An Empirical Study of Hardening Network Access Control Systems

Kalim Qureshi*
Department of Information Science ,Kuwait University, Kuwait
Mohsen Al-Shamali †
Department of Information Science ,Kuwait University, Kuwait.
Mostafa Abd-El-Barr‡
Former-Dean College of Computing Science and Engineering, Kuwait University, Kuwait.

Abstract

Network Access Control (NAC) is one of many solutions that plays a critical role in defining security policies in networking. Three open-source NAC solutions were analyzed and compared: OpenNAC, FreeNAC, and PacketFence. The results showed that the PacketFence solution has better performance in terms of security features. Network layer-2 attacks were introduced against the candidate solution to verify vulnerabilities. These are Cisco Discovery Protocol, Dynamic Host Configuration Protocol, Spanning Tree Protocol, Dynamic Trunking Protocol, and VLAN Trunking Protocol. An enhanced PacketFence was proposed to mitigate network threats in a simulated environment; by using the network simulator tool (GNS3) and through hardening a critical component of PacketFence via applying supportive configurations and commands. We observed that the proposed enhancement solution improved network security. This is measured in terms of 22% to 84% increase in the CPU utilization during an attack that lasted for 10 minutes. In addition to root cost increase from 0 to 12 after launching 3 STP attacks. This is a substantial surge in MAC address table entries. Interface status was also changed to trunk and the VLAN entries were manipulated either by adding or removing entries in the VLAN table.

Key Words:Network Access Control (NAC); Network Security; PacketFence; Policy Enforcement Point; Hardening Configuration; Security performance improvements.

1 Introduction

The demands for Information Security has recently increased to a level that demanded every firm to have a dedicated team responsible for identifying information vulnerabilities to mitigate diverse threats encountered. Access control is a security technique that is used to organize the accessibility of assets and resources. NAC delivers endpoint protection, access control, and performance monitoring, authentication, and network security enforcement as shown in Fig. 1. As shown in the Figure 1, the NAC solution consists of three major components. The first component is a policy decision point (aka Radius Server) which acts as a policy repository and authenticator using Authentication/Authorization/Accounting model (AAA). Secondly, the policy enforcement point which is a network switch that communicates with a radius server to manage the accessibility to the network's resources.

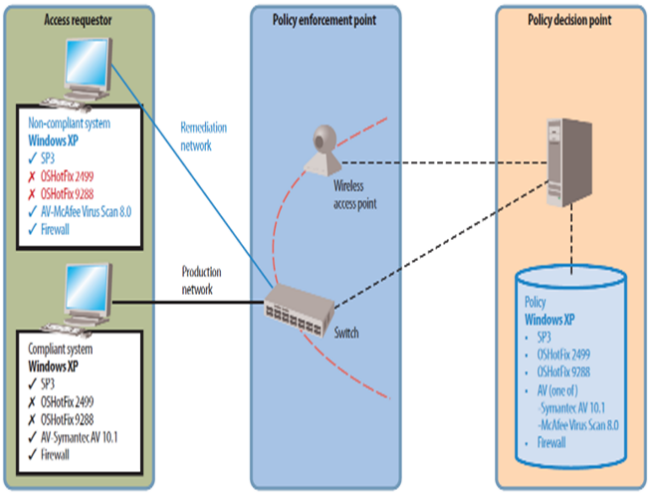


Figure 1: NAC System Components.

Lastly, an endpoint agent is implemented on endpoint devices to check the needed requirements based on pre-defined policies. Network access control compels all network users to adapt to its automated directives aimed to protect the network from security threats. With NAC system deployed, enforcing any anti-malware software on endpoint nodes is a compliment. Hence, the network will automatically take over the task of preventing malware attacks. Data theft and the desire to cause disruption are among the reasons why some attackers raid network access control. Examples of attacks purposed for stealing data are Denial of Service (DoS) and Address

*Department of Information Science ,Kuwait University, Kuwait. Email: :kalimuddinqureshi@gmail.com
†Department of Information Science ,Kuwait University, Kuwait. Email: :kalimuddinqureshi@gmail.com
‡Former-Dean College of Computing Science and Engineering, Kuwait University, Kuwait. Email: : mohsen.alshamali@ku.edu.kw

Resolution Protocol (ARP) spoofing. If a good defiance strategy is deployed, it will make the network impermeable to many of the arms brought about by these attacks [10], [6]. In this paper, an enhanced version of PacketFence is introduced by re-configuring the policy enforcement point (Cisco Switch) with the best practice guidelines provided by Cisco [4] to reinforce the NAC solution. A certain type of attack has been deployed over the simulated network using GNS3 and Yersinia attacking tool [10]. In this work, CDP attacks and root-claim attacks are introduced which show a vulnerability in a major part of the network access control system (unmodified PacketFence), also known as an enforcer device.

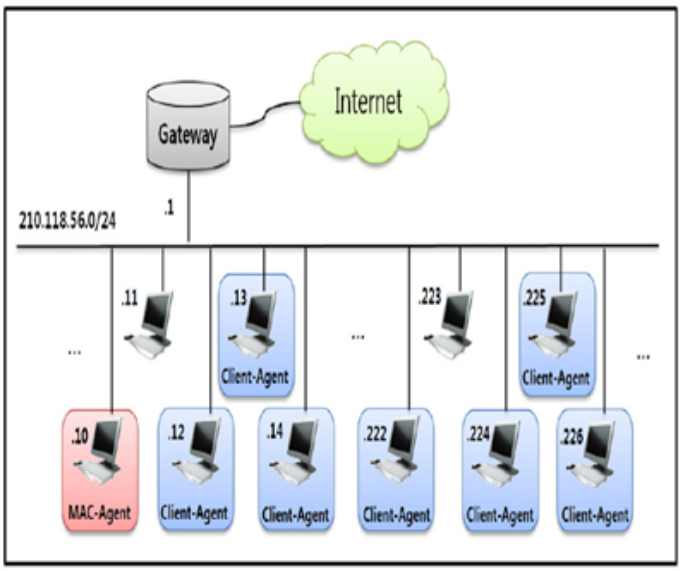


Figure 2: ARP spoofing suggested prevention method.

The paper is organized as follows: the literature review is presented in Section 2. In Section 3, NAC Tools, and an enhanced version of PacketFence is presented by configuring the policy enforcement point along with the implementation. The core setup and the tools used tobuild a network simulation are presented in Section 4. Experimentation and evaluation results are shown in Section 5, along with the performance metrics used. Finally, a closure of the research and future works is presented in Section 6.

2 Literature review

The study in [5] presents a method to lower the rate of attacks based on ARP spoofing. Along with that, a way to include stronger security measures for basic control systems without any additional cost is discussed. The method requires no changes of the required protocol or any extra appliances to prevent ARP spoofing. It only requires an actual detached PC with a MAC-Agent and a Client-Agent in each PC in the network, as shown in Fig. 2. The study in [7] points out consideration of the standards of conduct of the organization and its clients. It conveys an improved organization access control utilizing free BSDpfSense open-source systems. It uses the a committed edge firewall with the presentation of squid, squidGuard, Squid Analysis Report Generator (SARG), as well as the establishment of an Active Directory worker with client access arrangements to improve client access control and protect the LAN from abuse, virus attacks, and unauthorized entries. As a result of this study, two main problems were observed. The first is the limited bandwidth, and the second is the absence of a reliable network access control, which left the UMaT network exposed to all types of attacks such as DoS/DDoS attacks, Worms, Trojans attacks, etc. By applying a proxy server (Squid) which limits access to the network, the UMaT network has been improved, and network vulnerability has been greatly reduced.

The study in [3] contains the evaluation of the security provided by the PacketFence Network Access Control server installed within the GNS3 emulator. To prove that the attacks on the network are real, open-source software called Yersinia has been used to conduct various experiments. As a result, the study showed that the PacketFence system is vulnerable, and the attacks are viable and realistic, which may lead to a considerable cost during the attacks for the company. Additionally, the study discussed various types of attacks on the network; for instance, CDP attack, MAC Flooding attack, Authentication attack, and ways to secure the network against them.

Open NAC solutions are hybrid solutions composed of software and hardware components. The most popular open-source NAC solutions are cited in [10]. A number of researchers use open-source tools for educational purposes due to the availability of code to community. OpenNAC(Opennac.org, 2021) is an open-source network access control system that offers access to LAN/WAN networks based on privilege rights and policies. OpenNAC also proposes a secure connection among devices on the network by using 802.1X authentication protocols based on LDAP or an active directory. Figure 3 provides several features of OpenNAC to manage the network. FreeNAC [10] is a GPL (General Public License) open-source network access control system. It is completely free and supports both wired/wireless network infrastructure. It serves all different types of network devices while focusing on information security when communicating with different devices on the network. It achieves this by applying security modes such as 802.1X, MAB (MAC-Authentication-Bypass), and VMPS (VLAN management policy server). FreeNAC is hard to perform a posture assessment and bandwidth monitoring. Therefore, it is relying on other tools, for example, security assessment tools, from the server side and bandwidth monitoring tools. Please refer toFig,3 for an architecture of the tool.

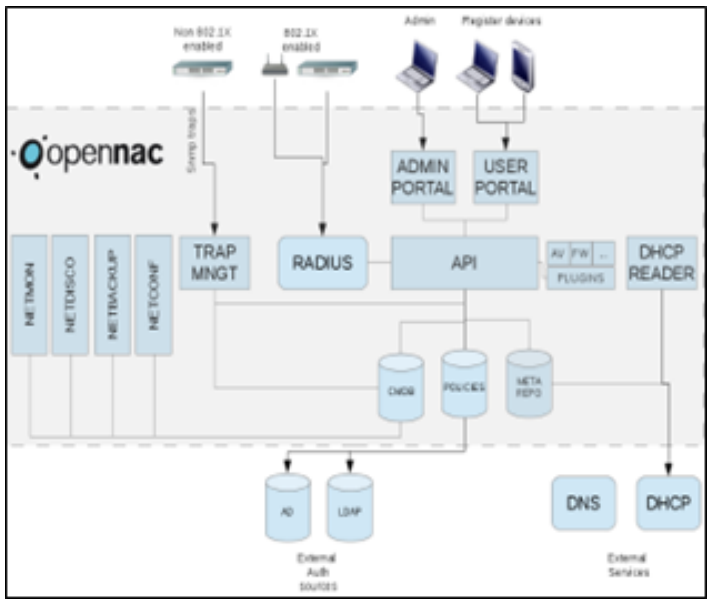


Figure 3: OpenNAC design architecture

PacketFence (packetfence.org, 2021) can be used in both wired/wireless networks with a unified management. It also uses a secure connection between the NAC elements in the network by using 802.1X, MAC-Authentication-Bypass (MAB), DHCP fingerprinting, and user agents which can be installed in the endpoint device. The system also uses a posture assessment which can be defined as a compliance verification to evaluate the endpoint device security perspective based on the pre-defined rules and policies in the PacketFence server. Moreover, it performs posture assessment using statement of health (SOH) protocols to collect the required data from devices in the network. PakcetFence can perform remediation through a captive portal and redirect the user to a different URL with a set of instructions for the specific situation to get access to the network. The basic implementation for deploying PacketFence consists of 3 major components, as shown in Fig.4.

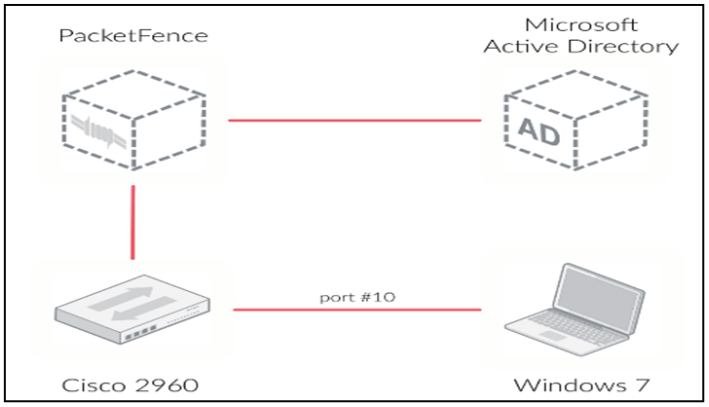


Figure 4: Basic implementation of PacketFence

The architecture design of PacketFence (see Figure 5) can analyse the traffic bandwidth and keeping track of it in case of any unusual or suspicious activity. Additionally, the system performs a set of actions to secure the network by quarantine or changing the access level of the device.

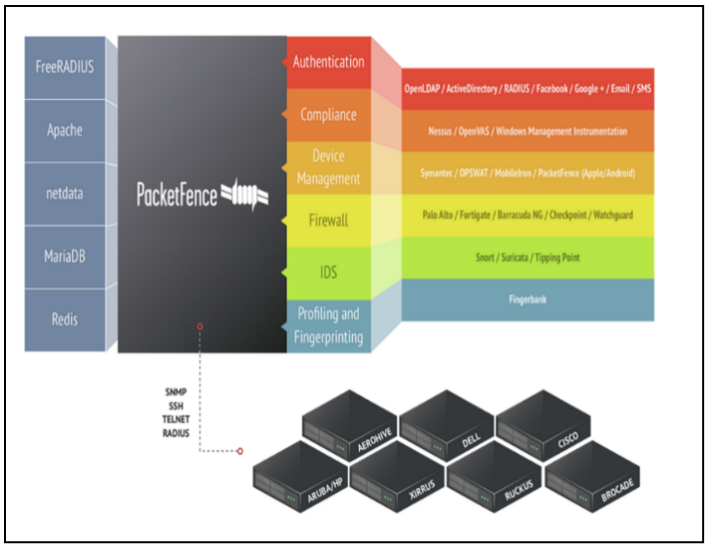


Figure 5: PacketFence - Component Architecture

Table 1 provides a summary comparison made based on a study paper reviewing open-source network access control tools for Enterprise educational networks [10]. The table consists of features that are mostly embedded with the security aspect of NAC tools.

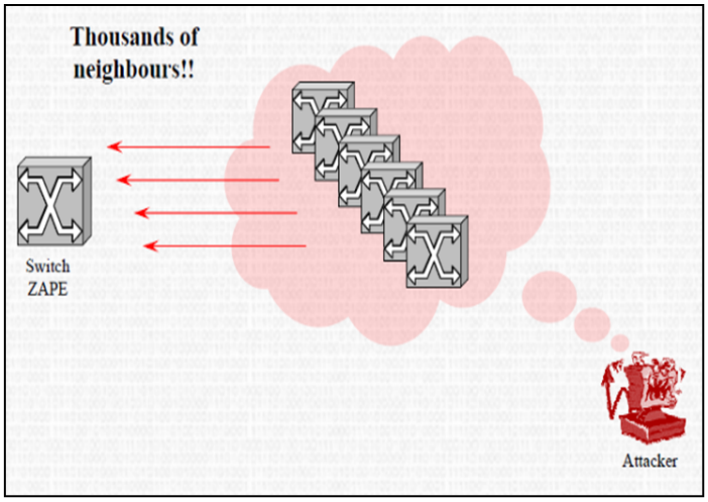


Figure 6: A basic illustration of CDP attack [6, 7]

From the table 1 and 2 it clear that PacketFence can manage the bandwidth, a robust posture analysis, multiple authentication protocols, and standards. While PacketFence succeeds other open-source solutions certain issues need to be tackled. In [3]. PacketFence is vulnerable to certain types of attacks; as a result, an eagle eye is needed to focus on this issue. Starting from

Table 1: A Comparison between open-source NAC security aspects

Feature	OpenNAC	PacketFence	FreeNAC
Tracking Bandwidth	Doesn't keep track of bandwidth usage	Keep track of bandwidth usage and takes action on any suspicious activity such as quarantine.	Relying on other Network monitoring tools to keep track of bandwidth usage
Authentication	802.1X based on LDAP protocol	MAB "MAC-Authentication bypass" 802.1X DHCP fingerprinting User agents	MAB "MAC-Authentication bypass" 802.1X, Cisco VMPS (Vlan management policy)
Posture assessment	Examine end devices' antivirus updates, OS updates, patches, and firewall	Use SOH protocol "Statement of health" to gather data from end-devices and analyze the posture	Unable to do posture assessment by itself, and relying on other security assessment tools from the server side to deploy posture assessment for end-devices
Wire/Wireless Networks	Supported	Supported	Supported
Design Scalability	Small/Medium Networks	Geographically Scalable	Small Networks

the results of this study, we introduce in the next section an enhanced version the PacketFence system.

3 The Proposed Enhancements to the PacketFence System

While PacketFence has the overall success among others, nevertheless, it has been shown that the system can fail due to certain vulnerabilities. Here, improving the security demands focusing on hardening a major component of the PacketFence system can be achieved by providing the configuration and implementation of the policy enforcement point (Network switch) where any attack against the switch can eventually lead to a system failure, fore more detail refer to Section 6.

Regarding the vulnerabilities on policy enforcement points, the following section will analyse the introduced attacks and prevention methods according to the best practice guidelines, as provided by the Cisco official website [4]. In the upcoming sub-sections, a set of attacks is exposed beside the resolving configuration for securing the policy enforcement point.

Table 2: A comparison of features between open-source NAC solutions

Features	OpenNAC	PacketFence	FreeNAC
Posture Analysis	Fairly	Yes	Not Inherent
Contrivance Authentication	Yes	Yes	Yes
Bandwidth Management	No	Yes	No
Network Vendor Support	Multivendor	Multivendor	Multivendor
Wired and Wireless Support	Yes	Yes	Yes
Software Integration	Yes	Yes	Commercial
Community Support	Active	Active	Fairly Active
Administrative Interface	Web Interface	Web Interface	Mainly Window-based
Reporting	Yes	Yes	Yes

3.1 CDP Attack

CDP (Cisco Discovery protocol) is a layer 2 protocol used in cisco devices which sends identification packets over the network in plain text. This protocol is usually used by network administrators for discovering neighbor devices and troubleshooting. The protocol is enabled by default in network devices such as routers, switches, and servers.

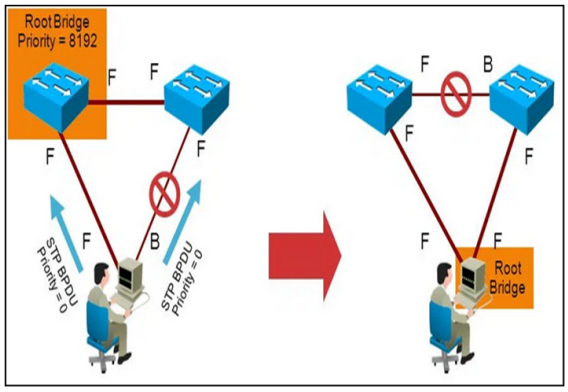


Figure 7: A basic illustration of STP attack (common attack types on switches, 2018)

3.2 STP Attack

Spanning Tree Protocol (STP) is a Layer 2 protocol that operates on switches. Primarily, STP is used to guarantee that you do not make loops when you have duplicate paths in your network; as a result, loops are fatal to the network stability.

Spanning Tree does not use multiple links to lead to the same destination. In addition, Spanning Trees is used in a network fault tolerance design in such way if one link dropped another backup link will take place to improve reliability and resilience. STP attacks (figure 7) focus on spoofing the root bridge in the network topology by broadcasting out a topology change enforcing an STP re-calculation. This also causes a DoS on the network by causing an interruption during the root bridge modifications, as shown in the below fig. 7.

In this kind of attack, mitigation can be attained by not using STP on unnecessary ports disable stp and using port security commands on the interface level. Moreover, the bridge protocol data unit (BPDU) guard bpdu guard command is used if the network is using a portfast feature in STP configuration. A command line root guard is mandatory to prevent the attacker to claim the root.

3.3 DHCP Starvation Attack

Dynamic Hosting Configuration Protocol (DHCP) is a layer 2 network management protocol based on a client/server model. The main purpose of DHCP is dynamically assigning an IP address to network devices by a set of requests between the network device and the server. As shown in Fig. 8 DHCP can be implemented on any network, regardless of the size of the network. It can work on layer 3 IP protocols when routers or gateways act like DHCP servers to receive globally unique IP addresses. DHCP Starvation Attack is a malicious attack often used to exhaust the DHCP server by sending numerous requests. This attack aims to stall the network by preventing legitimate devices from acquiring an IP address to gain access to the network. In this scenario, a prevention method is introduced by applying an ip dhcp snooping command on the switch, which drops undesirable DHCP traffic that may be requested from unauthorized DHCP servers [13].

3.4 DTP Attack

The Dynamic Trunking Protocol (DTP) is a layer 2 cisco proprietary protocol that can only be used between two cisco switches to negotiate a trunk link between them along with the encapsulation type. DTP is automatically enabled on a switch port by default when a certain trunking mode is configured on the switch port. There are two DTP trunking modes: first is dynamic desirable, and the other is Dynamic auto. For the first mode, the interface sends DTP packets constantly trying to establish the trunk link if possible. Meanwhile, in the dynamic auto mode, the interface just waits for DTP requests that are being sent. An attack can be launched to force interfaces that use DTP protocol to shift to trunk mode. This will leave the network vulnerable to various types of attacks such as traffic sniffing, man in the middle attacks, and VLAN hopping as a result of all VLANS being accessible for the attacker.

3.5 VTP Attack

VLAN Trunking Protocol (VTP) is another Cisco proprietary layer 2 protocol based on the client/server model which is used to broadcast VLAN information such as VLAN name and ID across the network. It has been introduced to synchronize VLAN information with all switches inside the network by using the same VTP domain and password. This protocol helps network administrators to efficiently manage VLANs with less time by creating or deleting a VLAN in one switch and syncing this information across all the switches in the network instead of creating or deleting VLANs in every switch.

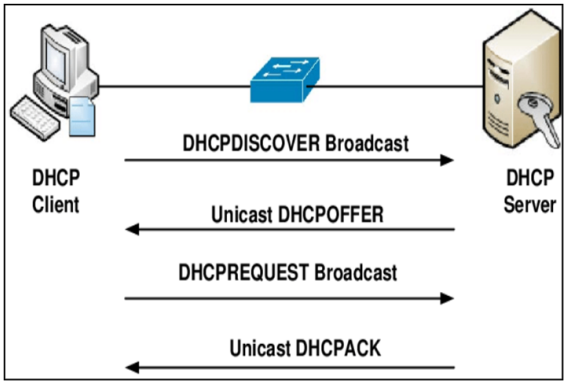


Figure 8: Dynamic Hosting Configuration Protocol is working between the client and the server [8, 9]

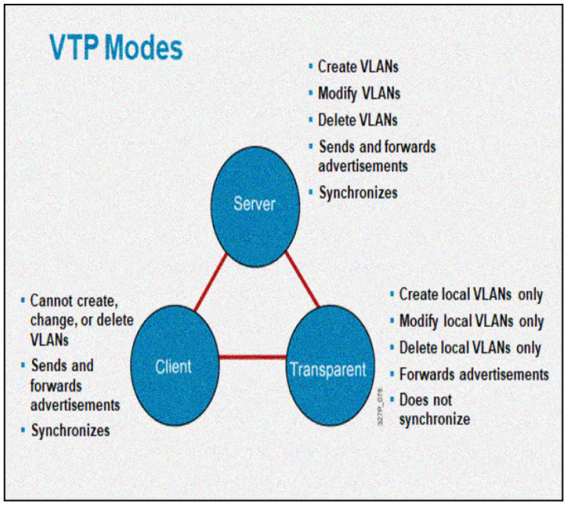


Figure 9: Virtual Trunking Protocol Modes [10]

As shown in figure 9, there are three VTP modes which can be configured in Cisco switches:

1- VTP server: Usually configured in the core switch that will advertise VLAN information across the network.

2- VTP client: Enable the switch to collect and save VLAN information.

3- VTP Transparent: The switch will be able to resend the VLAN information from the VTP server and, at the same time,

can create or delete VLANs locally in it. However, the switch will not sync on its own VLAN information to the network.

VTP attack methods launch on the network through modifying the switch's VLAN, either by adding a new VLAN or deleting an existing VLAN. This could cause a disruption in the whole network. Therefore, to protect the switch, the network administrator should use a VTP password and disable VTP protocol if not required.

4 Experimental Setup

The need for a hosting machine is essential for preparing the development environment to deploy the enhanced proposed solution (enhanced PacketFence). The core setup consists of three parts. Firstly, the hosting machine and its specifications. Secondly, software tools such as operating systems, Yersinia security attack tool, and the open-source NAC solution. Finally, a GNS3 network simulator. The reader should notice that the core switch (policy enforcement point) is integrated with a GNS3 network simulator. We explain this in the following sections.

4.1 Hosting configuration

In this part a set of tools and machine configurations are introduced that describe how the environment is prepared and settled is explain below.

1. Hosting Machine The hosting machine is a PC with Windows Server 2016 R2 64-bit OS. The processor is Intel Xeon E5-2687W v2 (dual processor) running @ 3.40 GHz. The installed RAM is 128 GB DDR3. It also has an Intel I210 Gigabit network connection ethernet adapter for network communication.

2. Operating System and Software tools PacketFence is the network access control solution used in the development environment along with Yersinia (Network Attacking tool). PacketFence can be downloaded from the official website (packetfence.org, 2021) as a standalone image or bundled with pre-deployed Linux OS images. In the case of using the PacketFence bundled image, a virtual machine is needed such as oracle virtual box is needed to import the image itself. In contrast, the standalone image needs to be installed manually in Linux OS. Both flavours need Linux OS. Yersinia [12] is an attacking open-source tool used to produce various types of attacks including CDP, root claim, DHCP starvation, floods, and so many others. Yersinia uses libpcap, ncurses, GUI, and libnet dependencies to function properly.

3. Network Simulator tool (GNS3) GNS3 is an open-source network simulator tool written in Python and can be downloaded from their official website (gns3.org, 2021). GNS3 can be integrated with many virtual machines including network components such as switches, routers, DHCP servers, firewalls, radius servers, and so many others. In addition, a variety of VM images, including operating systems, application servers, PacketFence, network firmware, and more, could be imported in GNS3. An Application Programming Interface (API), the so-called libpcap, is used to communicate between network components and endpoints. The product version 2.2.15 is used to deploy the environment.

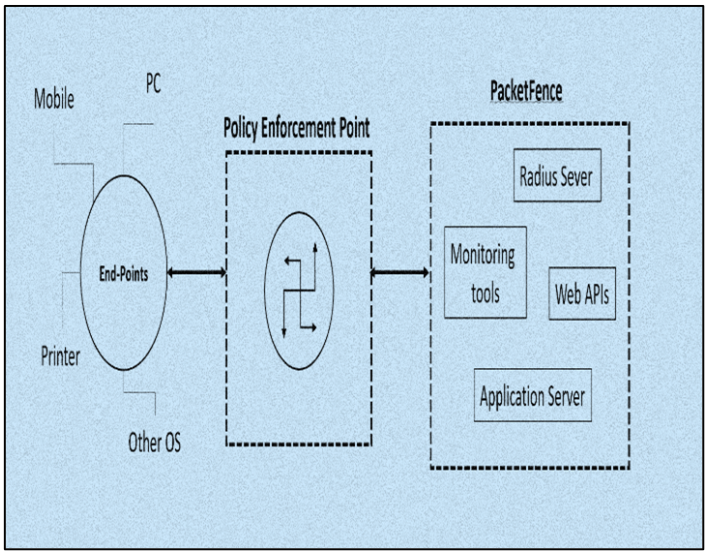


Figure 10: The relationship of policy enforcement points between PacketFence and endpoints

4.2 System Deployment

An overall deployment of network components, attacks, and configurations is presented in the following manner: network topology implementation, policy enforcement point configuration, launch of network attacks, and finally, hardening configuration on the policy enforcement point. In Fig. 10 a diagram is presented to illustrate the PacketFence system and how it communicates with the endpoint devices. PacketFence communicates with the policy enforcement point via a radius server as a built-in component to apply the pre-defined policies stored in the application server on the endpoint devices. It has its own application server and web APIs to configure and fetch information from devices in the network, such as the device's MAC address, IP, operating system, etc. This kind of information helps the PacketFence system set a profile for each endpoint device in the network and design a specific role for permitting/denying access to the network. Moreover, the system has an embedded monitoring tool which can perform on different levels: for example, at the system level, such as CPU utilization, Disk space level, system's RAM, and more. On the radius and authentication level, it shows the radius server latency, the number of requests on the radius, and the number of successful and unsuccessful authentications that occurred.

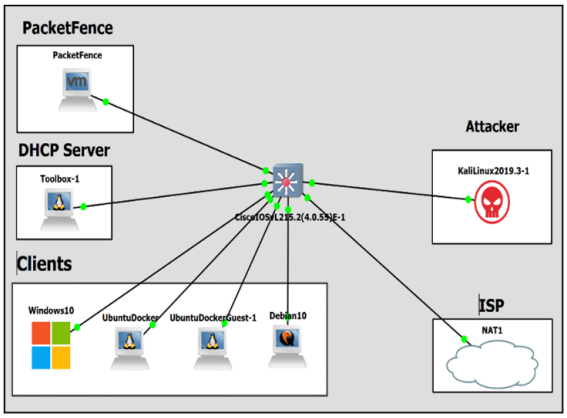


Figure 11: A network topology implemented in GNS3 simulator

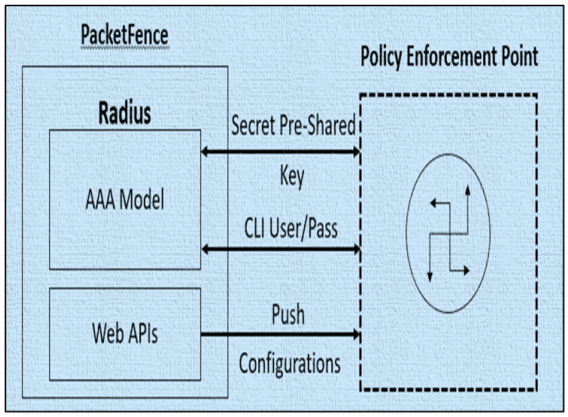


Figure 12: A communication between PacketFence and Policy Enforcement Point

The system can monitor the endpoint devices using SOH (Statement of health) protocols. It shows the number of devices on the network, security events that may happen when unregistered devices try to access the network, changes that take place on an authorized device (i.e., changing the device OS), or if the antivirus software is not up to date. In addition, the system can delegate unauthorized devices to a posture assessment process by redirecting the device to a defined URL which provides a list of guidelines to accommodate the network.

4.3 Network Topology Implementation

Network topology is an essential arrangement structure of network elements that aids network administrators to analyse the data flow, physical/logical interconnections, and transmission rates of the network. The network environment implementation forms a star topology. Mainly, the environment is composed of six elementary entities, as shown in Fig. 11.

(a) The open source NAC software solution (PacketFence). The DHCP server responsible for assigning IP addresses to network devices with a range of 192.168.122.0/24.

(b) The attacker is represented by Kali OS, which has great frameworks to launch a variety of different types of attacks. In this thesis, the Yersinia framework is used to launch various network attacks, as will be shown in part 3 of this section.

(c) GNS3 uses a NAT (Network Address Translation) to supply the internet connection to the network.

(d) Several nodes (clients) are used with different operating systems acting as endpoint devices.

(e) Finally, one of the major parts of the NAC solution (policy enforcement point) is represented by a layer 2 Cisco switch which applies the pre-defined policies inherited from PacketFence software on clients.

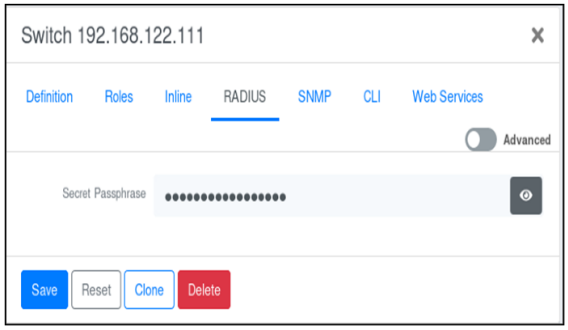
4.4 Policy Enforcement Point Configuration and Connectivity

In Fig. 12 we show the communication between PacketFence and the policy enforcement point.

The PacketFence initiates communication link between the embedded radius server and the policy enforcement point using AAA model (Figure 13).

```
aaa new-model
!
!
aaa group server radius packetfence
 server name pfnac
!
aaa authentication login default local
aaa authentication dot1x default group packetfence
aaa authorization network default group packetfence
```

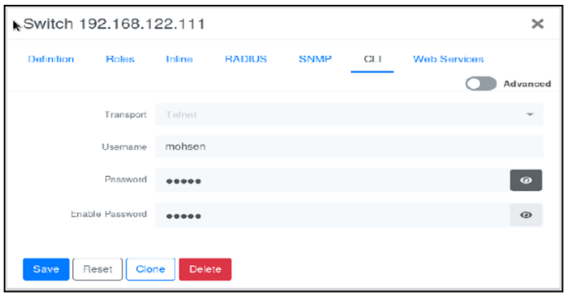
Figure 13: AAA model configuration in policy enforcement point



(a)PacketFence UI

```
aaa server radius dynamic-author
 client 192.168.122.126 server-key useStrongerSecret
 port 3799
!
```

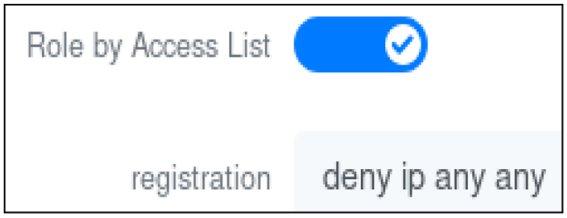
(b) The policy enforcement points of applying pre-shared key



(c) CLI Authentication configuration in PacketFence UI

Figure 14: Radius configuration in PacketFence UI

This can be done by setting a pre-shared key that should be configured in both PacketFence and the policy enforcement point as well as the CLI's (Command Line Interface) username and password. This way, the system can push the configuration to the policy enforcement point using the Web APIs, as shown in Figure 14. One of the methods used to control access to the network is the use of ACL (Access Control List). ACL is one of the essential structures in network configuration using permit/deny rules to manage network accessibility (Cisco CCNA/CCENT Exam 640-802, 640-822, 640-816 Preparation Kit, 2009). PacketFence uses ACL which can be defined in the system's UI and then pushed to the policy enforcement point (see figure 4.3) with the pre-defined rules as shown in Figure 15.



(a) PacketFence UI

```
ip access-list extended Auth-Default-ACL
 permit udp any range bootps 65347 any range bootpc 65348
 permit udp any any range bootps 65347
 permit udp any any eq domain
 deny   ip any any
!
```

(b)ACL configuration in policy enforcement point

```
interface GigabitEthernet2/2
 switchport mode access
 ip access-group Auth-Default-ACL in
```

(c) Applying ACL on the switch interface level

4.5 Launch of Network Attacks

In the second part of the previous section, five attacks on a simulated network created on GNS3 had been introduced and discussed. As shown in figure 4, the attacker is assumed to be an end-user with respect to the network. In our scenario, the attacker performs all layer-2 attacks within the network. Technically, these attacks are launched via the Yersinia tool, which is hosted on a virtual machine (Kali Linux distribution). In addition, all network components,including VM machines, are centralized and monitored on GNS3 via a set of APIs and services. All five attacks (CDP – DTP – VTP – STP – DHCP starvation) are directed toward the policy enforcement point (Cisco Switch) where the first four attacks directly disturb the switch. Meanwhile, the remaining one (DHCP starvation) affects the endpoint devices via the switch.

4.6 Hardening Configuration

The main objective in this paper is to present a methodology for hardening an open-source network access control system (PacketFence). Configurations that are used to harden the system will be categorized as follows: configurations recommended by the Cisco official guideline, and a proposed configuration that resolves some of the introduced attacks that Cisco did not include in their guideline [7].

A. Configurations Recommended by Cisco Guideline Here, a resolution will be highlighted for each attack by implementing configurations on the Cisco switch from the Cisco guidelines [7] .

1. CDP Attack

As mentioned in [7] for hardening Cisco devices, CDP protocol should be only enabled on switches that relate to a trusted network. Configurations that need to be applied are.

(a) no cdp enable command in the interface level or using.

(b) no cdp run command on the global level of the switch.

2. DHCP Attack

Cisco guidelines emphasize dropping undesirable DHCP traffic requested from unauthorized DHCP servers by executing ip dhcp snooping commands on the switch.

B. Proposed Configurations

1. STP Attack

Mitigation of STP attacks can be accomplished by disabling STP on unnecessary ports using disable stp command and port security command on the interface level. In case the network is using a port-fast feature in STP configuration, a bridge protocol data unit (BPDU) guard bpdu guard command is used. Moreover, root guard commands are applied to prevent the attacker from claiming the root.

2. DTP Attack A resolution to this attack is achieved by turning off DTP protocols on all switch ports by executing the switchport nonnegotiate switch command. This turns all ports from auto negotiations to off.

3. VTP Attack

In this attack, a precaution is needed for protecting the network VLANs from being manipulated (add/delete/modify).

It is required from the administrator to enable a VTP password command and to stop using VTP protocols if not needed. Furthermore, a good practice to follow is to use VTP modes wisely based on the criteria of the environment.

Table 3: CDP attack affects CPU Utilization over time

Minute	Number of packets	CPU Utilization
1	28,000	22%
3	85,000	31%
5	150,000	34%
7	210,000	72%
10	375,000	84%
Average		48.6%

5 Obtained Results and Discussion

In this part of the paper, we will show the results obtained in the research before and after introducing several network attacks. The dataset of the research is made up of five attacks tackling a major part of the PacketFence solution (the Policy Enforcement Point) which is a Cisco network switch under model numbers 2960, 2970, 3560 and 3750.

Yersinia is a leading tool to generate attacks and is heavily used in security network research (Opennac.org, 2021). The UI of Yersinia follows a WYSIWYG [[12], [8], [11]]. See Figure 17. All attacks introduced here are generated using the Yersinia tool operated on Kali Linux OS. Next, objective, and subjective results will be shown for each attack.

5.1 Cisco Discovery Protocol Attack Evaluation As we mentioned earlier, a CDP attack is launched using the Yersinia tool attacking the policy enforcement point. Figure 18 shows an empty table as an initial condition of neighbor devices that use CDP protocol. By contrast, figure 19 shows the generated devices from launching a CDP flood attack.

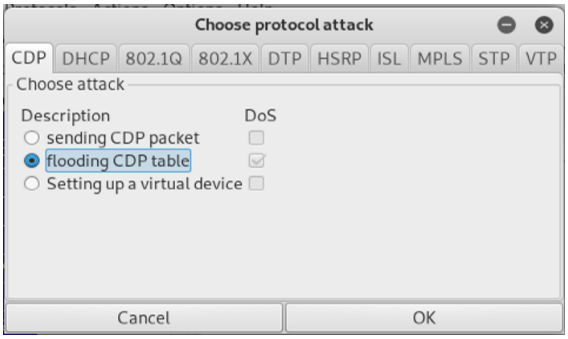


fig. 16.Yersinia UI for launching various type of attacks.

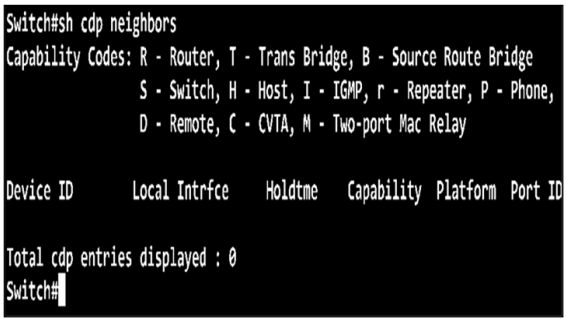


Fig. 17. A switch shows a CDP neighbor before launching the attack

The CDP flood attack dramatically increases the CPU utilization of the switch as the number of packets increases over time, as shown above in table 3, and is illustrated in the chart of figure 20 (e). The CDP attack is launched over time at different durations, starting from 1 minute up to 10 minutes. Figure 20 (a) shows the CPU utilization before starting the attack while (d) shows the CPU utilization after the end of the attack. As shown in (e), the attack significantly decreases the performance of the switch as CPU utilization increases for a period.

5.2 Dynamic Hosting Configuration Protocol Attack Evaluation

Another attack that can be generated using Yersinia is a DHCP starvation attack which can be avoided by applying a hardening configuration. The following figure 21 shows how the attack prevents the endpoint devices from obtaining IP addresses to access the network. After launching the attack, the dedicated process in the end-device system, which is responsible for obtaining IP addresses, is stalled due to the exhausted DHCP server, which gets numerous requests.

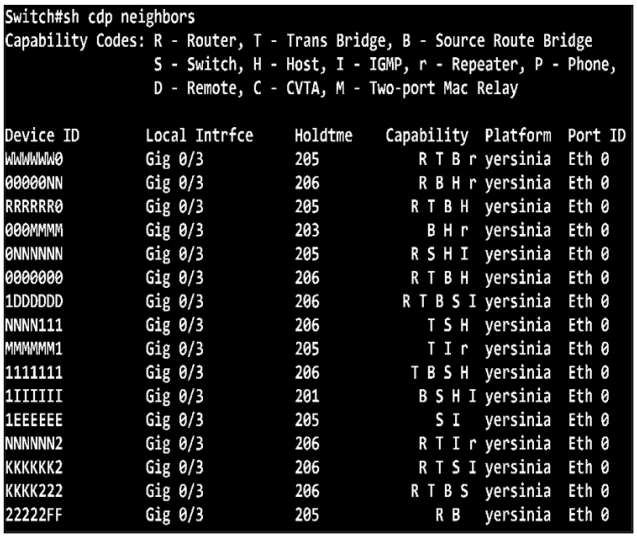


Fig. 18. A switch shows a CDP neighbor after launching the attack with dummy generated devices

```
Switch#sh processes cpu sorted | include CPU
CPU utilization for five seconds: 14%/0%; one minute: 27%;
Switch#
```

(a) CPU utilization before attack at t0(min)

```
Switch#sh processes cpu sorted | include CPU
CPU utilization for five seconds: 22%/0%; one minute: 21%;
Switch#
```

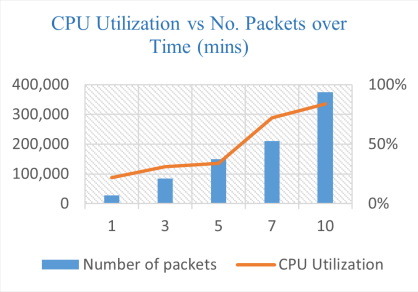
(b) CPU utilization after attack at t1(min)

```
Switch#sh processes cpu sorted | include CPU
CPU utilization for five seconds: 34%/0%; one minute: 40%;
Switch#
```

(c) CPU utilization after attack at t5(min)

```
Switch#sh processes cpu sorted | include CPU
CPU utilization for five seconds: 84%/0%; one minute: 58%;
Switch#
```

(d) CPU utilization after attack at t10(min)



(e) A chart illustrates a CDP attack affects the CPU utlilization over time

Fig. 19. Switch shows a CPU utilization before and after launching attack

```
root@UbuntuDocker:~# ifconfig
eth0      Link encap:Ethernet  HWaddr ae:3f:7d:0a:37:96
          inet addr:192.168.122.186  Bcast:192.168.122.255  Mask:255.255.25
5.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:4004 errors:0 dropped:15 overruns:0 frame:0
          TX packets:61 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:1158290 (1.1 MB)  TX bytes:19486 (19.4 KB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

root@UbuntuDocker:~#
```

(a) An endpoint device already obtained IP address before the attack

5.3 Spanning Tree Protocol Attack Evaluation

As discussed previously, this attack works on changing the cost of the switch and trying to alter the topology of the network. In figure 22, the initial cost will be zero. As attacks are involved, the cost will be changed as shown in (a) and (b). Moreover, the attack changes the traffic bandwidth interface port. As the number of attacks passes, the root cost of the device (switch) is accordingly changed, as shown in table 4. The assumption here is that the attack is launched with a secondary edge switch connected to the policy enforcement point switch (the device).

```
root@UbuntuDocker:~# ifconfig
eth0      Link encap:Ethernet  HWaddr ae:3f:7d:0a:37:96
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:610 errors:0 dropped:15 overruns:0 frame:0
          TX packets:14 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:177416 (177.4 KB)  TX bytes:4284 (4.2 KB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

root@UbuntuDocker:~#
```

(b) Same endpoint device is stalling and unable to obtain IP address due to DHCP attack Figure 20. A DHCP attack is applied and stalls the process of obtaining IP address in the system

```
Switch#sh spanning-tree root detail
VLAN0001
  Root ID     Priority    32769
              Address     0c0b.cde0.cb00
              This bridge is the root
              Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
VLAN0100
  Root ID     Priority    32868
              Address     0c0b.cde0.cb00
              This bridge is the root
              Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
Switch#
```

(a) Before launching STP Attack

```
Switch#sh spanning-tree root detail
VLAN0001
  Root ID     Priority    32768
              Address     5254.00ac.0f4c
              Cost        4
              Port        1 (GigabitEthernet0/0)
```

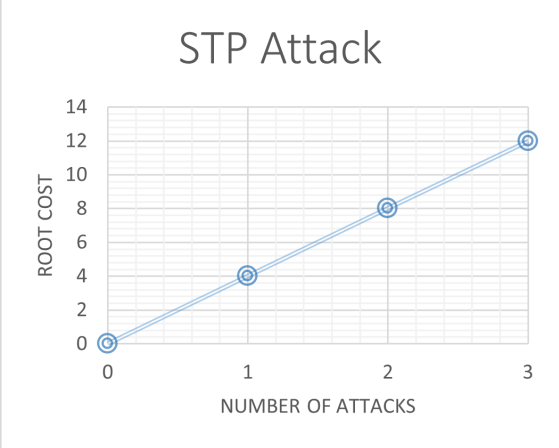
(b) 1st occurrence of STP attack by Yersinia

```
Switch#sh spanning-tree root detail
VLAN0001
  Root ID     Priority    32768
              Address     5254.00ab.0f4c
              Cost        8
              Port        4 (GigabitEthernet0/3)
```

(c) 2nd occurrence of STP attack.

```
Switch#sh spanning-tree root detail
VLAN0001
  Root ID     Priority    32768
              Address     2a5e.32c1.2a08
              Cost        12
              Port        1 (GigabitEthernet0/0)
              Hello Time   2 sec  Max Age 20 sec  Forward Delay  2 sec
VLAN0100
  Root ID     Priority    32868
              Address     0c0b.cde0.cb00
              This bridge is the root
              Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
Switch#
```

(d) 3rd occurrence of STP attack.



(e) A chart illustrated the occurrence of attacks and the root cost.

Fig. 21. The root cost of the device is affected due to an STP attack.

Table 4: Number of Attacks compared with changing root cost after launching an STP Attack

STP Attack Number	Root Cost
0	This bridge is the root
1	4
2	8
3	12
Average	8

5.4 Dynamic Trunking Protocol Attack Evaluation The idea of this attack is to force interfaces that use DTP protocols to be operated on trunk mode. This will leave the network vulnerable to various attacks. Figure 22 shows the switch before and after the attack.

```
Switch#sh interfaces trunk
Switch#
```

(a) Before launching DTP attack where no interface status is trunk.

```
Switch#sh interfaces trunk

Port        Mode         Encapsulation  Status        Native vlan
Gi0/3       auto         n-802.1q       trunking      1

Port        Vlans allowed on trunk
Gi0/3       1-4094

Port        Vlans allowed and active in management domain
Gi0/3       1-5,7,100

Port        Vlans in spanning tree forwarding state and not pruned
Gi0/3       none
Switch#
```

(b) The switch has been forced to enable Trunking mode under port Gi0/3 after launching attack.

Fig. 22. A DTP attack is applied, forcing the switch to enable trunk mode.

5.5 VLAN Trunking Protocol Attack Evaluation

A VTP attack alters the switch's VLAN, either by adding or deleting the VLAN entry. Some advanced methods can modify existing VLAN entries. This will lead to a disruption in the network. Figure 23 explains the status of the switch VLAN entries before and after launching the VTP attack. As shown in the figure, (a) No added/deleted entries or (b) Adding VLAN 10 under the name Attack to ensure the attack or (c) Deleting an entry in the VLAN table such as VLAN with id 7.

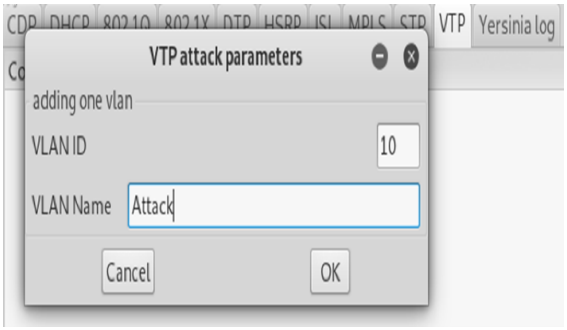
6 Conclusions

Although PacketFence is user-friendly and easy to integrate with many network components such as switches, proxies, and application servers such as radius and DHCP servers, the open-source solution has a wide array of issues that need to be altered and fixed. The second part of this research has been focused on a major part of open-source NAC solutions which is called the policy enforcement point, and which acts as a core switch for the overall solution. In this part, vulnerabilities were discovered after launching a set of attacks on the policy enforcement point component with the help of the Yersinia tool. The regular implementation of the policy enforcement points in open-source solutions (e.g. PacketFence) is missing a set of security guidelines. Although these guidelines are not dependent on starting up the overall solution, the marginalization of these guidelines during deployment with non-trusted networks might produce vulnerabilities.

Hence, hardening methods are represented as commands to enhance the device configuration. The third and final

part of this work concentrates on how to harden the policy enforcement point by protecting the network from exposed vulnerabilities, as mentioned in section 4. Guidelines and configurations are presented in this part of the work to tackle these vulnerabilities. As the results shows in section 4, systems that do not follow the recommended guidelines will suffer from exposed vulnerabilities that harness the network device (a.k.a PEP) in terms of CPU utilization, flooding MAC address table, VLAN entries, and root cost, all of which can lead to system failure

(a) VTP Attack via Yersinia UI



(b) Yersinia UI for adding VLAN

(c)Yersinia UI for deleting entry

Fig. 23. VTP attack is launched via Yersinia, which affects the switch VLAN entries

7 References

References

[1] Bawany, N.Z., Shamsi, J.A., Salah, K.: Ddos attack detection and mitigation using sdn: methods, practices, and solutions. Arabian Journal for Science and Engineering **42**, 425–441 (2017)

[2] Firoozjaei, M.D., Jeong, J.P., Ko, H., Kim, H.: Security challenges with network functions virtualization. Future Generation Computer Systems **67**, 315–324 (2017)

[3] Flores, J., Ramos, V., Lozada, R., Flores, T.: Analysis of solutions of network access control to improve in and out securities on corporative networks. In: 2017 CHILEAN Conference on Electrical, Electronics Engineering, Information and Communication Technologies (CHILECON). pp. 1–5. IEEE (2017)

[4] Hong, S., Oh, M., Lee, S.: Design and implementation of an efficient defense mechanism against arp spoofing attacks using aes and rsa. Mathematical and Computer Modelling **58**(1-2), 254–260 (2013)

[5] Inamdar, M.S., Tekeoglu, A.: Security analysis of open source network access control in virtual networks. In: 2018 32nd International Conference on Advanced Information Networking and Applications Workshops (WAINA). pp. 475–480. IEEE (2018)

[6] Iqbal, S., Sujatha, B.: Secure key management scheme for hierarchical network using combinatorial design. Journal of Information Systems and Telecommunication (JIST) **1**(37), 20 (2022)

[7] Iserovich, H.: Empowering network infrastructure Cybersecurity. Ph.D. thesis, The Interdisciplinary Center, Herzliya (2020)

[8] Kim, E., Kim, K., Lee, S., Jeong, J.P., Kim, H.: A framework for managing user-defined security policies to support network security functions. In: Proceedings of the 12th International Conference on Ubiquitous Information Management and Communication. pp. 1–8 (2018)

[9] Liu, D.: Cisco CCNA/CCENT Exam 640-802, 640-822, 640-816 Preparation Kit. Syngress (2009)

[10] Nunoo-Mensah, H., Akowuah, E.K., Boateng, K.O.: A review of opensource network access control (nac) tools for enterprise educational networks. International Journal of Computer Applications **106**(6) (2014)

[11] Rikhtechi, L., Rafe, V., Rezakhani, A.: Secured access control in security information and event management systems. Journal of Information Systems and Telecommunication **9**(33), 67–78 (2021)

[12] Roy, S., Sharmin, N., Acosta, J.C., Kiekintveld, C., Laszka, A.: Survey and taxonomy of adversarial reconnaissance techniques. ACM Computing Surveys **55**(6), 1–38 (2022)

[13] Syed, N.F., Baig, Z., Ibrahim, A., Valli, C.: Denial of service attack detection through machine learning for the iot. Journal of Information and Telecommunication **4**(4), 482–503 (2020)

Authors

Kalim Qureshi is an Associate Professor of Information Science Department, Kuwait University, Kuwait. His research interests include network parallel distributed computing, thread programming, concurrent algorithms designing, task scheduling, performance measurement and medical imaging. Dr. Qureshi receive his Ph.D and MS degrees from Muroran Institute of Technology, Hokkaido, Japan in (2000, 1997). He published more than 60 journal papers in reputed journals. His email address: kalimuddin.qureshi@ku.edu.kw and kalimuddinqureshi@gmail.com

Mohsen Al-Shamali is a network engineer at Central Bank of Kuwait. He completed his BS in Management Information and currently he completed his MSIT in 2022. His Email address is mohsen.alshamali@ku.edu.kw

Mostafa Abd-El-Barr received his PhD degree from the Department of Electrical and Computer Engineering, University of Toronto, Canada in 1986. He was with the Department of Information Science, College of Computing Sciences and Engineering (CCSE), Kuwait University 2003-2020. He was also an Adjunct Professor with the ECE Department, University of Victoria (UVic), BC, Canada 2009-2020. He is now the Chairman of the Electrical Engineering Department, Badr University in Egypt. His research interests include Information Security, Design and Analysis of Reliable Fault-Tolerant Computer Systems, Computer-Networks-Optimization, Parallel Processing-/Algorithms, Multiple-Valued Logic (MVL) Design Analysis, VLSI System Design, and Digital Systems Testing. He is the author and/or co-author of more than 185 scientific papers published in journals and conference proceedings/symposia. He has three books published (two are translated to the Chinese Language). Professor Abd-El-Barr is a Senior IEEE Member and a member of the International Association of Engineers (IAENG). He is a Senior International Associate Editor of the International Journal of Information Technology Web Engineering and a member of the Editorial of the International Journal of Computer Science and Security (IJCSS). He is also an official IEEE/EAC/ABET evaluator. Dr. Abd-El-Barr is a Certified Professional Engineer in Ontario, Canada.