from an external source. The DHCP service automatically provides IP addresses for our endpoint devices which saves us the effort of manually assigning static IPs for each of the devices. Proxy DNS service ensures that DNS queries are forwarded in response to the DNS server and DNS client accordingly. Lastly, the NAT service ensures that we have internet connectivity through the inline interface to the endpoint devices once the registration process has been completed.

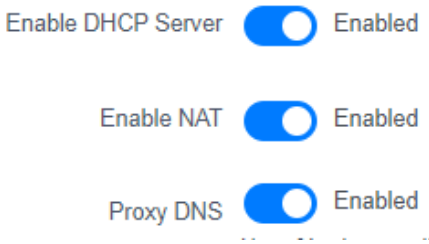

FIGURE 5.5: Enabling PacketFence DHCP, DNS and NAT services

Step 2: Database Configuration

In this stage, we will set up the MySQL server database account required by PacketFence. This is used to store registered devices, users, and any other configurations made by administrators. This step is very simple as PacketFenceZEN comes with a pre-configured MySQL database and established schema, as well as the appropriate user for operations. Once we start the SQL service, it will automatically generate and assign a password for the root account of the MySQL database, and a **'pf'** user will be created.

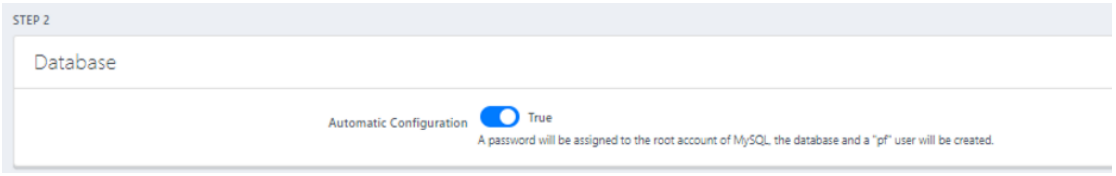

FIGURE 5.6: PacketFence Database Configuration

Step 3: System Configuration

This step revolves around the general PacketFence installation options such as specifying the domain, hostname, etc. We can simply skip this and the alerting section to leave the default values and proceed to other configurations as these options are not requirements for our setup. The only action we need to take here is to create an Administrator password which will be used to login into the Web-based Administrator GUI of PacketFence. Once we receive the success message, we can move to the next step.

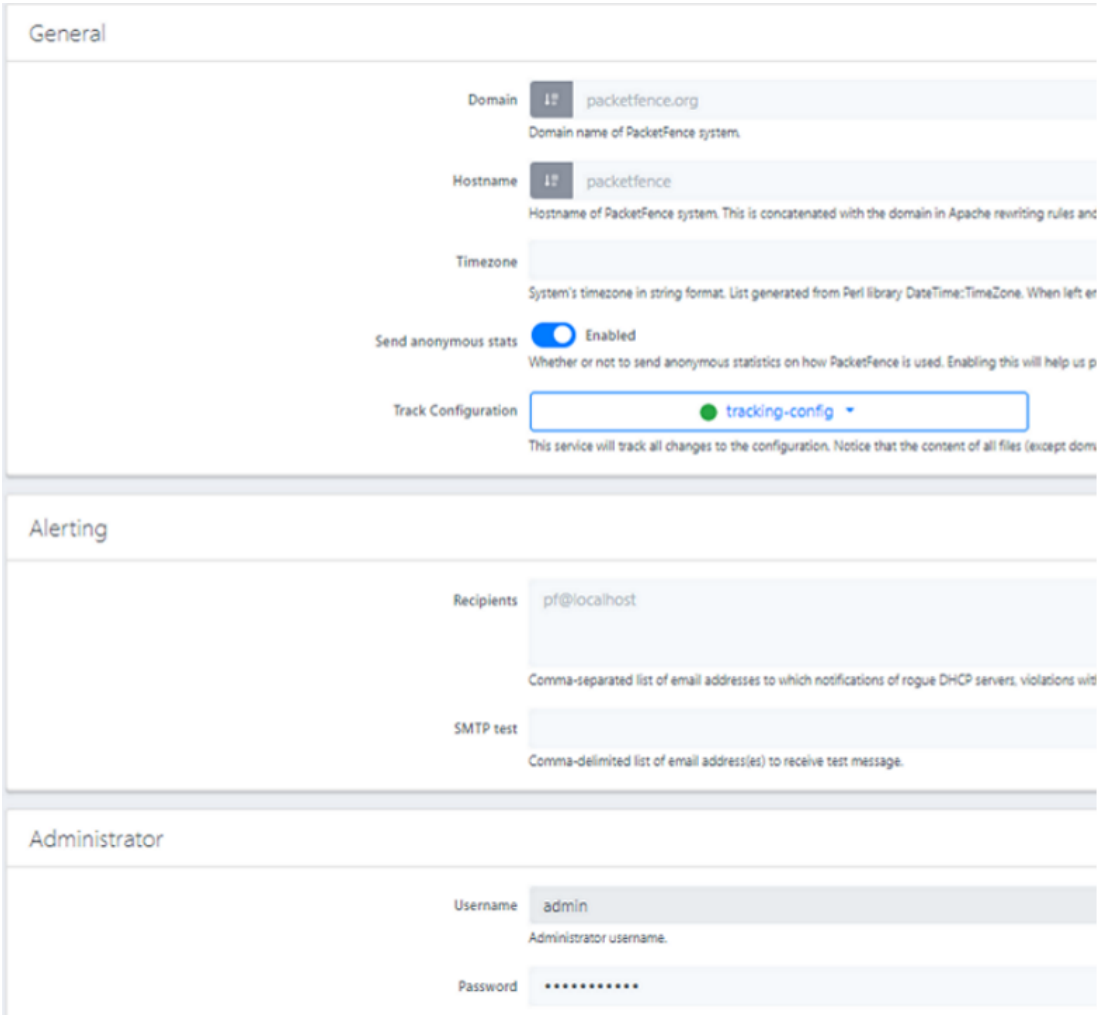

FIGURE 5.7: PacketFence System Configuration

Step 4: Configuration Confirmation

PacketFence will outline the login credentials of various accounts created from the previous configuration steps as shown below in Figure 5.8 below, we ensured that we captured these credentials in a secure place as we will require them in the future. Once the PacketFence server configuration process has been completed, we are now able to start the PacketFence. If everything is correctly configured, PacketFence will notify us with a success message and we will be prompted to a Web-based administrator login GUI.

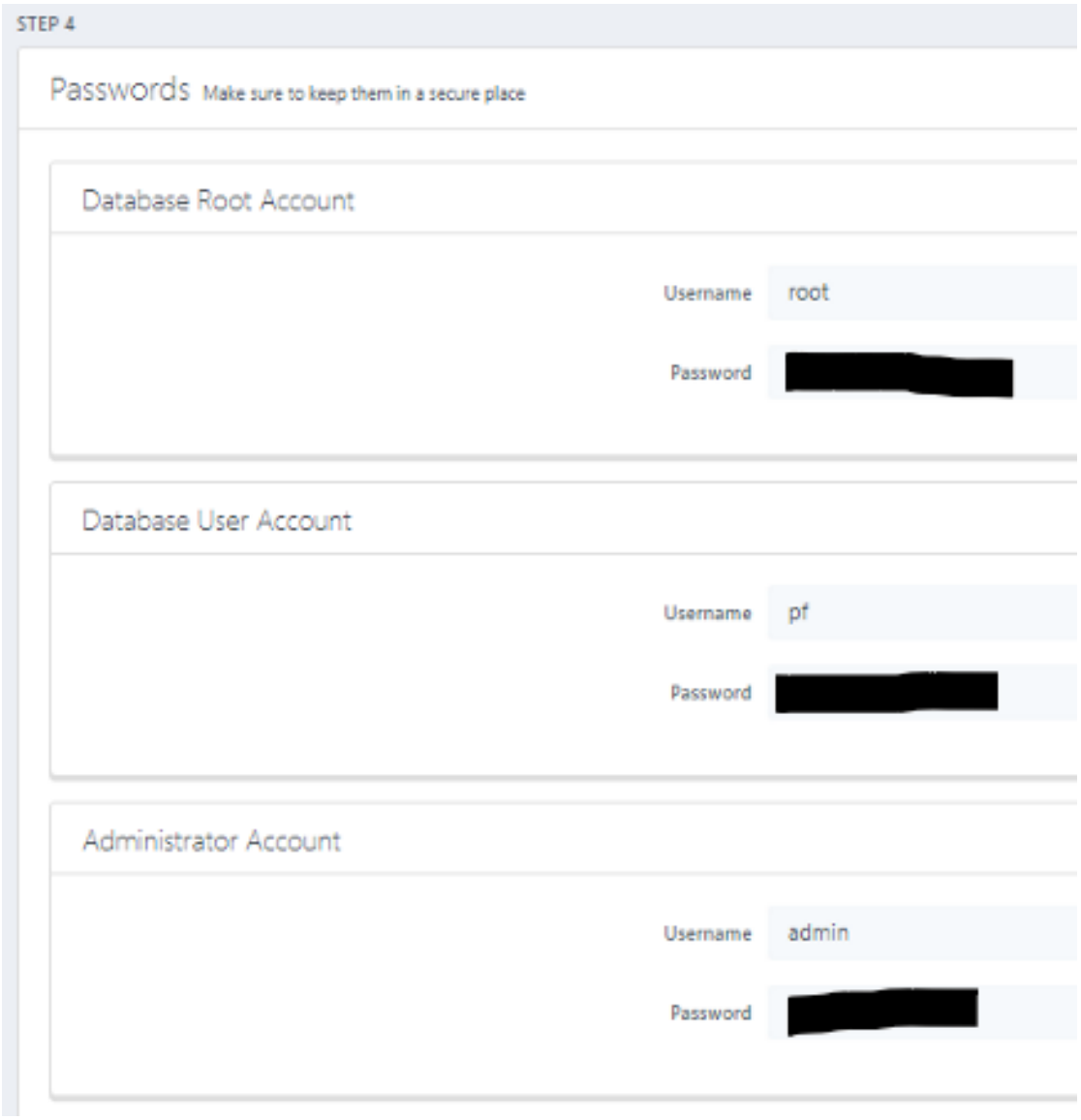

FIGURE 5.8: PacketFence Confirmation

Step 5: Log in as Admin in and Monitoring PacketFence environment

In the Web-based Administrator login GUI, we logged in with the credentials of admin and the password that we created in the configuration setup process previously to access the PacketFence configurator. Once this has been achieved, we are presented with a

status dashboard that provides us with key information regarding our PacketFence environment and server. It outlines the overall metrics of our PacketFence system where we can actively monitor details such as System load, Disk I/O, Disk Space Usage, and System RAM. The system load can fluctuate depending on the level of activities performed within the PacketFence environment. For example, when nodes/devices are added to the MySQL database, the system load rises up. The server's memory was an important metric for us that needed to be monitored which refers to the system's overall memory usage and we observed that the RAM consumption is almost at its limit with only **".12"** GB free RAM available. We had allocated only 6 GB of RAM previously in our PacketFenceZEN VM due to our computing resource constraints as we also had to run other components for our implementation. This RAM allocation for PacketFenceZEN VM served as a benchmark to run the system with minimal resources while freeing up some RAM allocation for other components.

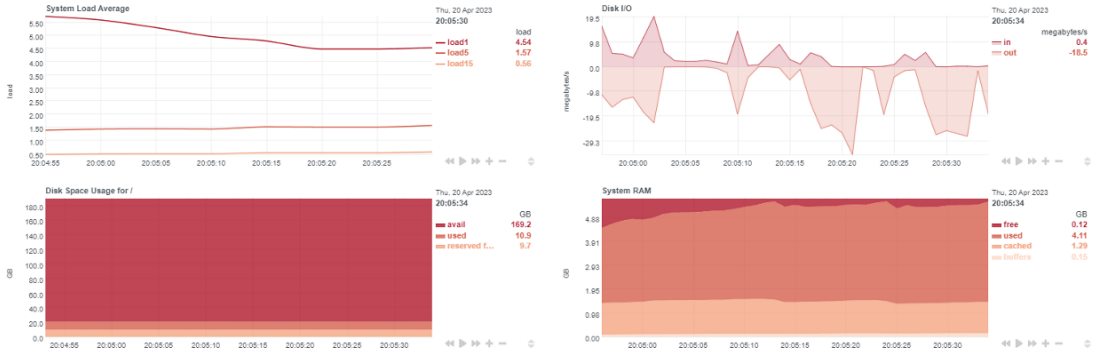

FIGURE 5.9: PacketFence Server Metrics

Another feature that we found essential was to monitor and enable the services provided by the PacketFence environment before any access control measures are activated. We are able to manage specific services in terms of their present status through the service dashboard where we can start and stop services when necessary. One critical service that we need to ensure is running at all times is the DHCP service provided by PacketFence to guarantee that the endpoint devices requesting access are assigned an IP address. This is required so that the nodes are dynamically assigned an IP address from the inline network subnet while connected to the switch on the side of the inline network. Restarting the services can be useful for refreshing the server and ensuring that all required services are operating as normal. One key example was when refreshing the **"iptables"** service was useful when the server was failing to update appropriately when a device had been granted access and registered.

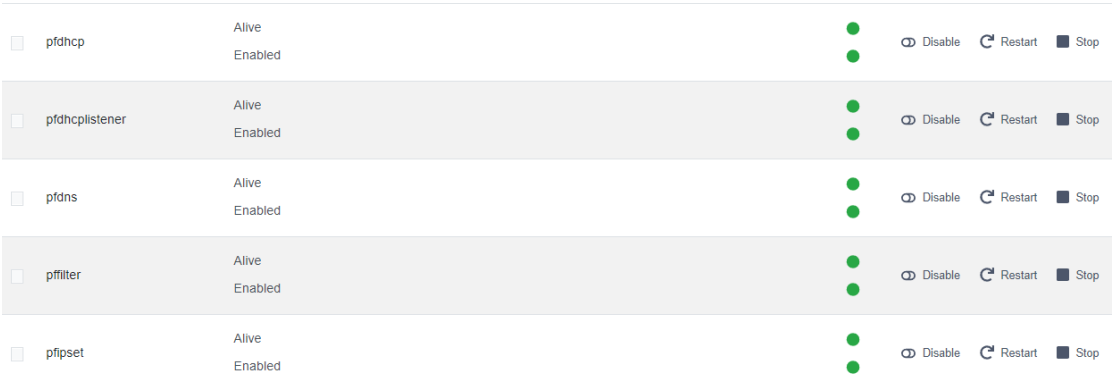

FIGURE 5.10: PacketFence service monitoring

The PacketFence Web-based Administrator GUI also includes the network setup page which serves as a critical feature that allows for the modification of network interfaces. This functionality proved to be an important factor as adjustments of network interfaces were required throughout the implementation phase, especially in the System Testing section [6.2] of the project. It makes it possible to add, delete and edit network interfaces as well as specify VLANs within the interfaces from this step. PacketFence excels in this aspect of configuration as it truly gives entire control over a given network through a more user-friendly GUI, leaving very little room for errors in regards to the network connection.

The last feature that we found useful to monitor was the Admin API Audit logs table which Administrators can utilize to analyze audit events of the PacketFence environment to maintain compliance, mitigate against unauthorized access, and as well as audit suspicious behavior within the network.

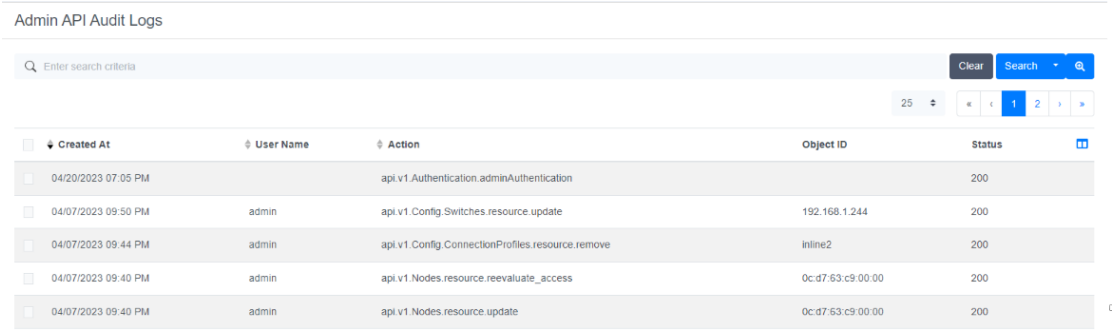

FIGURE 5.11: PacketFence Audit Logs

5.1.1.2 Phase 2:

Active Directory Creation

In this phase, we wanted to create an Active Directory to integrate it into the PacketFence server for user management. We went through the process of creating AD along with DNS and DHCP server in Windows Server 2016 which we acquired through the academic resources but we faced issues while integrating our AD domain into the PacketFence server which we have discussed in more detail in the Difficulties Encountered section [5.2]. Initially, our plan was to use Active Directory for centralized management along with providing DNS and DHCP services within our network Topology. However, we decided to take an alternative route of creating our users and roles along with enabling DHCP and DNS servers within PacketFence. This emphasizes the functionalities of the PacketFence NAC solution and further highlights the importance and features provided by this solution. This gave us more opportunities to showcase features that can be utilized within the PacketFence NAC solution. Nevertheless, this body of work can be used as a reference for future work and improvements within network infrastructures.

We deployed the Virtual Machine to the VMware Workstation 16 Pro and dedicated 4 GB of RAM for the VM to run smoothly and kept the other settings default as seen in Figure 5.12 below.

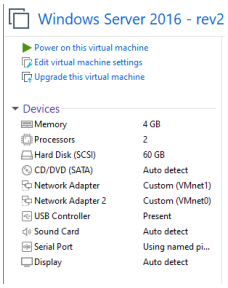

FIGURE 5.12: Windows Server 2016 VM settings

Step 1: The first thing that we did after logging into the Windows Server VM was changed our computer name to suit our project as shown in Figure 5.13 below.

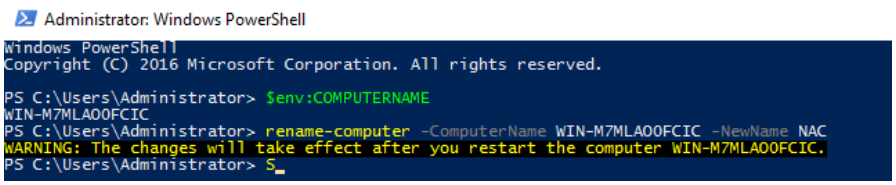

FIGURE 5.13: Windows Server computer name change

Step 2: We choose the selection type Role-based Active Directory installation which will authenticate a user and then associate rights with this user using configuration information saved on an Active Directory server.

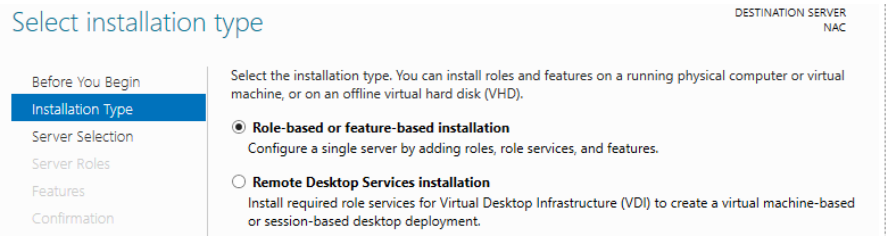

FIGURE 5.14: Role-based Active Directory

Step 3: We selected our destination server NAC listed in the server pool.

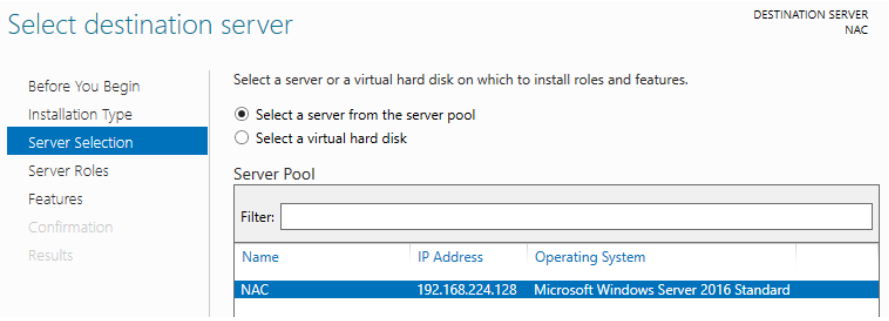

FIGURE 5.15: Destination Server

Step 4: We selected Active Directory Domain Services and DNS server creation.

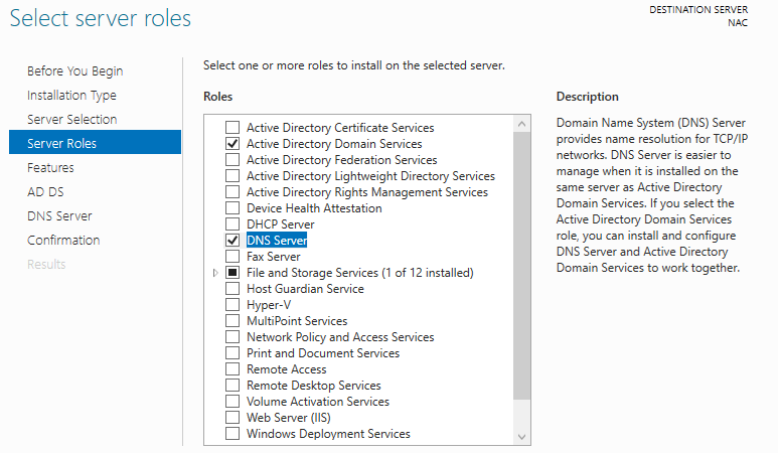

FIGURE 5.16: AD Domain and DNS server creation

Step 5: After the installation finished, We promoted the server that we created to a domain controller.

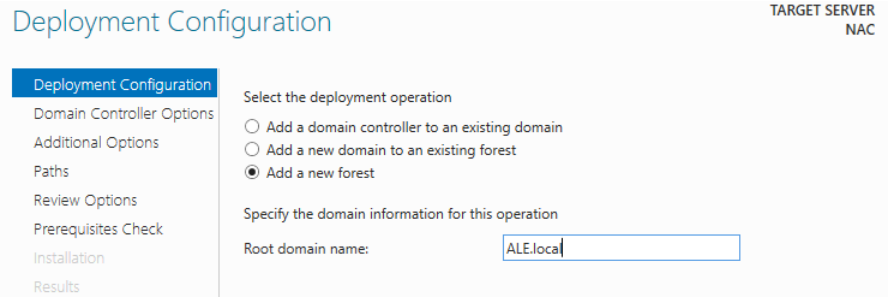

FIGURE 5.17: Promoting Server to the domain controller

Step 6: We selected our Operating system as the functional level of the new forest and root domain.

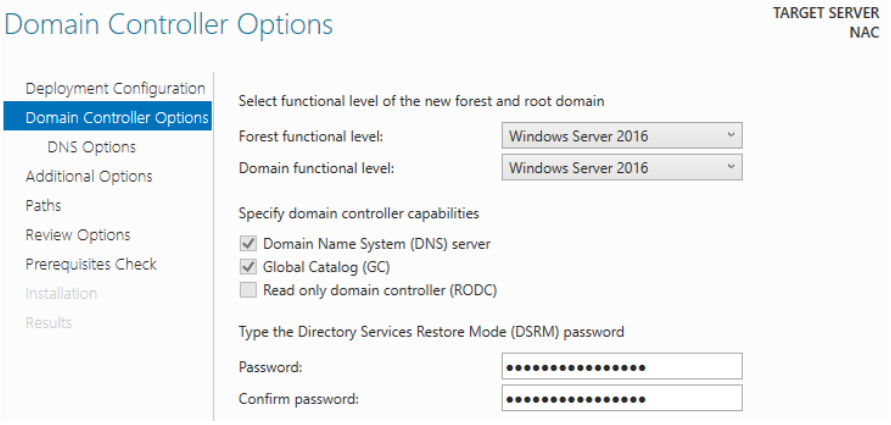

FIGURE 5.18: Selecting OS as New Forest and Root domain

Step 7: After the Active Directory configuration was completed, We went ahead and created our sample users to replicate a University AD. Due to the time constraint, we only created one user per group. Our approach to creating groups was that we wanted to have a basic group that a University would have. For example, Admin, Student, Staff, and Computer groups. Before we create groups and users, we want to create a dedicated OU (Organizational Unit) for the PacketFence server. An OU is a container in an Active Directory domain that can store various data from the same AD domain, such as other containers, groups, users, and computer profiles. Figure 5.19 below shows the OU (PacketFence) that we created along with the users, groups, and computers.

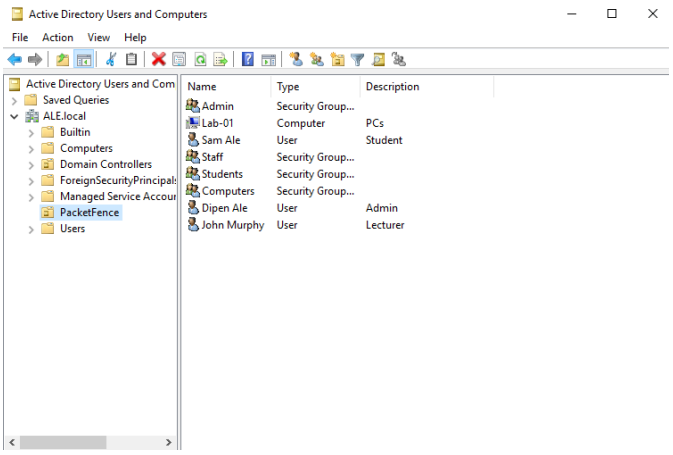

FIGURE 5.19: Active Directory sample users and group

Step 8: We also set up a DHCP server as essential for network IP address management which was the initial plan from the research phase to utilize Windows Server as a way to manage users and distribute the DHCP server. This will allow us to automatically allocate IP addresses to network clients which will lower the amount of manual IP configuration needed. It will enable the clients to communicate with one another and have access to network resources.

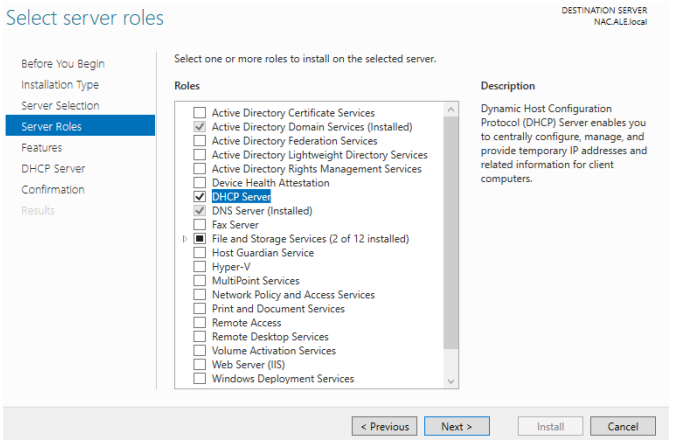

FIGURE 5.20: DHCP server setup

5.1.1.3 Phase 3:

This last phase of our implementation section involves integrating the PacketFenceZEN NAC solution into GNS3 (Graphical Network Simulator) software to emulate a simple network design and demonstrate some key features of PacketFence. GNS3 is an open-source tool that allows for the possibility of creating such a network without the use of dedicated network devices such as routers and switches.[12] GNS3 comes with an easy-to-use graphical user interface for designing and configuring virtual networks. It

also offers GNS3 Virtual Machine which is a very lightweight and robust method of designing networks compared to other emulators. For our project, we installed the GNS3 GUI application along with GNS3 VM to create a small-scale network topology. This was a relatively new technology for us so we did not have much experience with it but with some time allocation towards learning this tool, we are able to navigate through and build our network topology.

We allocated 2 GB RAM for our GNS3 VM for us to run some components of our topology which we will explore in the next page.

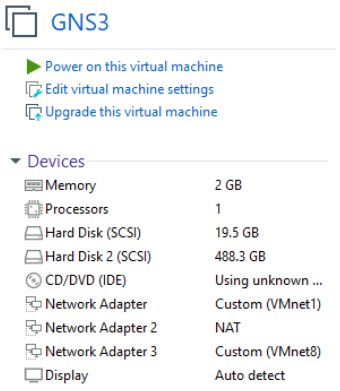

FIGURE 5.21: GSN3 VM settings

It was important to have a structured plan before creating any type of network topology to sure that we evaluate the components that we need to build our topology. We used the Visual Paradigm tool to create a rough network topology as shown below in Figure 5.22.

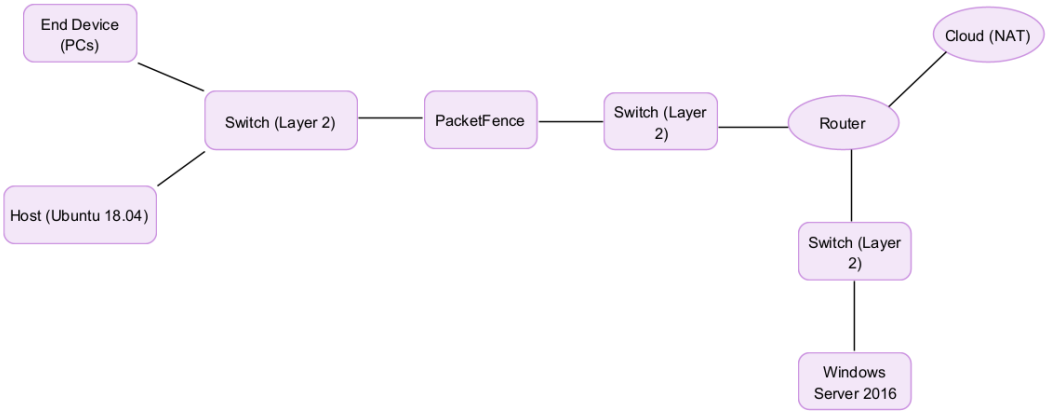

FIGURE 5.22: Network Topology Plan

Our Final GNS3 Network Topology:

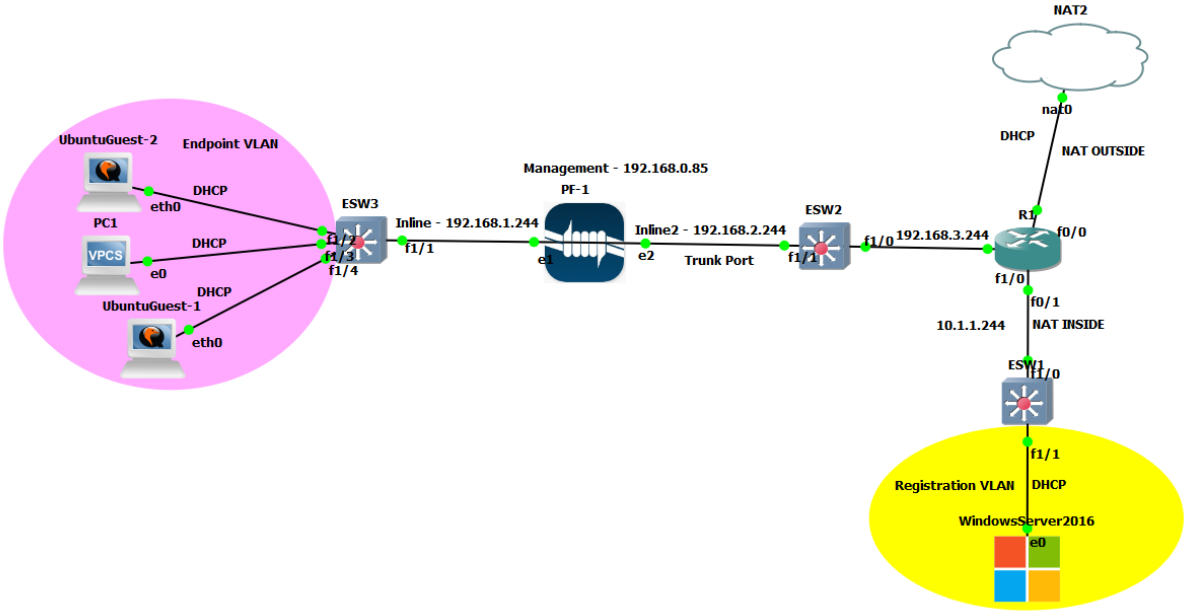

FIGURE 5.23: Network Topology Infrastructure

Discussion on the components of our topology:

- **NAT Node** - This component is available within the GNS3 GUI interface by default and it is used to connect a topology to the internet via NAT as the name suggests. It is handy whenever we need to install something from the internet, such as packages, and is much simpler than using other nodes like Cloud. We utilized this component mainly for the Windows Server VM as a demonstration of how internet access can be distributed within a GNS3 network topology using this component. Even though there is no functionality provided by the Windows Server VM component, we used it for verification purposes of internet access.

- **Router and Switches** - We needed to obtain IOS images for Layer3 (router) and Layer2 (switch) networking devices for our virtual topology where GNS3 will allow us to emulate Cisco devices and run the IOS images. Fortunately, we were able to obtain IOS images for a router (C3725) and Etherswitch (C2691) that we imported and used in our network topology. GNS3 allows us to simulate IOS in a variety of methods. One of the methods is using Dynamips which is an emulator technology used by GNS3 to emulate Cisco routers and simple switching via the Etherswitch module. It emulates older Cisco devices, such as 3725, and employs genuine Cisco IOS images.

- **Appliances** - GNS3 offers a wide range of appliances that we can import and use in our network topology such as our Ubuntu Desktop Client which we used

as an endpoint device. These appliances have been fully tested so there are little or no issues utilizing them whereas using other external images may need more configuration and debugging to get it to work. Many applications within the GNS3 marketplace are open-source where images can be obtained for free however some do not, For example in our case, for us to use vendors such as Cisco and Microsoft as appliances will require us to obtain IOS images directly from their websites with fees which was not feasible in our case.

- **VMware Virtual Machines** - GNS3 allows us to import VMware Virtual Machines to integrate into our network topology. We simply needed to navigate to the edit-preferences section within GNS3 and import our required VMs. For our topology, we imported PacketFenceZEN and Windows Server 2016 VMs which we configured in the previous phases. It is not optimal for us to run these VMs in GNS3 externally from VMware Workstation hypervisor as it requires a lot of computing resources such as RAM and CPU but due to some constraints that we discussed previously in the appliances section, we had to come up with an alternative solution.

Router(R1) configurations:

This part of the work could be deemed as external work that does not relate to the exploration of the PacketFence NAC solution. However, we wanted to configure our topology to have internet access through a NAT node to showcase alternative methods of providing various services within our network topology. NAT essentially translates internet network source IP addresses into publicly routable IP addresses so that there is internet connectivity. We configured the interface F0/0 connected to the NAT node as DHCP, set it as outside NAT, and the other interfaces as Static with NAT inside as seen in Figure 5.24 below.

```
interface FastEthernet0/0
 ip address dhcp
 ip nat outside
 ip virtual-reassembly
 duplex auto
 speed auto
!
interface FastEthernet0/1
 description trunk link to AD
 ip address 10.1.1.244 255.255.255.0
 ip nat inside
 ip virtual-reassembly
 duplex auto
 speed auto
interface FastEthernet1/0
 description trunk link to PF
 ip address 192.168.3.244 255.255.255.0
 ip nat inside
 ip virtual-reassembly
 duplex auto
 speed auto
!
```

FIGURE 5.24: IP configurations with NAT

Router-on-Stick Inter-VLAN Routing between R1 to ESW1, ESW2 and ESW3:

We created trunk links within our topology that were required to enable forwarding traffic within and between various VLANs. We created a total of four VLANs in all three of our switches to accommodate each segment of our topology. Furthermore, it is crucial to note that every port connecting to the endpoint devices must be assigned to the same VLAN as the inline interface port(access point). For example, we created a separate VLAN **"endpoint"** and placed all of the ports connected to the endpoint devices and the port connecting to the PacketFence to ensure communication between endpoint devices and PacketFence.

```
ESW3#sh vlan-switch br

VLAN Name                             Status    Ports
---- -------------------------------- --------- -------------------------------
1    default                          active    Fa1/0, Fa1/6, Fa1/7, Fa1/8
                                                Fa1/9, Fa1/10, Fa1/11, Fa1/12
                                                Fa1/13, Fa1/14, Fa1/15
10   endpoint                         active    Fa1/1, Fa1/2, Fa1/3, Fa1/4
                                                Fa1/5
20   registration                     active
30   isolation                        active
40   management                       active
1002 fddi-default                     act/unsup
1003 token-ring-default               act/unsup
1004 fddinet-default                  act/unsup
1005 trnet-default                    act/unsup
ESW3#
```

FIGURE 5.25: ESW2 VLANs Configurations

We then configured our trunking ports to the PacketFence and Windows Server VM interfaces as shown in Figure 5.26 below.

```
ESW2#sh int trunk

Port      Mode         Encapsulation  Status        Native vlan
Fa1/0     on           802.1q         trunking      1

Port      Vlans allowed on trunk
Fa1/0     1-4094

Port      Vlans allowed and active in management domain
Fa1/0     1,10,20,30,40

Port      Vlans in spanning tree forwarding state and not pruned
Fa1/0     1,10,20,30,40
ESW2#
```

FIGURE 5.26: ESW2 Trunking configuration showcase

The router-on-a-stick approach will require us to create a subinterface for each VLAN to be routed. This can be created by specifying the physical interface followed by a

period and a subinterface number, this is not required but a good practice to match the subinterface number to the VLAN number as shown below in Figure 5.27 below.

```
R1#sh ip int br | include up
FastEthernet0/0            192.168.122.223 YES DHCP   up                    up
FastEthernet0/1            10.1.1.244      YES manual up                    up
FastEthernet0/1.10         192.168.11.1    YES NVRAM  up                    up
FastEthernet0/1.20         192.168.22.1    YES NVRAM  up                    up
FastEthernet0/1.30         192.168.33.1    YES NVRAM  up                    up
FastEthernet0/1.40         192.168.44.1    YES NVRAM  up                    up
FastEthernet1/0            192.168.3.244   YES manual up                    up
FastEthernet1/0.10         192.168.10.1    YES NVRAM  up                    up
FastEthernet1/0.20         192.168.20.1    YES NVRAM  up                    up
FastEthernet1/0.30         192.168.30.1    YES NVRAM  up                    up
FastEthernet1/0.40         192.168.40.1    YES NVRAM  up                    up
R1#
*Mar  1 00:03:20.839: %SYS-5-CONFIG_I: Configured from console by console
R1#sh int f1/0.10
FastEthernet1/0.10 is up, line protocol is up
  Hardware is AmdFE, address is c201.06fd.0010 (bia c201.06fd.0010)
  Description: default gateway for vlan 10
  Internet address is 192.168.10.1/24
  MTU 1500 bytes, BW 100000 Kbit/sec, DLY 100 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation 802.1Q Virtual LAN, Vlan ID  10.
  ARP type: ARPA, ARP Timeout 04:00:00
  Last clearing of "show interface" counters never
R1#
```

FIGURE 5.27: Subinterface creation and details verification on R1

PacketFence provides an option to enable DHCP and DNS servers while configuring network interfaces in the setup process hence we leveraged this feature which will lease IP and DNS addresses to all of our endpoint devices. This eliminates having to configure devices manually with static IP and saves a lot of time in a real production network. We also wanted to create a DHCP pool to showcase and explore an alternative method of enforcing the DHCP server in our router so that it can lease IP addresses to the Management subnet (Active Directory Server) automatically which can be seen in Figure 5.28 below.

```
ip dhcp pool NAC
   network 10.1.1.0 255.255.255.0
   default-router 10.1.1.244
```

FIGURE 5.28: DHCP Pool with DNS server for Management Subnet in R1

5.1.2 NAC Vendors Technical Overview

After configuring and integrating the PacketFence NAC solution into a virtual Network topology, we wanted to evaluate how this solution compares to other vendors' solutions that are currently in the market. For this comparative analysis, we have selected the two most popular solutions such as Cisco NAC and Microsoft NAP. A large part of our implementation section consists of examining PacketFence features which is an open-source NAC so it was essential that we included this open-source tool as part of the comparative analysis.

5.1.2.1 PacketFence

PacketFence is a reliable, free, and Open Source NAC system. PacketFence can be used to effectively secure networks ranging from small to very large heterogeneous networks, thanks to an impressive feature set that includes a captive portal for registration and remediation, centralized wired and wireless management, 802.1X support, layer-2 isolation of problematic devices, and integration with the Snort IDS and the Nessus vulnerability scanner. Some of the key advanced features of PacketFence Include: [13]

- **Role-based access and Scalable VLAN Management** - The solution is based on the idea of network segregation via VLAN assignment which means that our existing VLAN configuration on our topology can be retained and only two new VLANs need to be introduced (Registration and Isolation VLAN). For example, with standard PacketFence configuration, a VLAN or a role can be granted to our printers or PCs based on what devices they are attached to. This enables us to simply have VLANs that can be associated with per building and per device.
- **BYOD (Bring Your Own Device)** - Most organizations now have a large number of consultants from different businesses on-site who require Internet connectivity to perform their work. In most instances, entry to the corporate network is granted with little to no audit of the person or device. Furthermore, they are rarely needed to have access to internal company infrastructure; this is done to reduce the administrative load (per-port VLAN management). PacketFence includes support for guest VLAN where if we use a guest VLAN, we can set up our network so that the guest VLAN only connects to the Internet, and the registration VLAN and captive portal are the components used to demonstrate to the guest how to register for access and how their access functions. This of course varies from organization to organization but some of the ways of registering guests are as follows:

  - Self-registration
  - Registration guest manually
  - Temporary password of the day
  - Email validation
  - Mobile Phone
  - Sponsoring entry for the guests

- **Integration of Firewalls** - PacketFence supports Single-Sign-On with a variety of firewall vendors. Upon connection on the wired or wireless network, PacketFence can automatically change the IP/user association on firewalls, allowing the implementation of per-user or per-group filtering rules if necessary. Some of the firewall solutions that are supported by PacketFence are Barracuda, FortiGate, CheckPoint, and many others.

- **Integration of Microsoft Active Directory** - PacketFence works seamlessly with MS Active Directory. A PacketFence server can be connected to numerous Active Directory domains without the need to create trust between them. Furthermore, PacketFence is perfectly compatible with Windows Management Instrumentation (WMI) where based on the WMI scans, endpoints can be registered. It can also conduct WMI checks during the registration process, at predetermined periods, or on every link to a wired or WiFi network.

PacketFence offers various enforcement modes to suit the requirement of the clients. The enforcement mode is a methodology used to enforce device registration and subsequent access to our network. The following enforcement modes are available with PacketFence: [11]

- **Inline Enforcement** - Unlike other NAC solutions, unmanageable devices like entry-level consumer switches or access points can be supported through this enforcement mode. In other words, PacketFence acts as the inline network's gateway, using NAT or routing traffic to the Internet through IPTables/IPSet (or to another section of the network). In terms of device configuration, no specific setup is required for unmanageable devices. You only need to verify that the device is "talking" on the internal VLAN. As the gateway for this VLAN, all data will be routed through PacketFence at this time. The access control is based primarily on IPTables/IPSet. When a user is not enrolled and joins in the inline VLAN, PacketFence assigns the user an IP address. At this stage, the user will be designated an unregistered in the ipset session, and all Web traffic will be rerouted to

the captive portal and other traffic will be prohibited. Due to the structure of this enforcement method, there are several limitations that should be addressed:

  - Everyone connected through an inline connection is on the same Layer 2 network.
  - Every approved user's packet is routed through the PacketFence server, which significantly increases the server's load.
  - Every approved user packet is routed through the PacketFence server, which serves as a single point of failure for Internet connectivity.
  - Ipset can hold up to 65536 entries, so an inline network class larger than a class B is not feasible.

- **Out-of-Band Enforcement** - This enforcement method operates through VLAN assignment which simply means the PacketFence server assigns the VLAN to the devices. This VLAN could be our own creation or a unique VLAN where PacketFence displays the captive portal for remediation or authentication. VLAN assignment efficiently isolates the hosts at the OSI Layer2 level, making it the most difficult to bypass and the most adaptable. We will discuss two different techniques for VLAN Assignment:

  - Wired: 802.1X and MAC Authentication - 802.1X supports port-based authentication, which includes interactions between a supplicant, authenticator, and authentication server. The supplicant is typically software on a client device (laptop), the authenticator could be a wired Ethernet switch or wireless, and the authentication server as a RADIUS server.
  - Wireless: 802.1X and MAC Authentication - Wireless 802.1X functions similarly to cable 802.1X, MAC authentication is practically the same as the wired version. The difference is that 802.1X is used to configure the security keys for protected transmission (WPA2-Enterprise), whereas MAC authentication is only used to validate a MAC on the wireless network. On wireless networks, the standard PacketFence configuration calls for two SSIDs (Open one and secure one). The open one is used to assist users in correctly configuring the secure one, and it needs authorization via the captive portal.

- **Hybrid Enforcement** - RADIUS could not be activated for inline enforcement mode in the earlier versions of PacketFence. All devices that enable 802.1X or MAC authentication can now operate with the new hybrid mode. Configuration of this enforcement can consist of both enablement inline enforcement and setting up access points to use VLAN assignment techniques.

- **RADIUS Enforcement** - The basic idea behind this enforcement is to avoid using PacketFence's registration, isolation, and portal features. Everything in this enforcement is only for RADIUS integration.

- **DNS Enforcement** - This enforcement enables us to manage the device's network access by using the pfdns service. The PacketFence server handles the DHCP and DNS. The PacketFence DHCP server will use the IP address of the network devices as the gateway and the IP address of the PacketFence DNS server to determine names. Routing is delivered by another piece of networking equipment (core switch, firewall, router, etc.). This enforcement option can be bypassed by the device using an alternative DNS server or its own DNS cache. The bypass through the DNS server can be mitigated by using an ACL on the routing equipment, while the DNS cache can be mitigated by merging DNS compliance with Single-Sign-On on the network equipment.

5.1.2.2 Cisco NAC

As a significant networking and data communication company, Cisco has not been left behind in the NAC world. Cisco has its own version of this security solution, which it calls Cisco NAC or CNAC. The company believes that it offers the best NAC solution and outlines how its NAC solution addresses security. Their approach to network admission is built into the network architecture, with support for industry standards including Extensible Authentication Protocol (EAPOL),802.1X, and RADIUS, as well as hundreds of third-party industry suppliers like anti-virus, IDS, IPS, and malware support providers. Cisco NAC also employs the network architecture to impose security regulations on all end-point devices that connect to the protected network. It does this by verifying that all devices before entering the network comply with the stated security policy and isolates that do not. Non-compliant devices that have been separated (otherwise referred to as "quarantined") can be resolved and returned to a "compliant" position by updating their devices with policy-specific data and thus can be part of the secure network.

Cisco NAC prioritizes network policy enforcement at the core network level e.g. switches or routers. Cisco's customers can leverage their existing network investments for security applications, as Cisco NAC collaborates with security solutions from IBM, McAfee, Symantec, Trend Micro, and more than 70 other companies as partners with the Cisco NAC framework approach, allowing solutions from various vendors to be integrated into the Cisco NAC. The Cisco NAC Appliance enables quick NAC implementation by providing self-contained endpoint assessment, policy administration, and remediation

services, as well as Microsoft Corp. and major antivirus vendor patching and updates. It takes into account network location and access methods such as LAN, wireless, remote access, and WAN. Cisco Systems Inc. provides policy enforcement on all devices, whether managed or unmanaged. It also provides a wide range of compliance data, for example, in addition to examining antivirus, firewall, or security patches, it can also check up on the encryption methods used in VPN, ensuring that whoever remotely connects to the network does not compromise the confidentiality and integrity of the data.

Cisco provides two implementations of NAC which are the Cisco NAC Appliance and the Cisco NAC framework:

1) Cisco NAC Appliance

The Cisco NAC appliance was originally known as the Cisco Clean Access [14]. This version provides a one-box solution for endpoint assessment, remediation, policy enforcement, and management. This method is recommended by Cisco for early NAC implementations, especially when starting from scratch as it decreases the degree of complexity since the NAC appliance does not require changes to existing network architecture and may also be installed as an overlaying approach. This appliance is made up of the following server components as shown in Figure 5.29 below:

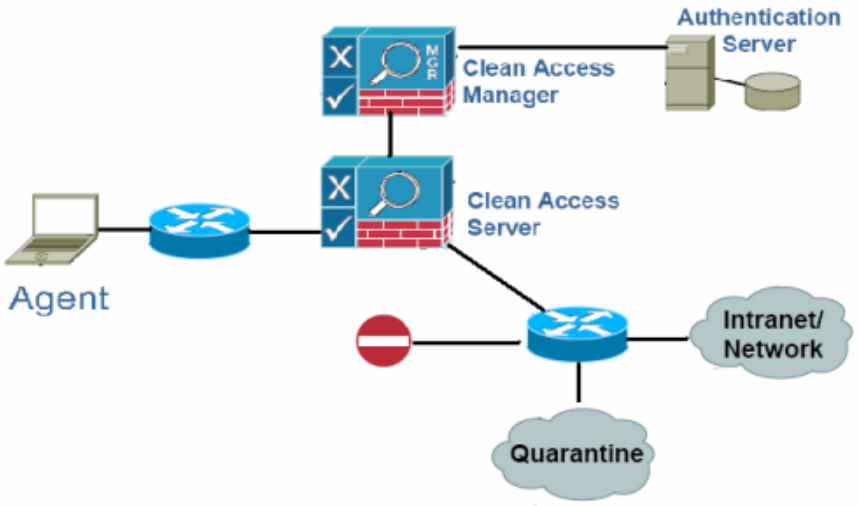

FIGURE 5.29: Core components of NAC Applicance[7]

- Clean Access Manager (CAM) centralizes administration through an HTML-based interface. It functions as a AAA RADIUS server, and the goal of the CAM is to create security policies and remediation requirements for the protected network.
- Clean Access Server (CAS) is the enforcement server that enforces all of the security policies that the system administrator has configured in the CAM. It communicates with the policy servers of various third-party suppliers to evaluate whether

an endpoint is suitable for network access. Antivirus servers, audit servers, and vulnerability management servers are examples of third-party products. As a RADIUS server, the CAS often ensures authentication by interacting with Microsoft Active Directory or LDAP to determine if the user has valid credentials to access the network.

- Clean Access Agent (CAA) is an optional lightweight client that is in charge of doing a thorough examination of the machine's security profile by examining registry settings, services, and so on. This agent ensures that the client has all of the necessary security apps to comply with the company's security rules and can also be used to authenticate users.

2) Cisco NAC Framework

Cisco NAC Framework approach integrates network infrastructure and third-party devices to enforce policy compliance on all endpoints. This is especially important when interoperability and interaction with other manufacturers, as well as scalability requirements since the NAC Appliance solution does not scale well. [7]According to Cisco's website and other promotional materials, the NAC framework design enables a wide range of devices utilized in a realistic network environment, as well as collaboration with more than 100 third-party security providers. The CNAC framework design requires an endpoint to communicate its authentication credentials and posture to a specific system responsible for that job, which then examines for its approval before granting access to the company network.

The following Figure 5.30 shows the architecture of the CNAC framework:

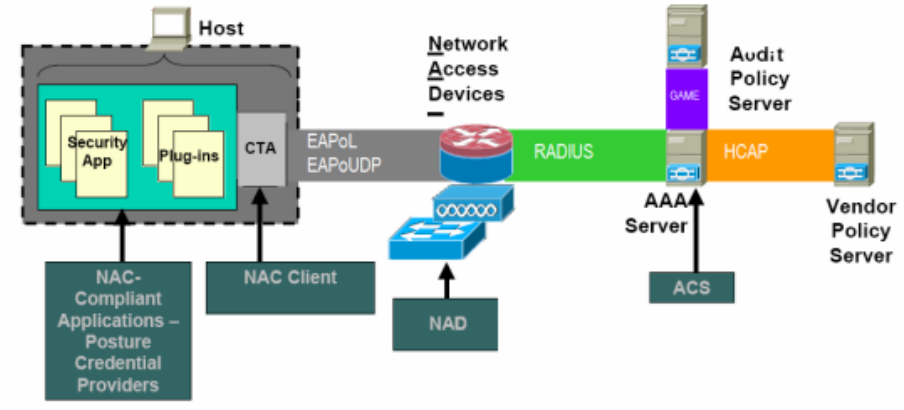

FIGURE 5.30: Core components of NAC Framework[7]

- Cisco Trust Agent (CTA) lives on the endpoint device and collects user credentials, authentication data, and the security posture of CNAC.

- Cisco Network Access Devices (NAD) can be a form of any device that is compatible with CNAC and can be utilized for network security policy enforcement. NAD is CNAC-enabled cisco systems devices that correlate to network installations such as LAN, WLAN, VPN remote access, and MAN.
- Cisco AAA Policy Server is composed of Authorisation, Authentication, and Accounting (AAA) is the process through which endpoints must authenticate before gaining access to the network, as well as tracking users for resource usage. The AAA policy server in this system is the Cisco Secure Access Control Server (ACAS). It handles all policy choices inside the NAC architecture, as well as establishes the endpoint state of connected devices.

5.1.2.3 Microsoft Network Access Protection

Network Access Protection (NAP) is a Microsoft Corporation solution aimed at contributing to Network Access Control with the main goal of keeping networks healthy [15]. NAP handles network access control by maintaining computer compliance of machines such as home computers, Intranet computers, and traveling portable computers, protecting them from cyberattacks, and enforcing compliance on system compliance. This platform has an integrated method for detecting the condition of a network client that is seeking to connect to a network and blocking network client access until the policy requirements for connecting to the network are fulfilled. Network administrators may use NAP to monitor the health of laptops that reconnects to the business network, whether through a VPN connection or by physically returning to the office.

NAP includes the feature of automated remediation; NAP may be set for automatic remediation so that NAP client components can attempt to update the client's computer automatically when the client is non-compliant. Moreover, NAP auto-remediation shortens the period a non-compliant machine is blocked from accessing the organization's network resources. Auto-remediation can quickly update the computer utilizing resources provided in the restricted network (quarantine), allowing the non-compliant client to certify its verify its repaired health condition and get unlimited network access.

Microsoft's NAP is not intended to protect a network from unauthorized users; rather, it is intended to assist administrators in maintaining the compliance of machines on the network, which aids in the overall integrity of the network NAP cannot prohibit an authenticated and authorized user using a compliant machine from propagating a malicious application or engaging in other malicious activity. However, it can accomplish so by adding associated functional components via its API.

The NAP architecture concludes with the following components as shown in Figure 5.31 below:

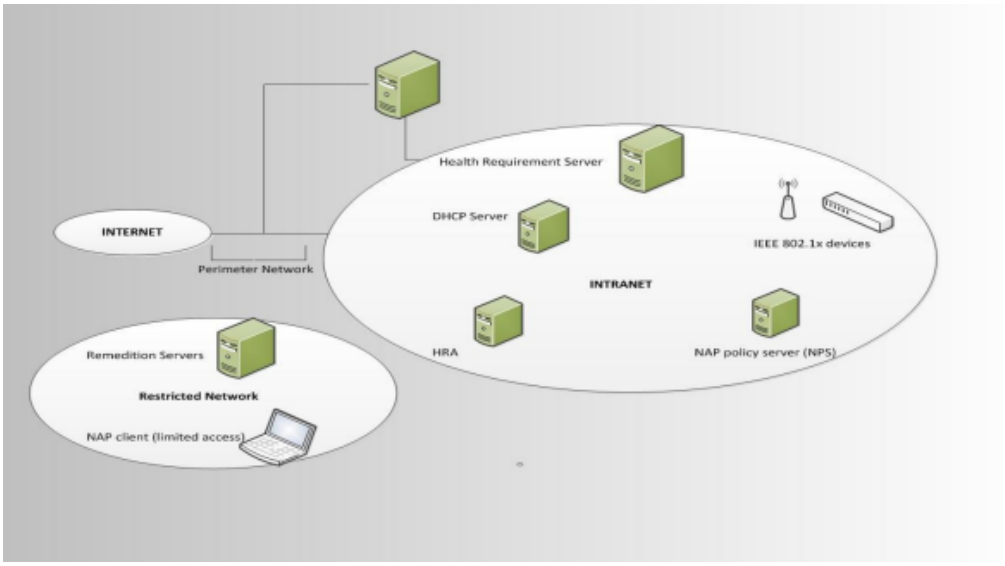

FIGURE 5.31: NAP Infrastructure[1]

- NAP client is essentially a computer that reports its essential system health to a NAP enforcement point, which then sends the client's health status to the NAP health policy server for thorough review over the Radius protocol.
- NAP enforcement services are devices that get a health certificate and are generally 802.1x switches and servers that restrict network access after evaluating a NAP client. VPN servers, DHCP servers, and wireless access points are examples of NAP enforcement points.
- NAP health policy servers are health servers that can be configured by the administration to contain security settings, as well as clients' health, needs to access the network. These servers decide which version or updates are required for NAP client installation.
- Remediation servers aid in the process of fixing NAP clients in the restricted network for non-compliant clients. We can acquire access to the servers here and install or upgrade their systems before conducting a full health evaluation.

NAP supports four different forms of enforcement: IPSec, DHCP, VPN, and 801.2X. Within a network, one or more enforcement types can be configured, and these technologies should be chosen based on the network's infrastructure.[15]

- 802.1x Enforcement uses 802.1x compliant switches and APs(access points). Before each computer is authorized, the server operating the NPS service will verify its

health condition, and if it is not consistent with the network's criteria, the machine will be sent to the remediation network. Special filters, like ACLs (Access Control Lists) and/or VLANs (Virtual Local Area Networks), are used by switches and access points to isolate non-compliant computers from the rest of the network. After the updates and patches are completed and the system is compliant, it is granted network access.

- DHCP Enforcement utilizes DHCP(NAP) server to check the health of the client and, if the client is healthy, leases a valid IP address to the client, allowing the client to access the network. If the client is not in good health, the DHCP server will lease limited IP addresses and a set of routes to Remediation Servers on a restricted network in order to offer updates and patches and take other steps to bring the client into compliance. After this operation is completed and in compliance, the DHCP Server will lease full IP addresses and the client will be able to join the network.

- VPN Enforcement operates when a VPN client attempts to connect to the network and a VPN server determines if the client is healthy or not, either through NPS or RADIUS server. After the VPN's health is established and correct, a VPN connection and the remote client are permitted to access the network. However, if the client is not healthy, the VPN Server employs a set of packet filters that quarantines the client by allowing it to connect only to the limited network where Remediation servers are located. Once the client has been updated or patched, the VPN server removes the packet filters and allows the client to connect to the network.

- IPsec Enforcement requires us to deploy the Health Registration Authority(HRA) role service and have established PKI to use the IPsec enforcement mechanism. The HRA role service is in charge of distributing system health certificates to NAP-compliant clients via a Web application named DomainHRA. We must specify the Health Registration Authority Web site location for clients using Group Policy if we want to utilize this enforcement method.

5.1.2.4 NAC Comparison Overview

Network Access Control is a centralized method of achieving network security that imposes regulations on all devices and users. The fundamental purpose of NAC is to prohibit unauthorized devices or users from joining a private network. This is usually achieved by employing zero-trust access solutions, which enable visibility into all devices on a private or corporate network. All of the above vendors that we provided an overview of including the open-source solution PacketFence will help clients to achieve this. The Network Access control technology is not a new innovation in the network security genre as it has been around for nearly two decades. Cisco Systems Inc, originally launched NAC back in 2003. NAC is made of up several components and new technologies spanning from hardware to software. According to Forrester Research, around 40 percent of companies began implementing NAC in 2006, with approximately 52 percent indicating a need for access control across all network mediums: wired, wireless, and remote access.

Both Cisco NAC and Microsoft NAP have strong claims to be the architects of NAC solutions. Cisco is the major provider of network infrastructure that can restrict network access and quarantine non-compliant devices while Microsoft dominates the operating system market and can integrate a policy-enforcement client into the OS directly. Each of these solutions requires a lot of effort to implement and integrate into an existing network. Cisco NAC solution mainly focuses on network policy enforcement at the core network level whereas Microsoft NAP mainly revolves around ensuring the maintenance of device compliance with the goal of keeping networks healthy. In comparison, PacketFence will provide both functionalities of enforcing network policies while also maintaining device compliance through the integration of third-party software which is impressive for an open-source solution.

Cisco's approach mainly relies on 802.1x, which needs supplicants on all network-attached devices, including network printers and other peripherals which may result in needing to upgrade the existing network to handle 802.1x protocol. Microsoft NAP on the other hand relies on a collection of operating system components that provide a foundation for secure access to private networks. PacketFence employs the Statement of Health (SoH) protocol to carry out an in-depth posture analysis or evaluation of registered devices with the help of the 802.1x protocol similar to the Cisco NAC solution.

The majority of the NAC market is centered on Microsoft Windows technology therefore other platforms may receive less support. Agentless vulnerability scans are often performed on machines that do not run Microsoft Windows because agent software for these systems are not installed or available. Less support for platforms other than Microsoft poses a threat to the open-source software ecosystem in the sense that if the evaluation

is based on trustworthy apps, a set of open source applications may be excluded from the list of trusted applications.

Architecture	Network Admission Control	Network Access Protection	PacketFence
Vendor	Cisco Systems, Inc	Microsoft Corporation	Inverse Inc Open-source
Solution Type	Appliance	Software (Servers)	Appliance
Enforcement Points	Cisco Clean Access Server, Switches & Access Points	Machines having Windows Longhorn Server	Router, Switches, Wireless Controller & Access Points
Deployment Setup	Inline & Out of band	Windows Server	Inline, Out-of-band, Hybrid, RADIUS & DNS
Cross Platform	Windows	Windows	Linux, Red Hat & Debian 11
Enforcement Technologies	DHCP	802.1X, IPSec, DHCP & VPN	802.1X, MYSQL, Apache, DHCP, FreeRADIUS, Snort, Suricata

FIGURE 5.32: Comparison overview of architectural components

5.2 Difficulties Encountered

Enumerate the different difficulties you have found when developing your solution approach. Create three categories of difficulties:

- **Easy**:

  1) Gathering of required Virtual Machines and Networking Equipment - While we planned to use open-source tools for the most part of our project, there were two particular tools (Win Server 2016 and IOS images of Routers/Switches) that we needed to acquire required licenses which would ultimately result in us having to spend money to get our hands on these tools. The level of difficulty would have to be increased to a medium where we would have had to find an alternative to use if we were not able to obtain these tools which would have altered our project plan slightly. However, due to our structured planning in the research phase, we were able to obtain these through our prior completed module (Windows Security) and also from our project supervisor.

  2) Learning to use GNS3 - This application was something that we had not used and were not familiar with before so we needed to learn to use this tool. Some

of the key functionality that we needed to learn was how to set up a GNS3 VM server for us to be able to run appliances, use Dynamips as an emulator (importing and configuring IOS Cisco Router/Switch images), import appliances to use for our topology and also importing external VMs. This difficulty was addressed pretty easily with a bit of reading of documentation on GNS3, however, it did affect our implementation schedule where we had to allocate time for learning.

- **Medium**:

  3) Using PacketFence GNS3 appliance - GNS3 offers PacketFenceZEN as an appliance which is much more straightforward than running the VM on VMware Workstation and then importing it in GNS3. This is also more feasible in terms of computing resource consumption where it would require less RAM and CPU resources to run appliances than imported VMs in GNS3. However, we ran into the issue of not being able to assign any IP address to the PacketFenceZEN appliance which means that we had no way of accessing the web GUI interface to configure it. We tried assigning IP through various methods such as manually putting in a static IP in **/etc/network/interfaces** configuration files but that static IP seemed to wipe out every time we rebooted PacketFence. DHCP assignment from our DHCP pool that we configured in the router also did not seem to be able to assign any IP address. This was quite a challenging difficulty as there was little to no information to be found on the web so our only solution was to use the imported PacketFenceZEN VM in GNS3. This resulted in the need to adjust our RAM allocation to other VMs that we needed to run at the same time which overall added extra overhead to our system.

  4) Physical Resource Constraints - This was a tricky difficulty that we faced while doing our implementation. Our initial plan was to run two imported VMs in GNS3 but as we discussed in our previous difficulty of not being able to run PacketFenceZEN as an appliance meant that we needed to compromise our RAM allocation accordingly. Our local system had a total of 16 GB of RAM that we could use for our implementation but from our research, this was not quite adequate enough to run VMs to their full capacity. This also meant that we needed to keep our network topology simple and the bare minimum which could possibly hinder the best possible outcome of our project. We were able to devise a solution by testing how much RAM each VM actually requires to run to keep our network topology functioning. We reached the point where we were almost using all of our available RAM (98.2 percent) while running all of the components at once. Figure 5.33 below shows us the computing resources consumption of our two servers which are used to manage VMware Workstation, Dynamyps, and KVM components within our topology. Our topology components are run as a GNS3

appliance downloaded from the GNS3 marketplace and some are run and directly imported from VMware Workstation.

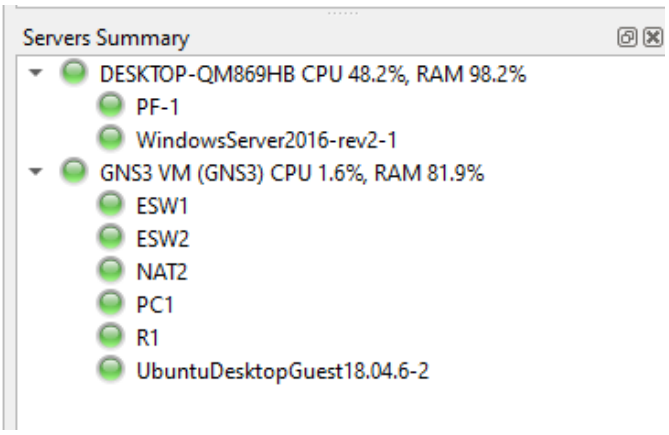

FIGURE 5.33: Resource Consumption Summary in GNS3

- **Hard**:

  5) Integrating Active Directory into PacketFence - This was one of the features that we planned to implement from our research phase planning for user management but after we had set and configured up both PacketFence and Active Directory, we were not able to successfully join the specified AD Domain as shown in Figure 5.34 below. This issue seems to be persistent even after performing various troubleshooting steps and debugging, we decided to take an alternative route and leave this component for future work. This did not impact our project goal significantly as we came up with a revised solution to create our users and groups within the PacketFence Web-based Administrator GUI which worked out seamlessly in the end.

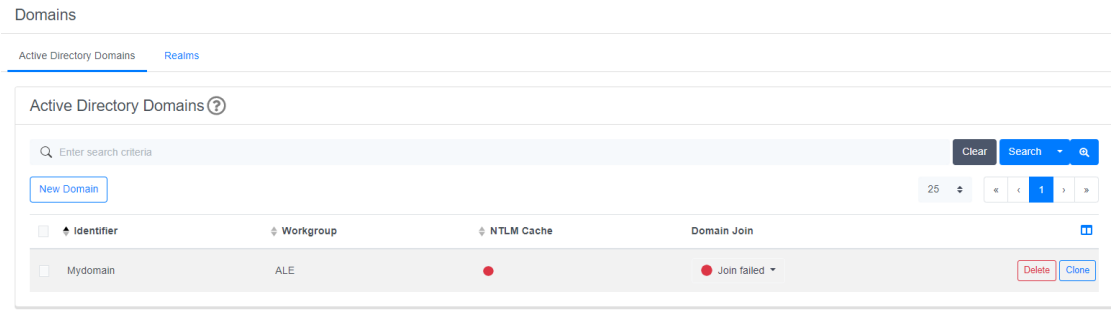

FIGURE 5.34: AD Integration Error

Chapter 6

Testing and Evaluation

6.1 Metrics

6.1.1 Network Configuration Verifications

1) Network Connectivity

Are the components within our topologies able to communicate with each other?

2) DHCP and DNS Server

Do DHCP and DNS services within PacketFence automatically distribute IP and DNS addresses to our endpoint devices?

6.1.2 Network Access Control through PacketFence

1) Have we tested the Role-based Access Control?

2) Have we tested the Guest Access Control?

3) Were we able to monitor these tests in the PacketFence Web-based Administrator GUI?

6.1.3 Management of Networking Equipment through PacketFence

1) Were we able to add our Cisco Switches to the PacketFence?

2) Were we able to manage Endpoint devices through Cisco Switches?

6.2 System Testing

We set up three endpoint devices within our Network Topology as shown previously in the implementation section to test the authorization and registration provided PacketFence NAC solution. PC1 was used mainly for verifying network connectivity within our topology as it provided an easy way to run our various tests through its command line interface. The two Ubuntu desktop clients were then used to test the network access control of authorization and registration of endpoint devices for our testing phase. The local user credentials of Ubuntu Desktop Clients were supplied in the installation guide from the GNS3 marketplace as this appliance comes with a pre-configured version with local user **"osboxes.org"** and password **"osboxes.org"** already created.

We also used Wireshark to observe information on various packets and analyze the flow of traffic on each interface for our network verification. This tool is pre-installed that comes with the initial GNS3 setup. GNS3 enables us to use this external feature by simply right-clicking on the connection link between the interfaces and provides us with the option to start packet capture with Wireshark. We can utilize this to analyze and monitor the types of data transmitted within our network and possibly use it as a debugging method.

6.2.1 Network configuration verifications

1) Distribution of IP addresses through DHCP server

We verified that PacketFence provided DHCP IP addresses for all of our endpoint devices through the Inline interface. We provided a screenshot as an example from one of our endpoint devices (VPC) receiving a DHCP IP address and observed the DHCP packets in Wireshark at the connection point between the switch and the PacketFence VM. For further verification, we also analyzed if the node (VPC) had been detected in the PacketFence Admin GUI interface and we verified that the device had been registered as a node as seen in Figure 6.1 below.

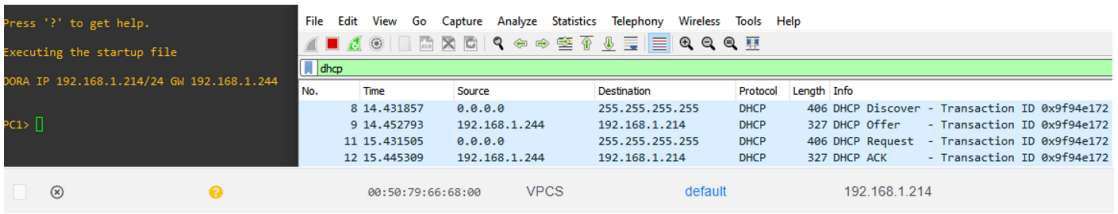

FIGURE 6.1: Endpoint device IP lease and node detection

2) Network Connectivity

We tested to make sure that our end devices can establish connectivity to the PacketFence VM management network interface (192.168.0.85) for example via ping command from PC1 and observed the ICMP packets.

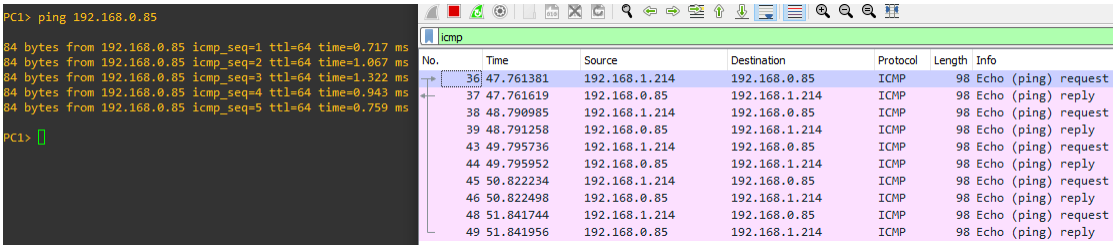

FIGURE 6.2: Ping test to PacketFence management interface with wireshark

We also tested the connectivity to the Inline IP address (192.168.1.244) to ensure our inline interface can communicate with the endpoint devices. This is the first step towards enforcing network access control within our topology.

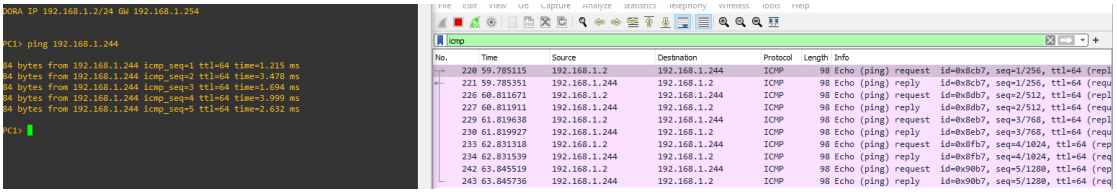

FIGURE 6.3: Ping test to Inline PacketFence IP with Wireshark

6.2.2 Network Access Control through PacketFence

The main goal of this project was to showcase network access control through PacketFence by enforcing an additional layer of security to limit users accessing network resources such as Internet access. PacketFence will restrict internet access within endpoint devices even after the IP address has been allocated for that device through the DHCP service provided by PacketFence. PacketFence provides authorization and registration features [2] for an unregistered device through a Captive Portal where only registered and managed users within the PacketFence database can self-register and access the network resources on their end devices. A captive portal is a type of Web page that public network users must read and log in with valid credentials before gaining access to the private network. One of the ways that Captive Portal can be configured is through Web authentication (Hotspot-style authentication) which is what we tested in this section.

The majority of PacketFence ZEN network access control features are derived from the Web-based Administrator GUI through the node section on the interface. Once clicked on a particular node and from the edit tab we are able to enforce various actions against the node once the node is registered on the network. These actions include changing the device owner, registering or unregistering the device within the network through the status option, and editing roles based on the access required. The violations that can be applied include access time balance and bandwidth balance. We are then able to re-evaluate access for a particular device by defining the required actions to deny access to the network and prompt the device user to the captive portal for logins again. This enabled us to efficiently examine various authentication methods in the captive portal using the same device through the **"Re-evaluate"** access feature of PacketFence.

6.2.2.1 Role-based Access Control Test

It is important to have a segregated network in any real production environment which is one of the key features of the PacketFence NAC solution. For instance, Students will not have the same degree of network access compared to a Lecturer Staff member. We created three different roles (Student, Staff, Admin) and three sample users to test and explore this feature. The nodes assigned represent the number of devices each user can register depending on their role. For example, we set two nodes for Students, five for Staff members, and unlimited for admins for this particular demo. Node specifies the number of devices that each user role can register at a particular time.

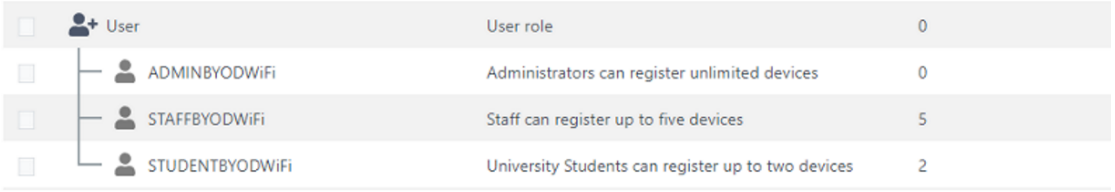

FIGURE 6.4: Role-based Access Control sample roles

The following sample users were created and stored within PacketFence Web-based Administrator GUI to accommodate the Authorisation and Registration process in the Captive Portal to test role-based access control as shown in Figure 6.5 below.

Username	Password
Dipen (Student)	demouser1
Jack (Admin)	demouser2
John (Staff)	demouser3

FIGURE 6.5: Creation of sample users within PacketFence

We specified different access types and policies for each user which can be seen in Figure 6.6 below.

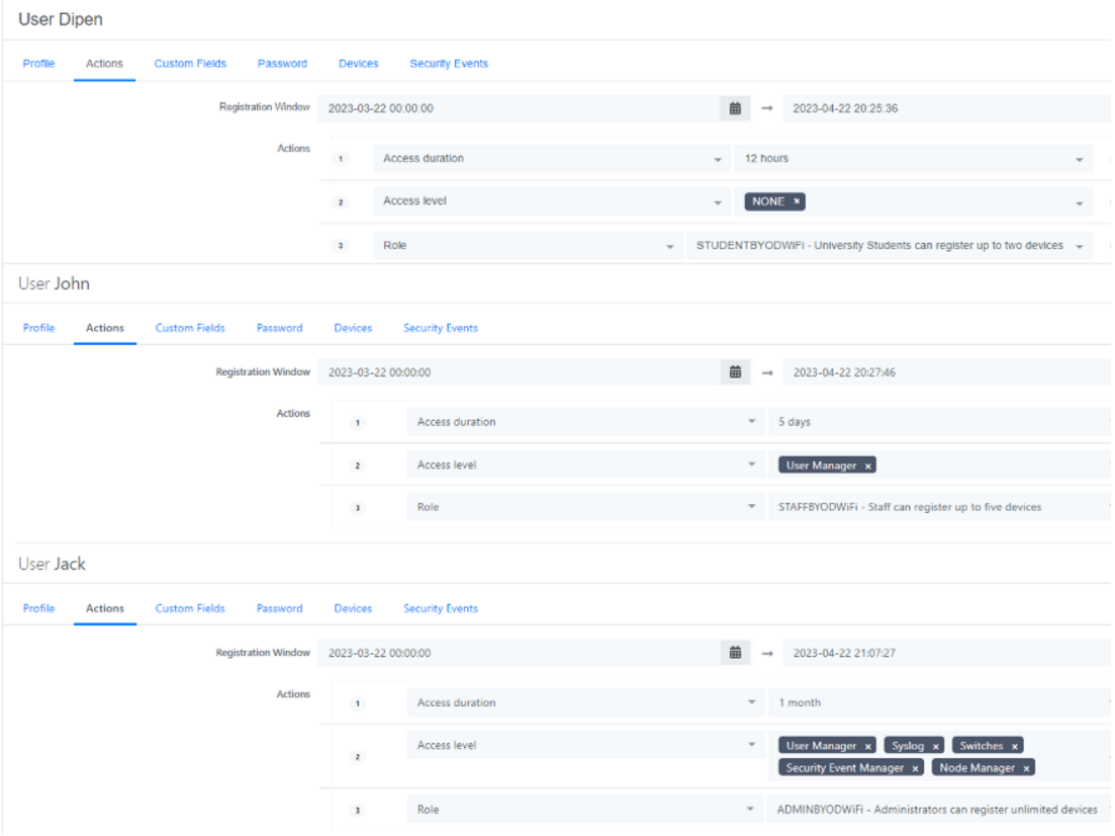

FIGURE 6.6: Role-based Access Control sample users access policies

For this instance, User **"Dipen"** is assigned a Student role therefore the access level is set to **"none"** meaning the user does not have any permission from an administration perspective. In regards to user **"John"** is assigned with Staff role and has access level set to **"User manager"** meaning John can manage users from an administration aspect. Lastly, user Jack is assigned to the Admin role and the access level is set to being able to administrate various aspects such as managing users, nodes, security events, accessing syslogs, and switches.

Now, we looked to examine and showcase that Packetfence provided the functionality of registration for endpoint devices through a captive portal in order to log in and gain access to network resources (Internet). We used Ubuntu Client PC as an unregistered device where we received an IP address **"192.168.1.99"** from the DHCP server provided within PacketFence. Even at this stage, we will have no access to the Internet as PacketFence restricts this resource until the local user of the device has logged in with a valid credential that resides within the PacketFence database. For a demo, We then opened a web browser where we are prompted to log in to the network when we tried to access the PacketFence website as shown in Figure 6.7 below.

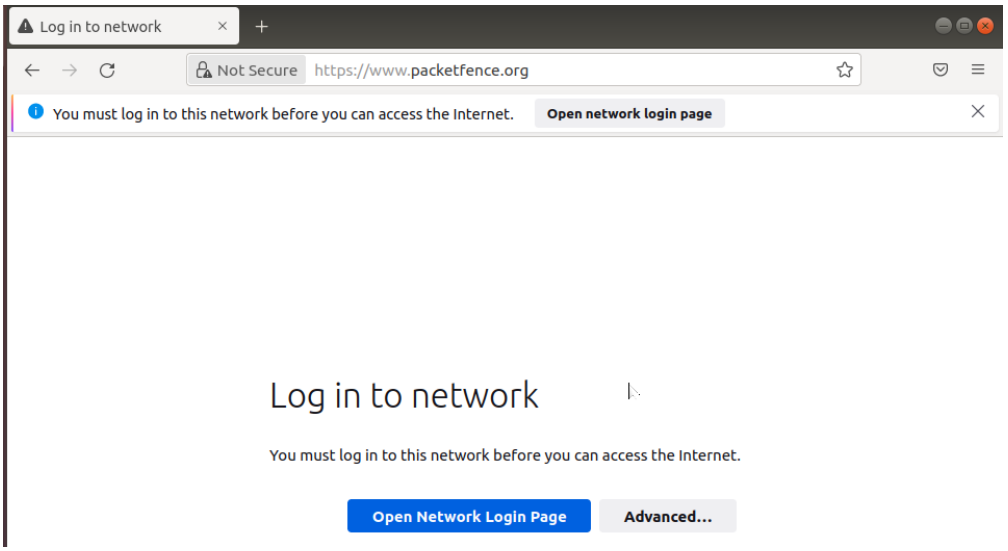

FIGURE 6.7: Network login prompt in unregistered device

Once we clicked on **"Open Network Login Page"**, we were prompted to choose our authentication method for our logon. As this particular test explores role-based access control, we will log in with one of the users that we created in the PacketFence Web-based Administrator GUI previously by clicking on the **"Username/password login"** option.

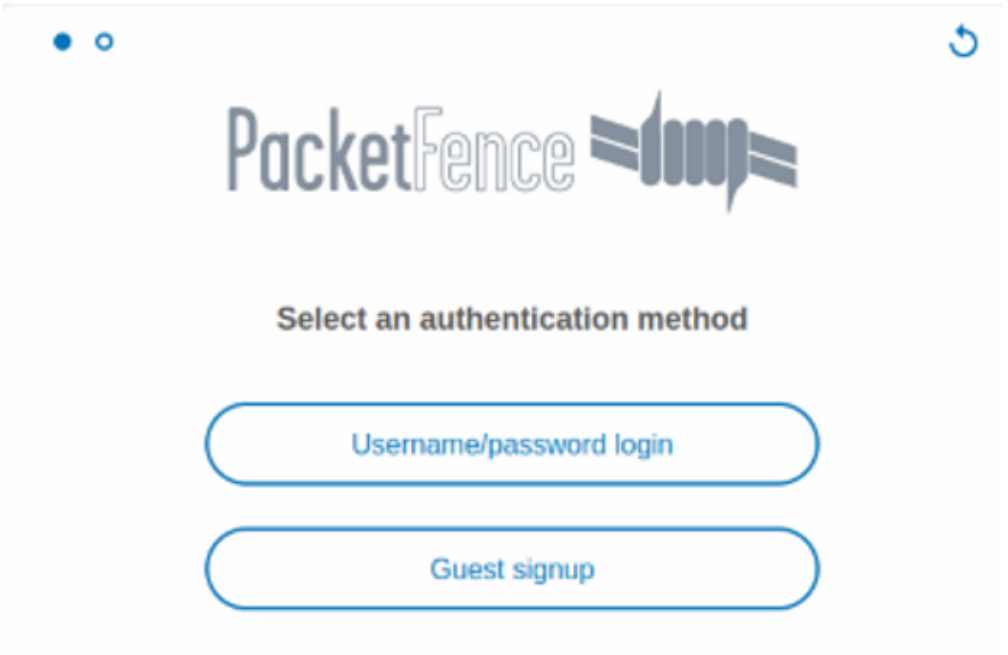

FIGURE 6.8: Captive Portal Authentication methods

We successfully logged in with user Dipen (Student) credentials and we were presented with a notification that specified the type of role associated with the user and a message stating our network access being enabled along with a progress bar.

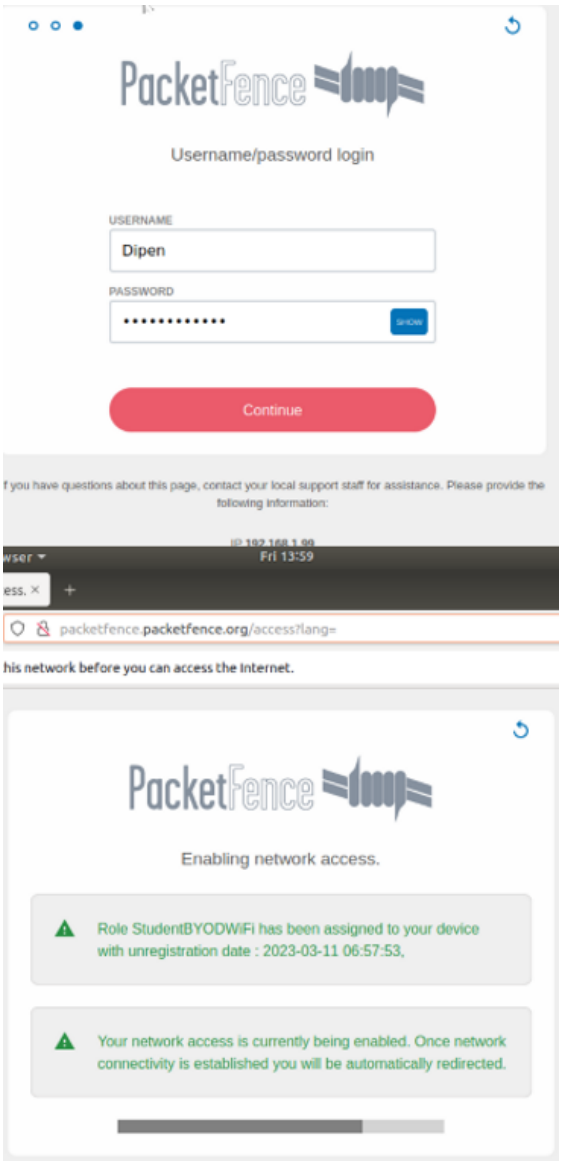

FIGURE 6.9: Role-Based Access Student role Sign In

After the previous step had been completed, we were able to access the PacketFence website for example, and were successfully able to gain network and internet access as seen in Figure 6.10 below.

FIGURE 6.10: Network and Internet Access from Captive Portal

6.2.2.2 Guest Access Control Test

In this section, the guest authentication method was examined which is an important feature for any organization to have an option for guests to access their network resources. PacketFence provides multiple sources for gaining access to the network resources for Guest roles as shown in Figure 6.11 below.

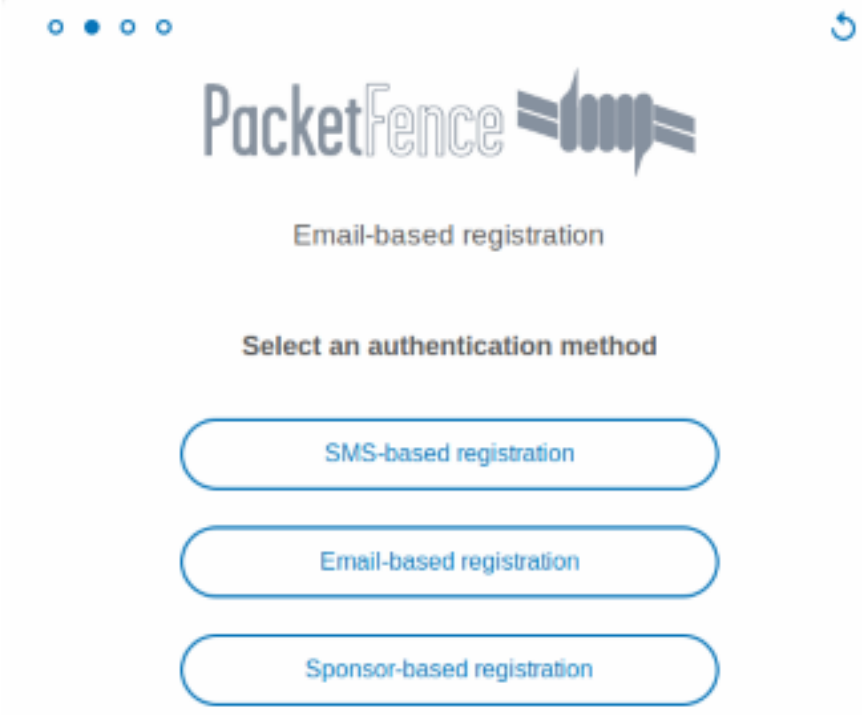

FIGURE 6.11: Guest Authentication Methods

Both SMS and Email based registration serve as convenient methods for registration where guests simply have to provide their phone or email details to gain access to the Internet. For this particular test, we proceeded with email-based registration to showcase the process of registration with a sample email **"demoharry@gmail.com"**. Upon successful registration, temporary network access will be provided for 10 minutes, and for full access duration the guests must open the link sent to their email as shown in Figure 6.12 below.

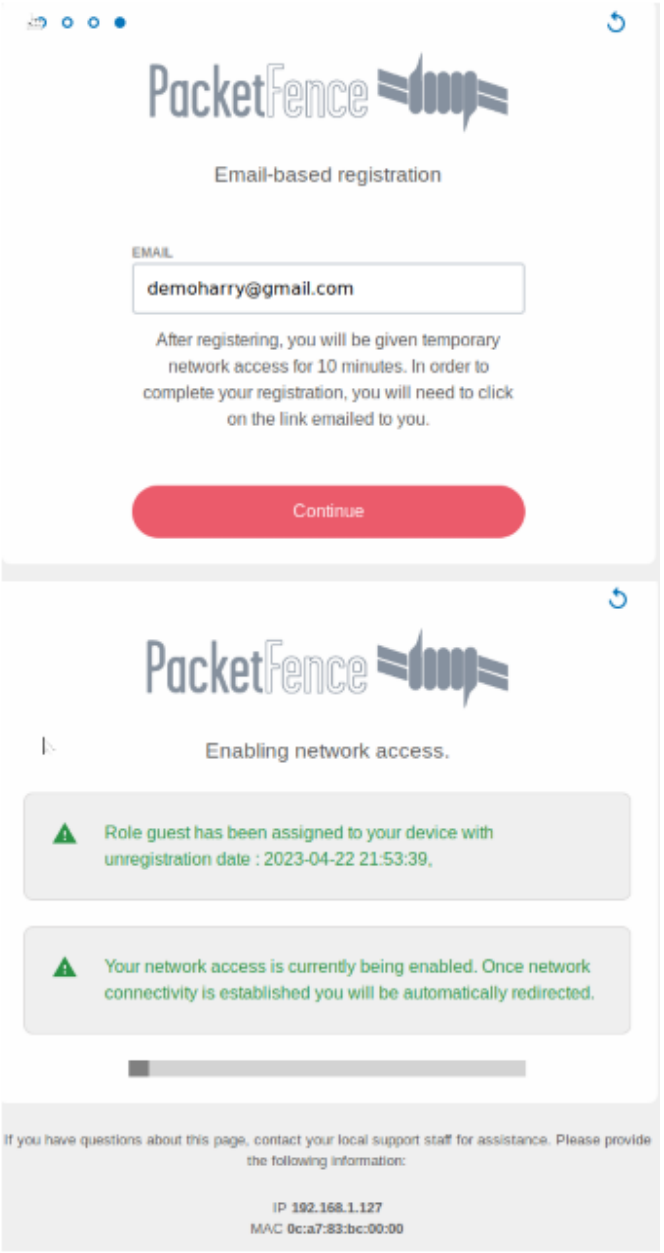

FIGURE 6.12: Email-based registration for guests

Another Guest registration method that we examined was Sponsor-based which is a more secure way of enforcing Guest access control. For this method to function, a user within PacketFence must be assigned with the policy of being able to act as the sponsor for a particular guest. This can be defined in the users' section within the PacketFence Web-based Administrator GUI and assigned with the policy **"Mark as sponsor"**. For this particular test, we assigned user **"Jack"** who is an Admin with the policy **"Mark as sponsor"** to restrict only admins from being able to sponsor Guest users as shown in Figure 6.13 below.

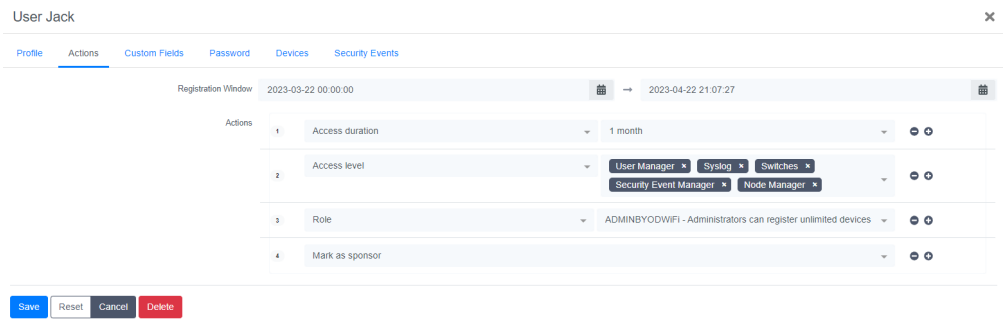

FIGURE 6.13: Assigning sponsor policy to an admin

This feature was then tested through the guest authentication method and choosing the **"Sponsor-based registration"** source. Firstly for demonstration purposes, **John@gmail.com** was entered as the sponsor email to showcase the error when guests try to enter a user **"John"** (Staff) who has not been assigned with the access policy of a sponsor.

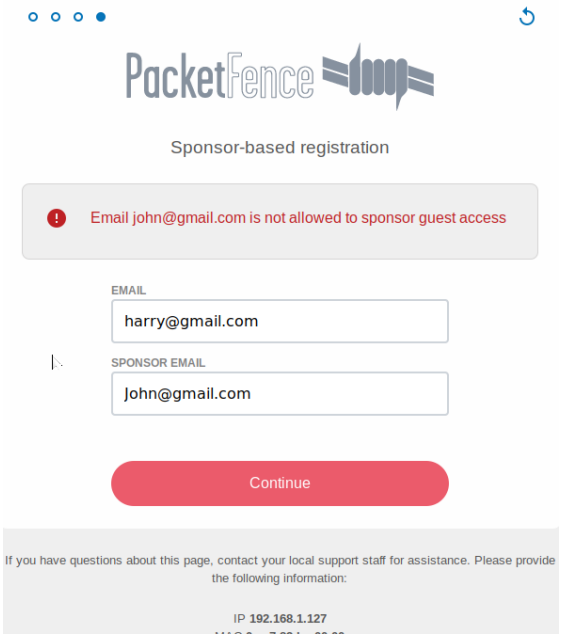

FIGURE 6.14: Sponsor-based registration error

Now finally, a valid user (Jack) who has been assigned as a sponsor previously was entered and successfully registered. The guest user is then prompted to wait for approval by the sponsor (Admin) and only then access to the network will be granted.

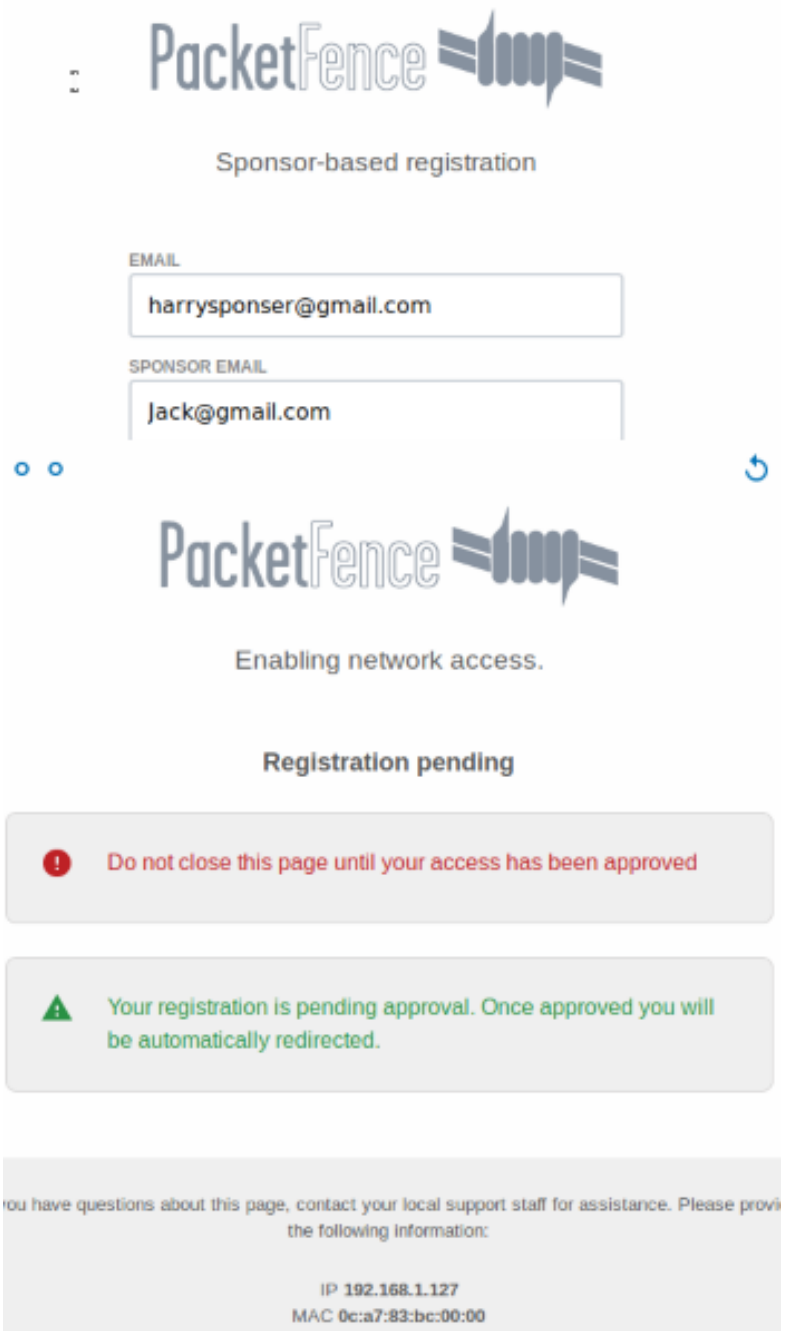

FIGURE 6.15: Sponsor-based registration access with valid sponsor

This access can be manually granted in the nodes section within the PacketFence Web-based Administrator GUI. The device requesting guest access can be identified through its IP address or MAC address within the node section. Once the device has been identified, the device properties can be edited manually to give access by specifying the owner, status, and role fields in the edit tab as shown in Figure 6.16 below.

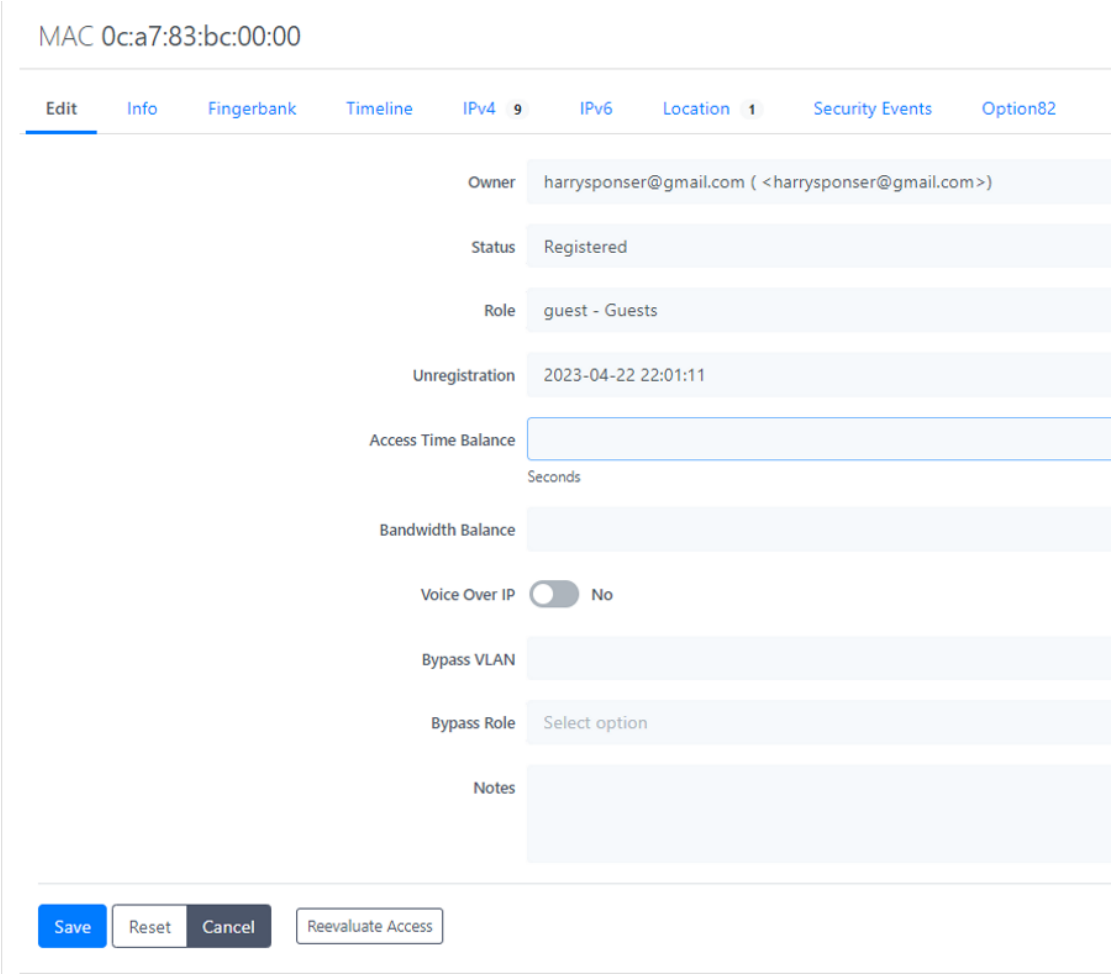

FIGURE 6.16: Manually granting access for sponsor-based guest registration

Once the access has been granted manually by the Administrator (Sponsor), the guests are then redirected to the process of enabling network access with the status progress bar as shown in Figure 6.17 below.

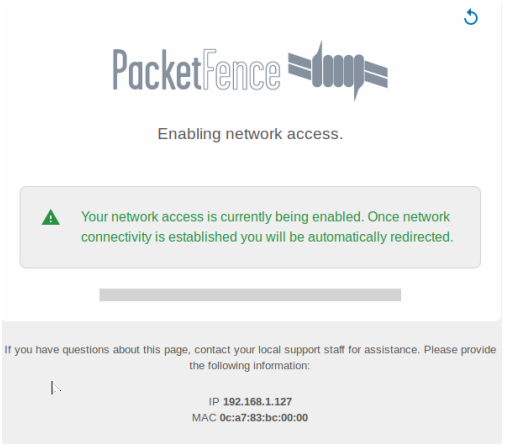

FIGURE 6.17: Sponsor-based access after manually granting access

Lastly, we explored an additional feature of a custom-created authentication method for guest access control. Packetfence provides various options to customize your captive portal where various authentication sources can be manually created as required. For this first test, we created a new connection profile within the PacketFence Web-based Administrator GUI to add a new authentication source where Guests can sign in without providing any valid credential for authentication as an example. We defined our source as **"null"** (no credential required) and specified the inline interface IP address **(192.168.1.244)** that we configured in the implementation chapter as shown in Figure 6.18 below. We will explore the other authentication sources for Guest registration in the role-based access control section later when we have created our sample users.

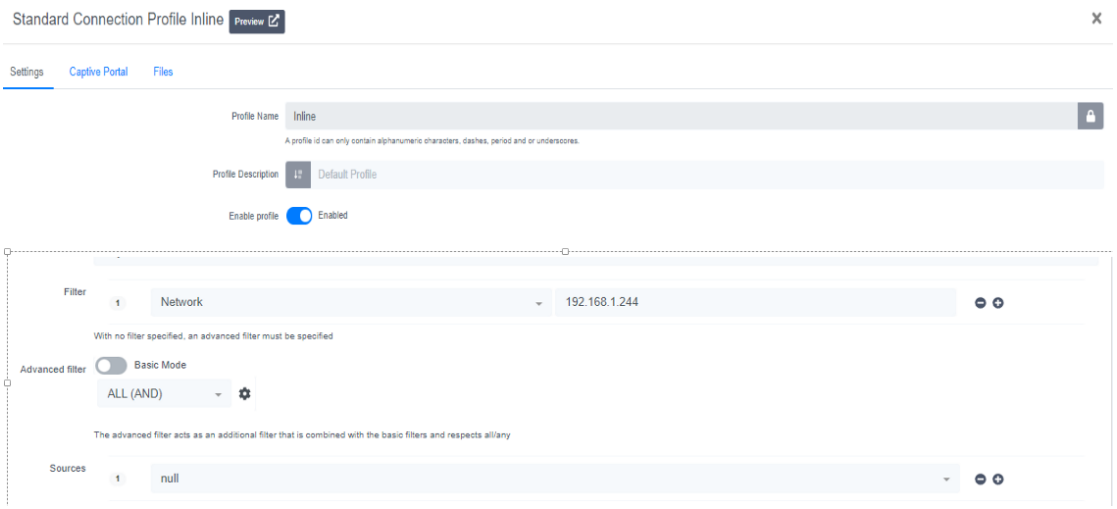

FIGURE 6.18: Connection Profile to create a new authentication source in Captive Portal

After the creation of a new connection profile with a null source, the authentication methods for guests has updated with an additional option for guests to register without any credentials through the null source method as shown in Figure 6.19 below. However, this presents a huge risk from a security perspective in a real-world environment but this was purposely created to showcase that other customized authentication methods can be created within PacketFence Web-based Administrator GUI.

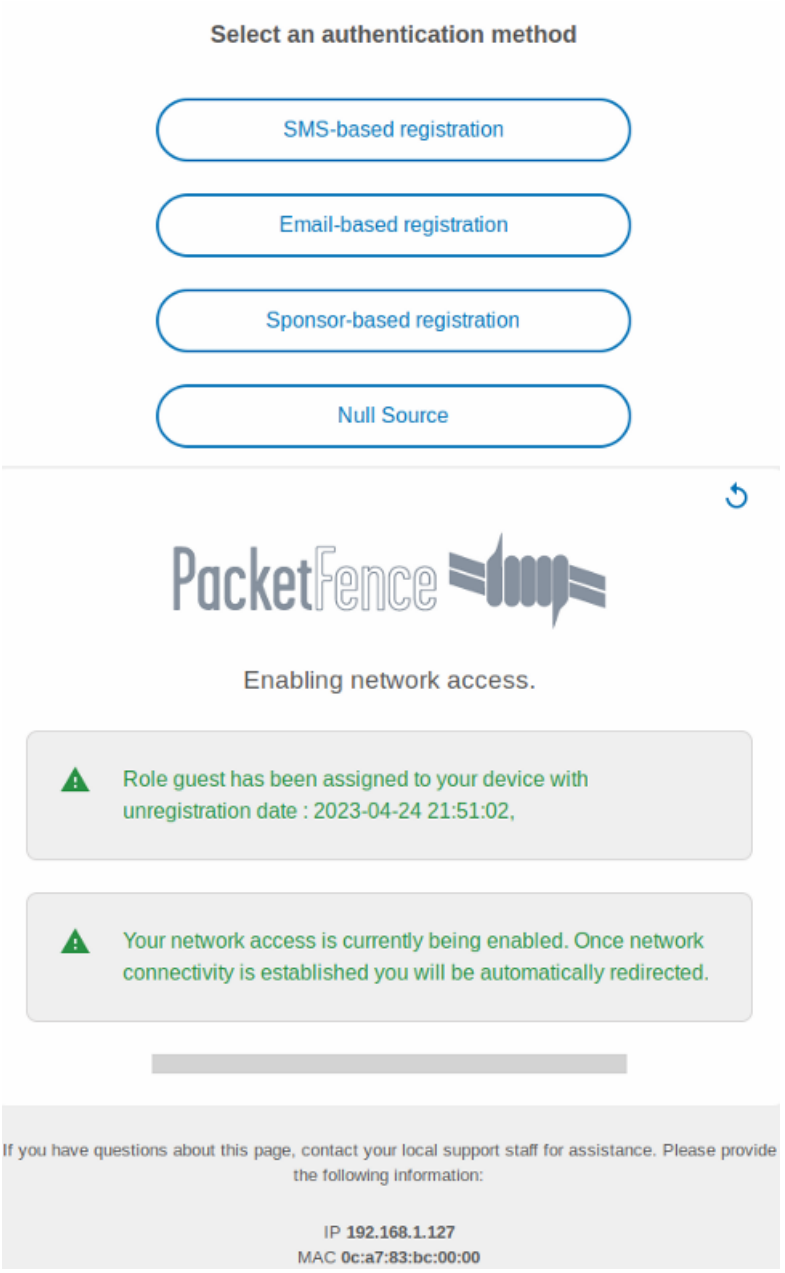

FIGURE 6.19: Null-source authentication and Network Access for the device

6.2.3 Management of Networking Equipment through PacketFence Test

PacketFence enables the management of switches within the network infrastructure. We are required to manually add our switches in the configuration page within the PacketFence Web-based Administrator GUI which was achieved by specifying the switch IP address. Once the switch was added, it enabled us to configure the switch with various service options such as VoIP, VoIP DHCP detect, and dynamic uplinks from the Definition tab. The role tab allows for role mapping by VLAN ID which is more suitable

for VLAN enforcement. The inline tab allows for specifying conditions for inline mode. These features were not exercised due to time constraints and were deemed to be outside the scope of the project but nevertheless, these could be tested in the future.

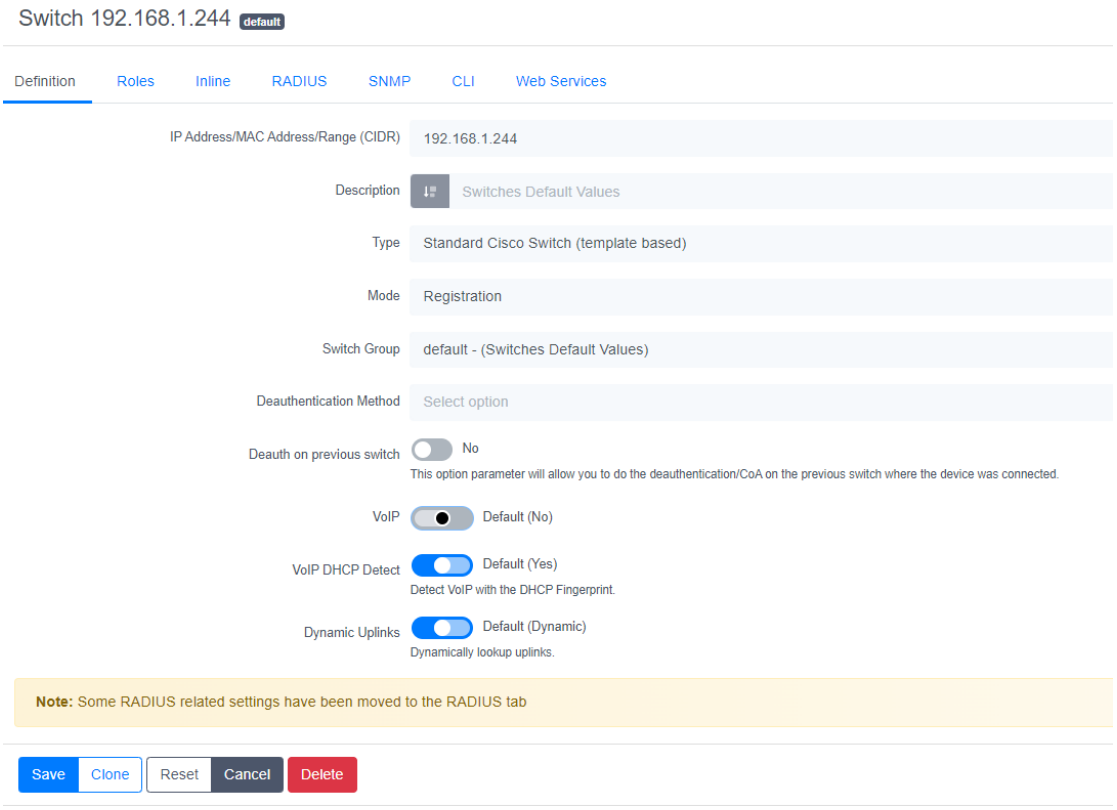

FIGURE 6.20: Manually added Switch into PacketFence

6.3 Results

6.3.1 Network Configuration Results

We successfully configured our network devices to communicate with each other and enabled DHCP and DNS servers within our topology which made it efficient for IP distribution for our endpoint devices. One of the key features of NAC that we discussed in the research phase was Node Identification [2]. Node Identification is the process of NAC detecting each node that connects to the local network in order to extend NAC capabilities. PacketFence allows for this to be verified in the nodes page within the PacketFence Web-based Administrator GUI where every endpoint device connected to the network is detected.

Status	Online	MAC Address	Computer Name	Owner	IPv4 Address	Device Class	Role
✓	?	00:0c:29:4c:e1:93	NAC	default	192.168.1.172		Machine
✗	?	00:50:56:c0:00:02	DESKTOP-QM869HB	default	192.168.1.126		
✗	?	00:50:56:c0:00:06	DESKTOP-QM869HB	default	192.168.1.164		
✓	?	00:50:79:66:68:00	VPCS	default	192.168.1.23		Machine
✗	?	00:50:79:66:68:01	VPCS	default	192.168.1.42		
✗	?	0c:05:7e:54:00:00	osboxes	default	192.168.1.134		guest
✓	?	0c:bd:45:31:00:00	osboxes	Dipen	192.168.1.99		StudentBYODWiFi

FIGURE 6.21: PacketFence Node Detection from our Topology

6.3.2 Network Access Control through PacketFence Results

6.3.2.1 Role-Based Access Control Results

Another functionality provided by PacketFence NAC is its ability to construct an interactive diagram of our network mapping through the Node Identification feature 2. This asset management page detects devices connected within our network and presents us with a linked network diagram format showcasing each component such as nodes, switches, PacketFence, etc. This can be particularly useful to analyze and locate which endpoint device is connected to what switch and helps to get a quick overview of each node's properties. We navigated to the Asset Management and the nodes page in the PacketFence Web-based Administrator GUI to verify the node detection of the device that registered as a valid user (Dipen). The device is listed in both the nodes and asset management page along with the device properties such as the status, the owner, name of the OS, IP/MAC addresses, the role associated with the device, and the IP address of the switch that the device was connected through.

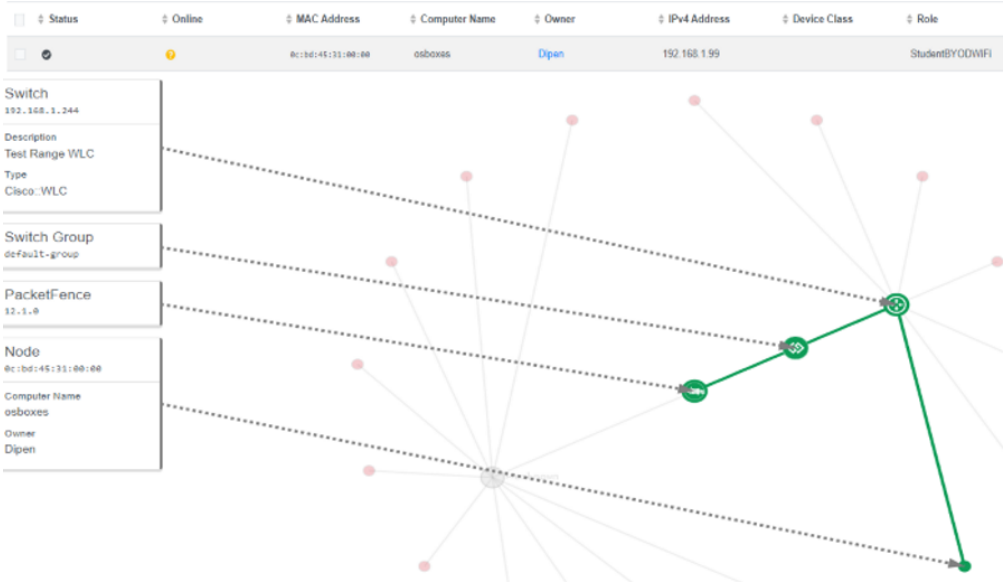

FIGURE 6.22: Role-Based access results

6.3.2.2 Guest Access Control Results

Once the Guest has registered through the Email-based authentication method, a user with an email **"demoharry@gmail.com"** which was used to register by the guest user had been automatically created and populated within the users' page within PacketFence Web-based Administrator GUI. The device was detected and listed in both the nodes and the asset management page through the Node identification feature.

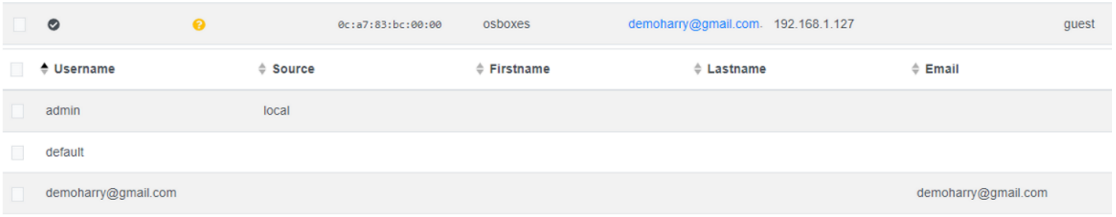

FIGURE 6.23: Guest user automatically created with email-based authentication

Similarly for the Sponsor-based authentication method, after granting access manually by specifying properties for the device in the nodes page, a user with **"sponser-harry@gmail.com"** was created and the device was detected in the nodes page as shown in Figure 6.24 below.

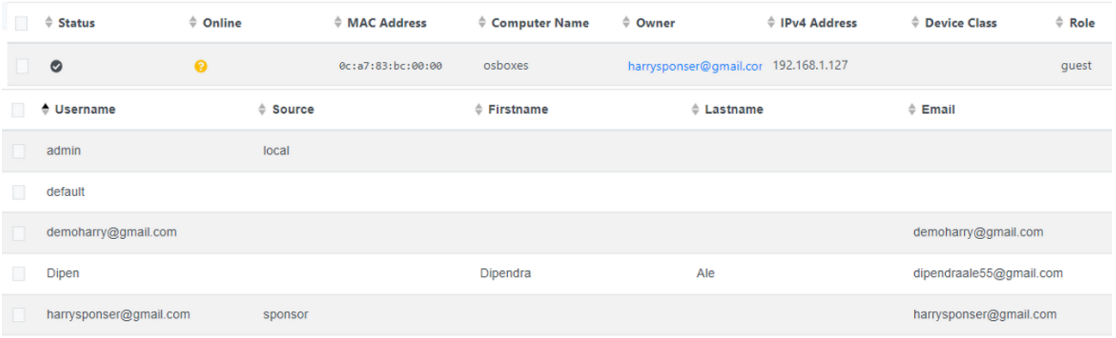

FIGURE 6.24: Sponsor-based authentication results

Lastly for the custom-created **"null-source"** we once again navigated to the nodes and asset management page in the PacketFence Web-based Administrator GUI to verify the detection of the registered node similarly to previous tests.

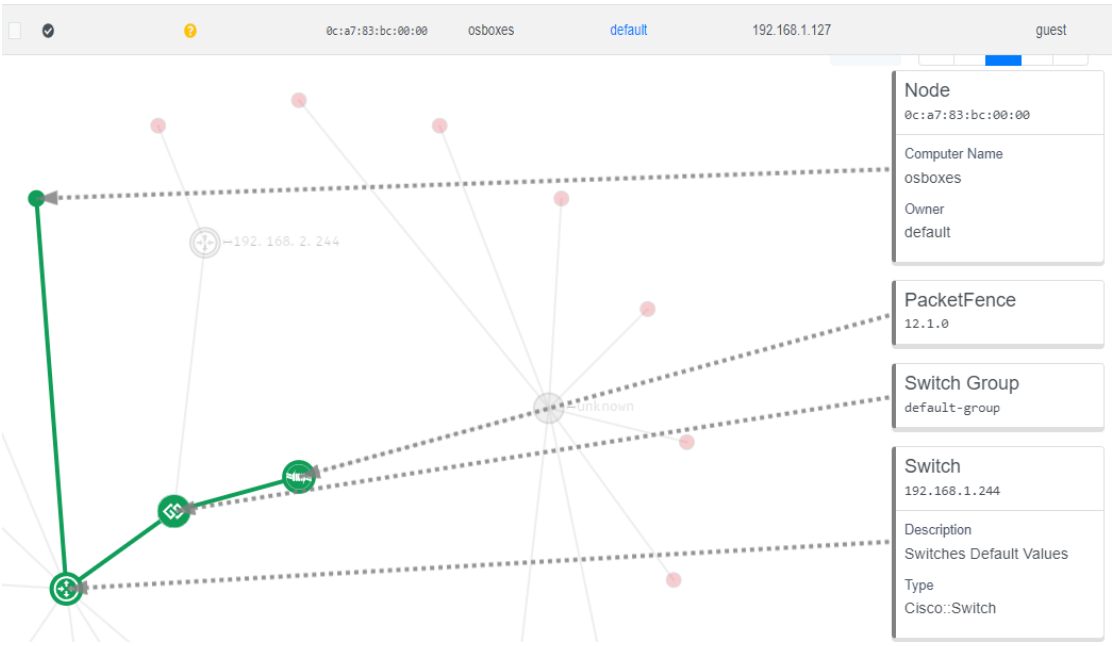

FIGURE 6.25: PacketFence Assets Overview

6.3.3 Management of Networking Equipment through PacketFence Results

Once we had successfully added our switches from our Network Topology into PacketFence Web-based Administrator GUI, the switch page presented us with the list of nodes (devices) that were connected through the switch as shown in Figure 6.26 below. This enables better management and distinction of endpoint devices as each switch only presents the list of devices connected through it. Furthermore, this allows for identifying what devices are connected through which switch which would be more efficient for larger enterprise networks.

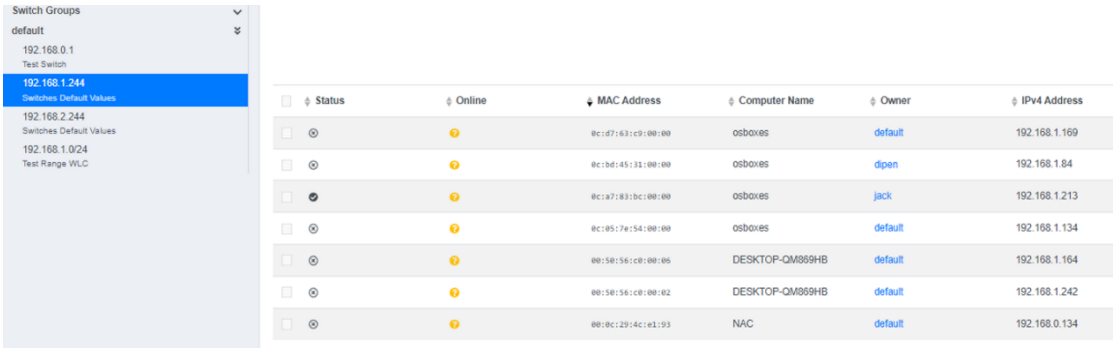

FIGURE 6.26: Switch Node detection

This enabled us to manage our end devices through the switch in the PacketFence Web-based Administrator GUI which provided us with the ability to perform various actions

such as reevaluating device access, editing owners, and performing device information audits as seen in Figure 6.27 below.

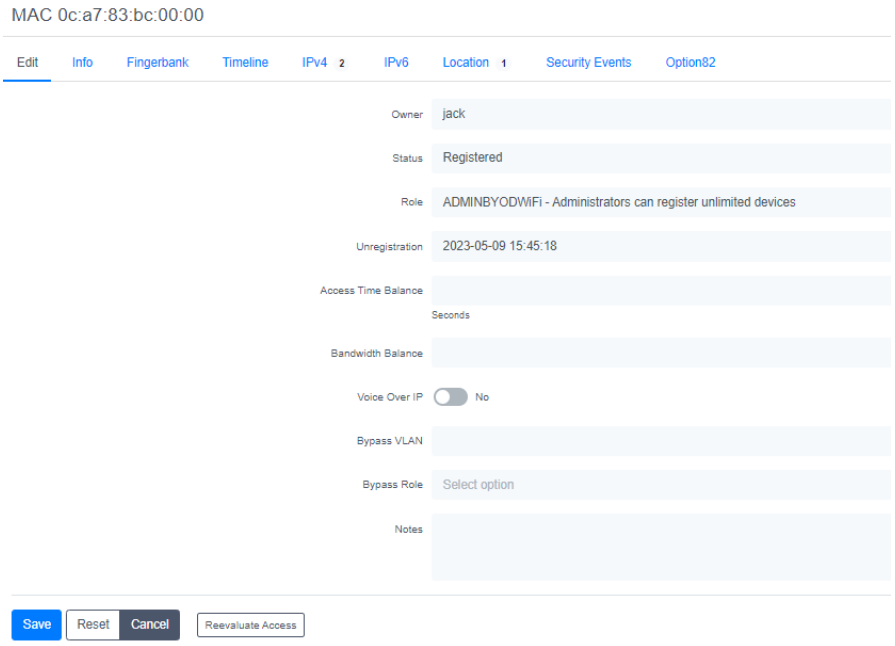

FIGURE 6.27: Management of Devices through the Switch

6.4 Additional Tests performed within the Network Topology

6.4.1 Analysis of packets using Wireshark while performing Network Access Control Tests

Wireshark enabled us to examine the traffic flow between the PacketFence server and the endpoint device (Ubuntu Guest) where were able to analyze the end-to-end communication and data transmitted through various packets.

6.4.1.1 DHCP Packets

The DHCP service runs on ports 67 and 68 as identified in the packet information below. PacketFence DHCP server had distributed **"192.168.78"** IP address for the endpoint device that was used throughout the testing phases of the project. We looked at the process behind how this IP address is distributed by analyzing the DHCP packets flow between the device and PacketFence inline interface. This is the first type of

communication that is established by the Ubuntu Client and PacketFence server which is as follows:

- The Ubuntu Guest sent a discover request packet within the network to see if it can obtain an IP address through a DHCP server.
- The PacketFence DHCP server then responded with an IP address offer packet to the device.
- The Ubuntu Guest sent a DHCP request packet for that IP address offered by the PacketFence DHCP server.
- Lastly, the server sent a DHCP ACK packet to acknowledge that the IP address has been allocated to the Ubuntu Guest.

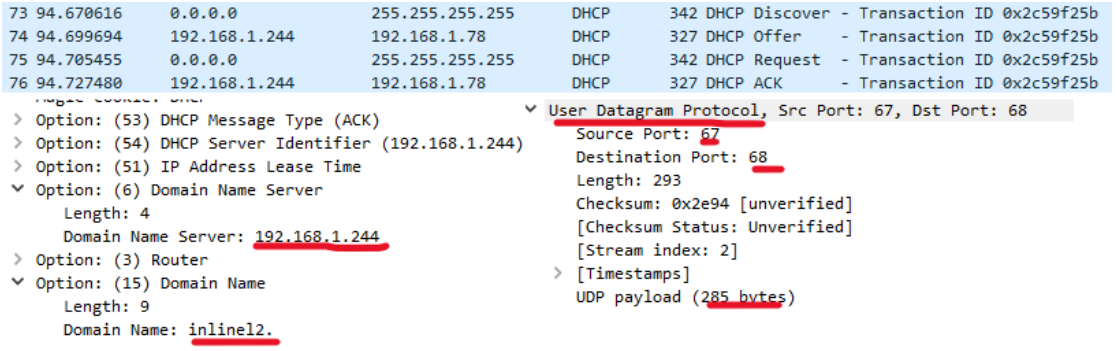

FIGURE 6.28: Wireshark DHCP Packets Analysis

6.4.1.2 TCP Packets

TCP (Transmission Control Protocol) is a secure method of transferring data between two devices through the use of PAR (Positive acknowledgment with re-transmission). We analyzed the initial TCP 3-Way handshake initiated between the Ubuntu Client and the PacketFence server **(66.70.255.147)** to get the Captive Portal HTTP web page when the local user has logged into the device. Other various TCP packets were initiated such as the GET **"nmcheck.gnome.org"** requests to check the internet connectivity within the Ubuntu client which was observed in the TCP stream. This is simply a host that NetworkManager uses to check for internet connectivity. The basic sequence of these TCP packets sent is as follows:

- The Ubuntu Guest sent an SYN (Synchronize sequence number) packet to the PacketFence server to notify the initiation of communication along with the sequence number.

- The PacketFence server then responds to Ubuntu Guest with an SYN-ACK signal bit set packet. SYN identifies the sequence number with which the packets are likely to begin and ACK (Acknowledgement) is the response to the packet received from the Ubuntu Guest.

- The third and last sequence of TCP 3-Way handshake ACK packet signifies the acknowledgment by the Ubuntu Client in regards to the response of the PacketFence server. At this stage, a secure and reliable connection is established where data transmission can be started. For example, HTTP GET request for the Captive Portal for users to log in to the web page URL is accessed which is highlighted in Figure 6.29 below.

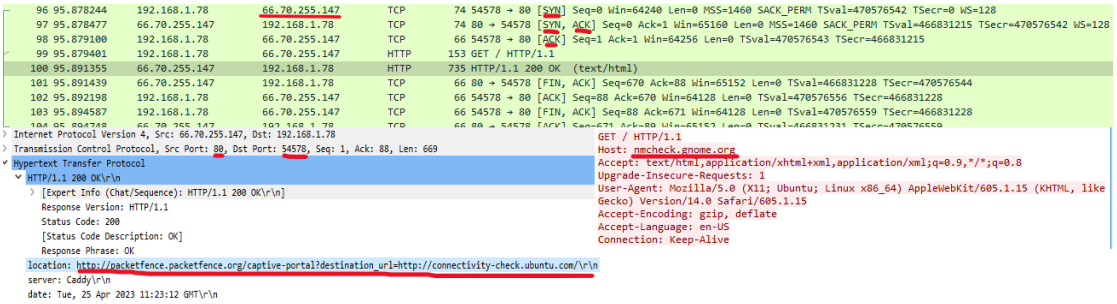

FIGURE 6.29: Wireshark TCP Packets Analysis

6.4.2 DHCP and DNS server for Windows Server

We verified that the Windows Server VM received an IP address through the DHCP pool from the router by running the [**sh ip dhcp binding**] command in CLI and it outputted the IP address that was provided from the DHCP pool which we can see in figure 6.30 below.

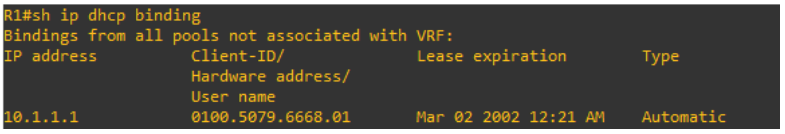

FIGURE 6.30: DHCP lease test

We also confirmed that our Windows Server 2016 received DNS server received an IP address with a simple [ipconfig /all] in the command prompt as shown in Figure 6.31 below.

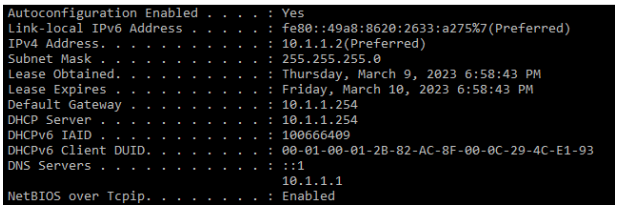

FIGURE 6.31: Windows Server DNS IP Address

6.4.3 Internet Connectivity within the network

We verified that there was internet connectivity by doing a ping test to 8.8.8.8 (DNS server of Google) in the router (R1) which we have provided a screenshot of in Figure 6.32 below.

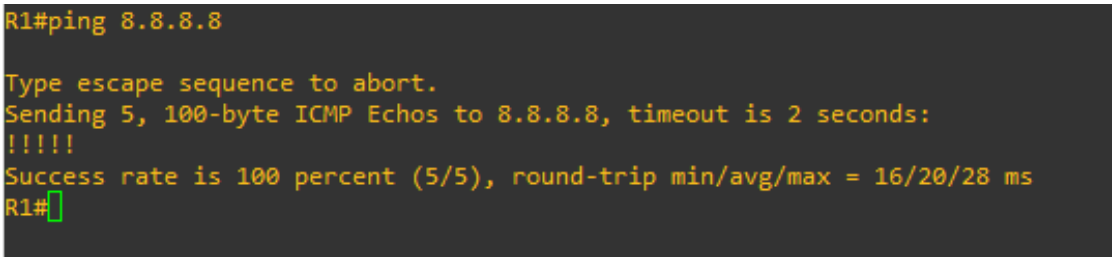

FIGURE 6.32: Ping test connectivity to internet

The last test we performed was to verify that our devices had internet connectivity, we ran a ping command to 8.8.8.8 once again on the Ubuntu client and Windows server VM.

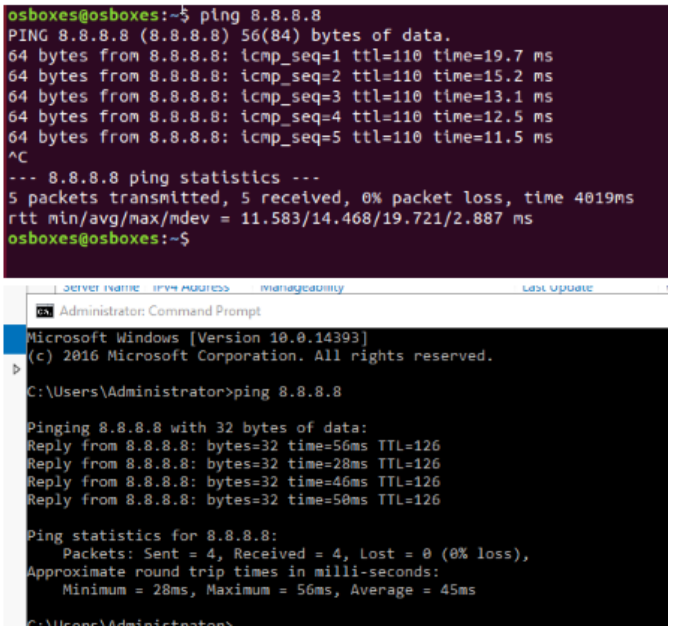

FIGURE 6.33: Internet Connectivity Test on Ubuntu and Win Server VM

6.4.4 Additional Non-Functioning PacketFence feature exercised

One of PacketFence's primary features is the ability to run vulnerability checks on newly registered devices. PacketFence provides options to integrate various third-party vulnerability scanners such as Nessus or OpenVAS. This feature was hoped to be exercised within PacketFence to have the capability to run scan operations while in the authorization and registration process of endpoint devices. This would allow for PacketFence to detect whether the registered device has a specific vulnerability that needs to be fixed before access to the network is granted. We already had a Kali Linux VM setup with Nessus vulnerability scanner installed and configured that was used for module Security Penetration Testing. Our plan was to import Kali Linux VM within the network topology in GNS3 and integrate it into the PacketFence but this was not functioning as we wanted it to. A new scan engine was created within PacketFence specifying the details of the host machine (Kali Linux) IP address and the account credentials for Nessus that was installed within the Kali Linux VM. Once this was created, we expected this newly created scan engine to enable vulnerability scanning for the endpoint devices upon registration but there had been no such feature enabled. We reached the point where this feature had to be left for future work due to the time constraint set for this project.

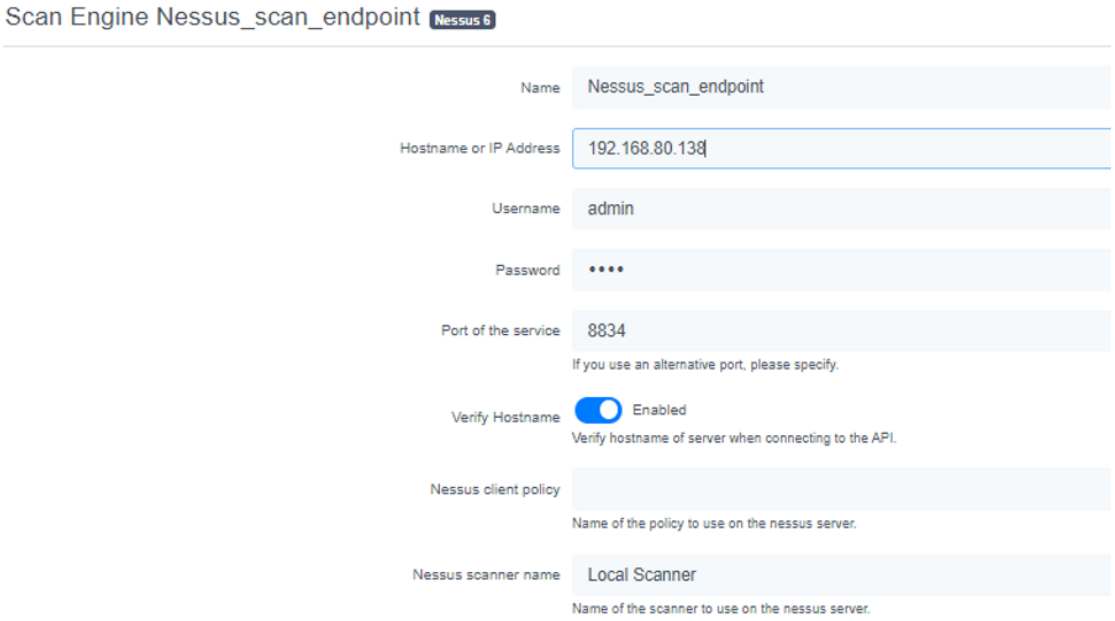

FIGURE 6.34: Nessus Scan engine created

Chapter 7

Discussion and Conclusions

7.1 Solution Review

The aim of the NAC solution is to manage endpoint device access to the network which is what we wanted to primarily cover in this project. NAC aims to provide the unified security that is desperately needed in today's network infrastructure in our opinion, and when correctly implemented, there are a variety of benefits that organizations can gain from NAC which we have discussed and analyzed throughout our project. The project scope defined at the start of the implementation phase accounted for quite an extensive workload with time constraints for the successful completion of the project. Despite various challenges that were encountered throughout the project, we were able to complete all of our implementation tasks bar some other functionalities such as integration of AD and vulnerability scanner that we did not manage to accomplish but this did not affect our primary goal of showcasing network access control.

Our project solution was divided up into two parts. The first task was to implement an open-source NAC solution called PacketFence into a virtual network environment to enforce network access control for endpoint devices. The second task was to provide an overview of NAC vendor's solutions along with a detailed technical analysis of each of their functionalities. Through these two parts of the project, we felt that our project provided the right balance between the theory and practical side of NAC. We put our focus on vendors such as Cisco NAC and Microsoft NAP for our project as we felt that these were the most popular NAC solutions out there in the market currently. We also added in PacketFence Open-source solution as part of this comparative overview which was appropriate as it gave us a better insight into its features and functionalities. More importantly, it gave us a detailed breakdown of the differences between paid versions versus open-source NAC solutions which were important to highlight for our project.

The integration of PacketFence NAC in a network environment was achieved through the use of the GNS3 network simulator which allowed us to build a virtual own network topology in our local machine. Our solution diverted from configuring solely just the PacketFence and Active Directory through the use of Windows Server 2016 to incorporating all of our components tools into one single network topology. This was quite an ambiguous goal due to the limited computing resources that were available to us so we had to evaluate our components for the topology carefully to match our available resources. This forced us to allocate RAM for some VM components of our topology well below the recommended amount to be able to run everything at once while still maintaining our computing resources which was quite a tricky task. Furthermore, it limited us to implementing our network topology in GNS3 to the bare minimum where we could only afford to run only compulsory devices to showcase the primary solution which was the testing of the Authorization and Registration features of NAC.

The documentation of our solution was given extra importance throughout the report as we wanted to showcase the work performed and knowledge gained throughout the implementation phase. We documented the configuration process of each component used in our network infrastructure to enable this report to act as a guide for readers. This project touched on various subsets of computer science that were required to successfully complete this implementation which was mainly networking, security, and database.

7.2 Project Review

The scope defined at the start of the implementation was quite large which was a challenge by itself given the time constraints of our project. We learned from our research phase that implementation of NAC was not an easy task, especially when working with an open-source tool where there is no guarantee and support available. We also knew that there was a large amount of learning involved while working through this project, especially with the plan of utilizing GNS3 to run a virtual network simulator. Despite these difficulties and complexity we successfully achieved our goal of integrating all of our components to create a virtual network infrastructure and enforce network access control for endpoint devices.

The implementation of NAC seemed like quite a challenging task to begin with due to the lack of research papers and guides online, especially for an open-source NAC PacketFence. While PacketFence provided an official guide on their site, we felt that it was very vague that lacked clear-cut instructions for proper implementation. The initial phase of PacketFence configuration for us was almost like a trial and error which consumed a lot of our time. It is recommended that one should consult an IT professional to

effectively integrate NAC into their network environment as it requires good knowledge of networking and security concepts. NAC solutions cannot function on their own, it requires other components to bring out their full capacity which meant that we had to acquire and establish a connection with other devices to PacketFence.

Our implementation compromised various phases of tasks of installation and configuration of various tools that were difficult to get functioning which added extra workload that affected our scheduled timeline. While Open-source tools are available for free of cost and were an integral part of our project, there were also many downfalls of utilizing them especially when we ran into errors. During the installation and configuration processes of our tools like PacketFence and GNS3, we encountered many errors which we had to debug ourselves as there were very few solutions available from online resources which were very time-consuming and frustrating.

Lastly, we want to highlight the importance of creating snapshots of each Virtual Machine on a regular basis to capture the completed configurations states, and data. A snapshot of a VM records the state and data of the virtual machine at the point in time the snapshot of taken. We created snapshots of PacketFence VM and GNS3 VM at each stage of our configuration as a risk precaution. We encountered a few failures of PacketFence VM throughout the implementation phase where we utilized the snapshot feature in such events.

7.3 Conclusion

This project's main goal was to provide insight into Network Access Control as a mechanism for applying a degree of security to a network's access and resources. Throughout the course of our research and implementation phases of this project, we have been able to make various conclusions on NAC as to whether it is an effective security solution that should be implemented within a network infrastructure.

Primary Conclusions

PacketFence ZEN NAC

PacketFence ZEN was the primary open-source NAC solution that was used to demonstrate the concept of Network Access Control within a simple network infrastructure. The vast number of features provided by PacketFence was quite impressive considering the fact that this is an open-source solution making it an excellent tool to demonstrate the topic of NAC. There was an initial learning curve during the configuration process at the beginning due to the lack of documentation published online but it proved

to be a rewarding experience due to its capabilities which we explored throughout the implementation phase. We were limited to exploring only the core features of the PacketFence NAC solution which was mainly the integration of network access control for the endpoint devices within our network due to time constraints and limited computing resources. The Inline enforcement method for PacketFence implementation proved to limit us to only the demonstration of authorization and registration of our endpoint devices and switches in the network.

While we wanted to explore and configure other PacketFence features such as Active Directory integration and network vulnerability scanner (Nessus), we were not able to incorporate these into our project within the constrained time. We concluded that one of the downfalls while implementing and working with the PacketFence tool was that it lacked proper documentation and online resources to be able to successfully explore its features. We allocated quite a large portion of our project time to debugging issues which overall affected our project schedule. Implementing PacketFence as a solution is a challenging task in itself as in a real production environment, it usually takes months to properly integrate within the networking infrastructure.

PacketFence's capacity to handle registration and remediation is a vital factor in displaying important NAC solution capabilities. Its ability to run compliance and health scans makes it an excellent solution for evaluating the status of endpoint devices within the network. Based on this implementation and overall study of NAC, it was concluded that PacketFence truly excels in providing many features that can be configured or integrated. However, it is vital to note that this solution would not be considered an appropriate testing method in terms of simplicity of use and implementation. PacketFence configuration was a challenging task by itself considering the time and effort required for the overall setup. It may seem to be difficult to set up as there is no straightforward method for enforcing PacketFence, this will largely depend on the requirement of individuals' network infrastructure. It was clear from this implementation that the setup took a significant amount of trial and error for the network configuration, necessary equipment, and VM configurations. However, once the initial hurdle was completed, it was evident that PacketFence provides a heavily controlled management interface. While navigating through the administrator GUI, it was clear to identify that the application contained well-managed and exceptional administrative functionalities. As an administrator, we felt that we had complete control over the network and our endpoint devices.

The most important NAC capability showcased was the endpoint device authorization and registration mechanism within a network. This was not as simple to implement as the PacketFence administrator guide claims but once configured correctly, PacketFence evoked vital components of NAC. Lastly, this implementation asserted that a

complete version of PacketFence could be integrated within a much larger scale network environment to bring out the full capabilities of Network Access Control.

NAC vendors solutions

From our technical overview of NAC vendors, we evaluated that these solutions lack effective post-admission control. The three solutions that we discussed in our project are pre-admission control-based and lack the default post-admission control capacity. Post-admission control refers to the monitoring of network devices for any malicious activities. Currently, post-admission control is accomplished by software support, which involves integrating solutions from third-party providers, such as, Threat management in Cisco's NAC is handled by security agent software. This is also the case with the Microsoft NAP and PacketFence where integration with other third-party solutions to achieve post-admission control.

We also concluded that the NAC market is oversaturated with technology that can be utilized to solve aspects of the NAC visions and some aspects of lack of interoperability. As a result, we view the NAC market as highly immature and the standardization movement for this market may eventually die down. Aside from the interoperability issues of the various technologies, an implementation must involve the collaboration of networking and security professionals with expertise in network administration, security, endpoint management, and authentication services. When one technology or group evolves, ripples are created down the line therefore management of compatibility is difficult. In the current state, the sheer number of people and device connections has increased by an order of magnitude which makes policy management a tremendous task. For large organization networks, ensuring granular access within the network might necessitate millions of access-control lists. The fact that NAC solutions are priced based on the number of connected devices, the initial investment, as well as continuing operations and maintenance may deem to be expensive in comparison to other solutions.

During the examination of these NAC vendor solutions, not a single vendor seems to offer a real-time backup strategy for a NAC failure. Any component failure that occurs in real-time should be accommodated by NAC. As NAC has a variety of architectural and functional components, debugging could be a difficult task as it can be hard to determine the root cause of the errors.

Secondary Conclusions

Project Solution Approach

We encountered various challenges within our project solution approach where we had to propose some adjustments to achieve our desired solution. Our initial plan to integrate Active Directory into PacketFence from the research phase was not successful so some alterations were made. Our revised solution of creating users and groups in the PacketFence Admin GUI web interface for enforcing network access control worked well overall. This process worked seamlessly as it reduced the timeframe from a user management perspective and allowed us to put more focus on our main goal of enabling access control.

Applicability of PacketFence into Physical Network

This project highlighted that PacketFence can be an excellent security solution for any network infrastructure, especially for the management of end-point devices. Implementing any type of NAC solution can be a difficult task as there is a process of integrating your existing security solutions with it to utilize the full benefits. This process can take some time depending on the amount of integration required and the ability of the team to enforce the NAC solution. Additionally, teams from various departments such as networking, security, and administration must collectively work together to bring out the full effect of NAC solutions. This is mainly because defining and establishing a unified policy within the NAC platform is a difficult task that necessitates strong collaboration among administrative teams.

GNS3

This open-source emulator software allowed us to integrate every component of our project into one Network Topology and allowed us to import IOS images of Cisco hardware (Routers and switches). This tool was quite new to us as we had not used it before so there was an initial learning curve at the start but we had allocated enough time from our project schedule to explore this tool. Utilizing the GNS3 VM would be recommended due to its lightweight and flexible way of creating topologies and it will consume less computing resources than importing VMs. GNS3 served as a crucial tool for our implementation of NAC that enabled us to experiment with various features of PacketFence. GNS3 facilitates the possibility of building and testing network infrastructures in a virtual environment without the need for any physical equipment. This enables exercising and building knowledge on a multitude of concepts in regard to the networking and security field within the topic of computer science.

7.4 Future Work

Integration of Active Directory in PacketFence

This was one of the proposed solutions from the research phase that we were not able to achieve in our implementation. We could pin down the root cause and allocate more time to come up with a resolution for future implementations. We suspect that there could be some issues with the DNS server not being able to recognize the PacketFence server but more troubleshooting steps could be followed up and given more time.

Integration of Vulnerability Scanner (Nessus)

This was a feature that we hoped to enforce and examine in order to add an additional security layer that would complement the feature of authorization and registration via the captive portal. Adding in a scan engine such as the Nessus vulnerability scanner within PacketFence would scan each endpoint device before network access is granted and provide an alert if any vulnerabilities were detected in the device. We were not able to provide this feature in this implementation due to errors but we believe this can be achieved in future work with more time allocation.

VLAN Enforcement

This type was not feasible for this project due to not being able to obtain compatible Cisco networking equipment (Switch) that supported the required commands to enable features of VLAN Enforcement. For future PacketFence NAC implementations, VLAN enforcement should be considered as the focal point as this method brings out the best of PacketFence. VLAN enforcement would enable administrators to implement more features such as remediation, MAC detection, isolation, quarantine, etc.

Increased Computing Resources

Our Network Topology was limited due to the lack of computing resources and somewhat affected our ability to showcase other features such as the integration of security measures and exercising some network attack scenarios. This project can be used as a stepping stone to building larger Network Topologies within GNS3 with increased computing resources.

Improved Network Configurations in Networking Equipment

As our project mainly revolved around testing the capabilities of the PacketFence NAC solution, the focus on network configuration was not heavily practiced. However, this could be something that can be explored for future work with more configurations such

as implementing network segregation within the network topology in regard to improved management of the network.

7.5 Project Management

We defined the Trello board as the project management tool in the research phase. This tool enabled enhanced planning of the implementation phase where each task was identified and executed for the completion of the project overall. Tasks that are completed in the duration of the implementation were marked with green labels and tasks that were not achieved are marked with a red label as shown in Figure 7.1 below.

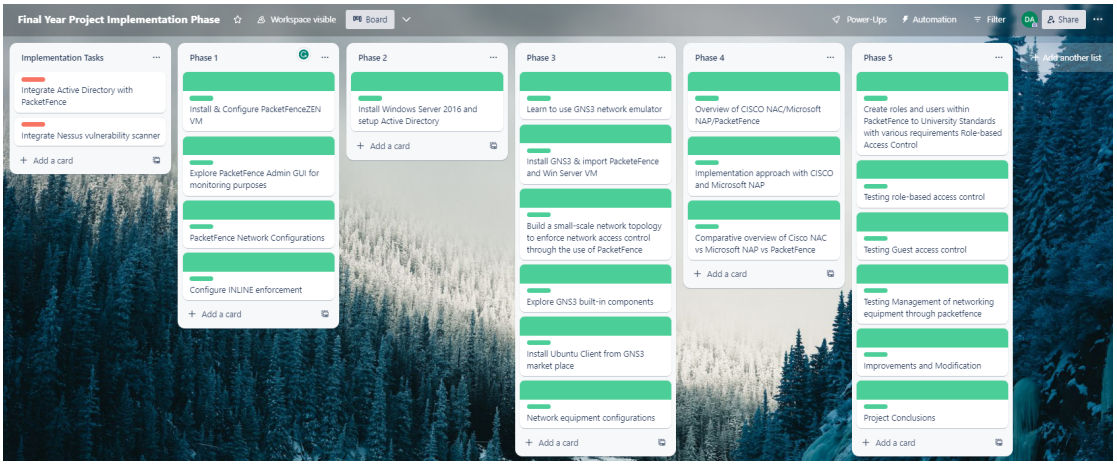

FIGURE 7.1: Trello Board for Project Management

Bibliography

[1] J. L. Yusuf Adewale, Vilhelm Wareus, "Nac implementation for end point devices," *NAC*, vol. 5, no. 7, pp. 1–10, June 2010. [Online]. Available: http://www.diva-portal.org/smash/get/diva2:346438/FULLTEXT01

[2] Cynet, "Network attacks and network security threats," *Network Attacks and Network Security Threats*, vol. 5, no. 5, pp. 1–10, Nov 2014. [Online]. Available: https://www.cynet.com/network-attacks/network-attacks-and-network-security-threats/

[3] C. Nationwide, "Getting endpoint security right in 2022," *Getting Endpoint Security Right in 2022*, vol. 5, no. 6, pp. 1–10, May 2022. [Online]. Available: https://computersnationwide.com/getting-endpoint-security-right-2022/

[4] P. I. LLC, "The third annual study on the state of endpoint security risks," *NAC*, vol. 5, no. 8, pp. 1–10, Jan 2020. [Online]. Available: https://www.morphisec.com/hubfs/2020

[5] N. Sharma, "Network access control (nac)," *NAC Architecture*, vol. 3, no. 3, pp. 1–10, Nov 2013. [Online]. Available: https://www.helpnetsecurity.com/2007/11/26/network-access-control-nac/

[6] S. N. N. Rafique Agyare, Christian Adu-Boahene, "Secure remote network management and network access control, the case of university of education-kumasi campus," *International Journal of Systems Engineering*, vol. 72, no. 1, pp. 4477–4479, May 2022. [Online]. Available: https://www.researchgate.net/profile/Solomon-Nikoi-4/publication/361792806_Secure_Remote_Network_Management_and_Network_Access_Control_the_Case_of_University_of_Education-kumasi_Campus/links/62c5669f01adca54aa63cad7/Secure-Remote-Network-Management-and-Network-Access-Control-the-Case-of-University-of-Ed
pdf

[7] H. U.-D. Qazi, "Network access techonologies," *NAC*, vol. 5, no. 13, p. 1, Sept 2014. [Online]. Available: http://www.diva-portal.org/smash/get/diva2:23688/FULLTEXT01.pdf

[8] K. L. Lueth, "State of the iot 2020," *IoT*, vol. 5, no. 9, p. 1, Nov 2020. [Online]. Available: https://iot-analytics.com/state-of-the-iot-2020-12-billion-iot-connections-surpassing-non-iot-for-the-first-time/

[9] Purplesec, "Cyber security trends in 2021," *Cybersecurity*, vol. 5, no. 10, p. 1, Apr 2021. [Online]. Available: https://purplesec.us/cyber-security-trends-2021/

[10] Kaspersky, "Cyber security trends in 2020," *Cybersecurity*, vol. 5, no. 11, p. 1, Dec 2020. [Online]. Available: https://www.kaspersky.com/about/press-releases/2020_the-number-of-new-malicious-files-detected-every-day-increases-by-52-to-360000-in-2020

[11] B. Schwartz, "Packetfence administration guide," *PacketFence*, vol. 5, no. 12, p. 1, Sept 2014. [Online]. Available: https://www.packetfence.org/doc/PacketFence_Installation_Guide.html#_supported_enforcement_modes

[12] GNS3, "Gns3 open-source network emulator," *GNS3 Documetation*, vol. 5, no. 13, p. 1, Jan 2022. [Online]. Available: https://docs.gns3.com/docs/

[13] PacketFence, "Packetfence advanced features," *NAC*, vol. 5, no. 13, p. 1, Jan 2022. [Online]. Available: https://www.packetfence.org/about.html#/features

[14] C. NAC, "Cisco nac," *Cisco*, vol. 5, no. 13, p. 1, Jan 2023. [Online]. Available: https://www.cisco.com/c/en/us/products/security/what-is-network-access-control-nac.html

[15] D. Popescu, "Nap enforcement," *NAP*, vol. 5, no. 13, p. 1, Jan 2014. [Online]. Available: https://www.poweradmin.com/blog/nap-enforcement-network-access-protection/

Appendix A

Code Snippets

A.1 DHCP Pool Creation Commands:

```
R1(config)#ip dhcp pool NAC
R1(dhcp-config)#network 10.1.1.0 255.255.255.0
R1(dhcp-config)#default-router 10.1.1.244
R1(dhcp-config)#exit
R1(config)#
```

Figure A.1: DHCP Pool configuration in R1

A.2 Router-on-stick Configuration Commands:

```
ESW1#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
ESW1(config)#vlan 10
ESW1(config-vlan)#name endpoint
ESW1(config-vlan)#exit
ESW1(config)#vlan 20
ESW1(config-vlan)#name registration
ESW1(config-vlan)#exit
ESW1(config)#vlan 30
ESW1(config-vlan)#name isolation
ESW1(config-vlan)#exit
ESW1(config)#vlan 40
ESW1(config-vlan)#name management
ESW1(config-vlan)#exit
ESW1(config)#int vlan 40
ESW1(config-if)#ip add 192.168.99.3 255.255.255.0
ESW1(config-if)#no shut
ESW1(config-if)#exit
ESW1(config)#ip default-gateway 192.168.99.1
ESW1(config)#int f1/1
ESW1(config-if)#switchport mode access
ESW1(config-if)#switchport access vlan 20
ESW1(config-if)#no shut
ESW1(config-if)#int f1/0
ESW1(config-if)#switchport mode trunk
ESW1(config-if)#no shut
*Mar  1 00:04:07.099: %DTP-5-TRUNKPORTON: Port Fa1/0 has become dot1q trunk
*Mar  1 00:04:07.599: %LINEPROTO-5-UPDOWN: Line protocol on Interface Vlan40, changed state to up
ESW1(config-if)#no shut
ESW1(config-if)#exit
ESW1(config)#end
ESW1#ci
*Mar  1 00:04:20.079: %SYS-5-CONFIG_I: Configured from console by console
ESW1#copy running-config startup-config
Destination filename [startup-config]?
Building configuration...
[OK]
ESW1#
```

FIGURE A.2: ESW1 Router-on-stick Configuration

```
ESW2#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
ESW2(config)#vlan 10
ESW2(config-vlan)#name end
*Mar  1 00:00:24.191: %SYS-3-CPUHOG: Task is running for (2028)msecs, more than (2000)msecs (1/0),process = Exec.
-Traceback= 0x60A67DE0 0x60A8A61C 0x60A8A9F4 0x60BA8B10 0x60BA984C
ESW2(config-vlan)#name endpoint
ESW2(config-vlan)#exit
ESW2(config)#vlan 20
ESW2(config-vlan)#name registration
ESW2(config-vlan)#exit
ESW2(config)#vlan 30
ESW2(config-vlan)#name isolation
ESW2(config-vlan)#exit
ESW2(config)#vlan 40
ESW2(config-vlan)#name management
ESW2(config-vlan)#exit
ESW2(config)#int vlan 40
ESW2(config-if)#ip add 192.168.99.2 255.255.255.0
ESW2(config-if)#no shut
ESW2(config-if)#exit
ESW2(config)#ip default-gateway 192.168.99.1
ESW2(config)#int f1/1
ESW2(config-if)#switchport mode access
ESW2(config-if)#switchport access vlan 30
ESW2(config-if)#no shut
ESW2(config-if)#exit
ESW2(config)#int f1/0
ESW2(config-if)#switchport mode trunk
ESW2(config-if)#no shut
*Mar  1 00:03:35.203: %DTP-5-TRUNKPORTON: Port Fa1/0 has become dot1q trunk
*Mar  1 00:03:35.703: %LINEPROTO-5-UPDOWN: Line protocol on Interface Vlan40, changed state to up
ESW2(config-if)#no shut
ESW2(config-if)#exit
ESW2(config)#end
ESW2#copy ru
*Mar  1 00:03:56.519: %SYS-5-CONFIG_I: Configured from console by console
ESW2#copy running-config startup-config
Destination filename [startup-config]?
Building configuration...
[OK]
ESW2#
```

FIGURE A.3: ESW2 Router-on-stick Configuration

```
ESW3#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
ESW3(config)#vlan 10
ESW3(config-vlan)#name endpoint
ESW3(config-vlan)#exit
ESW3(config)#vlan 20
ESW3(config-vlan)#name registration
ESW3(config-vlan)#exit
ESW3(config)#vlan 30
ESW3(config-vlan)#name isolation
ESW3(config-vlan)#exit
ESW3(config)#vlan 40
ESW3(config-vlan)#name management
ESW3(config-vlan)#exit
ESW3(config)#int vlan 40
ESW3(config-if)#ip add 192.168.99.4 255.255.255.0
ESW3(config-if)#no shut
ESW3(config-if)#exit
ESW3(config)#ip default-gateway 192.168.99.1
ESW3(config)#int range f1/2 - 5
ESW3(config-if-range)#switchport mode access
ESW3(config-if-range)#switchport access vlan 10
ESW3(config-if-range)#no shut
ESW3(config-if-range)#int f1/1
ESW3(config-if)#switchport mode trunk
ESW3(config-if)#no shut
*Mar  1 00:17:15.659: %DTP-5-TRUNKPORTON: Port Fa1/1 has become dot1q trunk
*Mar  1 00:17:16.159: %LINEPROTO-5-UPDOWN: Line protocol on Interface Vlan40, changed state to up
ESW3(config-if)#no shut
ESW3(config-if)#exit
ESW3(config)#end
ESW3#
*Mar  1 00:17:22.195: %SYS-5-CONFIG_I: Configured from console by console
ESW3#copy running-config startup-config
Destination filename [startup-config]?
Building configuration...
[OK]
ESW3#
```

FIGURE A.4: ESW3 Router-on-stick Configuration

```
R1#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)#int f1/0.10
R1(config-subif)#description default gateway for vlan 10
R1(config-subif)#encapsulation dot1Q 10
R1(config-subif)#ip add 192.168.10.1 255.255.255.0
R1(config-subif)#exit
R1(config)#int f0/1.20
R1(config-subif)#description default gateway for vlan 20
R1(config-subif)#encapsulation dot1Q 20
R1(config-subif)#ip add 192.168.20.1 255.255.255.0
R1(config-subif)#exit
R1(config)#int f0/1.30
R1(config-subif)#description default gateway for vlan 30
R1(config-subif)#encapsulation dot1Q 30
R1(config-subif)#ip add 192.168.30.1 255.255.255.0
R1(config-subif)#exit
R1(config)#int f0/1.40
R1(config-subif)#description default gateway for vlan 40
R1(config-subif)#encapsulation dot1Q 40
R1(config-subif)#ip add 192.168.40.1 255.255.255.0
R1(config-subif)#exit
R1(config)#int f1/0
R1(config-if)#description trunk link to PF
R1(config-if)#no shut
R1(config-if)#exit
R1(config)#int f0/1
R1(config-if)#description trunk link to AD
R1(config-if)#no shut
R1(config-if)#exit
R1(config)#
```

FIGURE A.5: R1 Router-on-stick Configuration

Appendix B

Wireframe Models