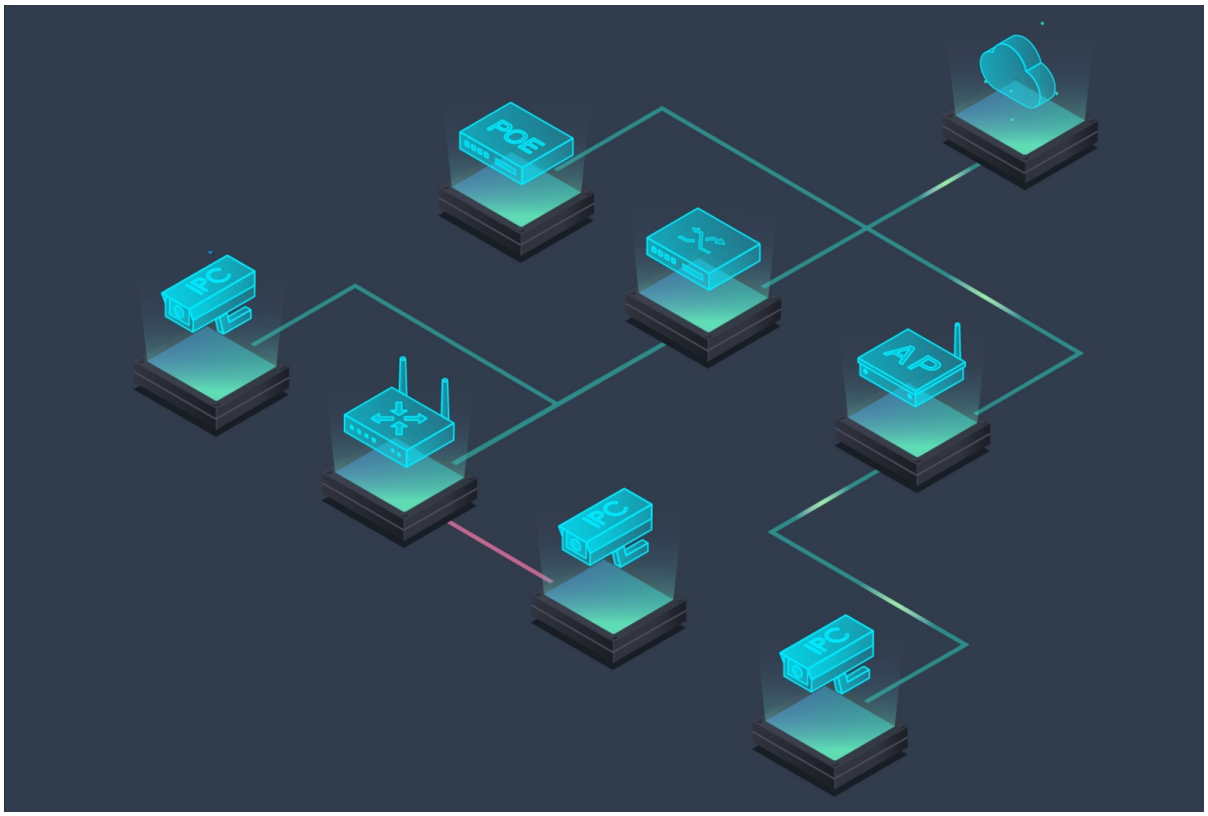
Design Report

The AAA Project



Name: Andres Vivas Rodriguez
Student number: 345155
Institution: Hanzehogeschool Groningen
Education: HBO-ICT, Network & Security Engineering

Internship Supervisor: Thies Keulen
Internship Lecturer: Jos Bos

Company: Axians ICS Groningen
Company supervisor: Peter Eek

Table of Contents

1. Introduction

This report will be known as the design report. This is the next phase of the AAA project, the *design-phase*. The design-phase is about giving an idea on how the project will be done/implemented later on.

The analysis report was about the gathering of information about the current environment and also the current problem which the internship project is based on. This design report will use that information in order to build a design idea on how the project will be implemented. The point of this report is to establish a future environment in which the project will be built, configured and tested. There are alot of requirements needed in order to make this future environment a reality. This report will go through all of the necessary components needed in order to achieve the best possible scenario for the future environment.

The next chapter will be about the chosen AAA solution and how it came to be. Chapter 3 will discuss the future environment, which is how the project will be implemented along with the necessary components. Chapter 4 will be discussing the potential security measurements that will be enforced and lastly, the conclusion.

2. The Chosen AAA Solution

In order to bring the project to a design phase, a decision must be made on which specific AAA solution should be implemented. The decision on which AAA solution can not come from the internship student but from Axians. This chapter will be about the chosen solution and how it came to be.

2.1 The Meeting

After talking with the company supervisor, it was decided that in order to come to a decision on which AAA should be implemented there needs to be a meeting set up with a specific group of employees of Axians. This group of people will be the representatives of Axians in this case. The meeting participants will consist of the internship student and 4 Axians employees, which are:

- Company Supervisor
- Sales Representative
- Project Supervisor
- Senior Network Engineer

The idea is to perform a presentation showing the different possible choices of AAA solutions and provide a proposal in the end on which of the different AAA solutions would be the most attractive choice. From here on the representatives can discuss which solution they think would be the best choice.

2.2 Potential AAA solutions

Thanks to the extensive research done and displayed in the analysis report (Rodriguez, 2022a), the gathering of information was already done. What follows next is to present the information and discuss the different AAA solutions with the representatives of Axians. Each solution has their own pros and cons on how they perform. It is the intention that the internship student goes through these pros and cons in order to give the representatives a good understanding of the differences between the solutions.

You can see a summary of the potential AAA solutions that will be discussed, along with each of their respective software options in the table below.

RADIUS	DIAMETER	TACACS+	CLEARPASS
freeRADIUS	freeDIAMETER	TACACS.net	Aruba ClearPass
Network Policy Server (NPS)		TACACS+-as-a-Service	
RADIUS-as-a-Service		CISCO ISE	

2.2.1 RADIUS solution

freeRADIUS

This a very good choice because of it being lightweight due to it running on a Linux-based operating system. Aside from that is the low cost (free) a very financially attractive choice and also relatively simple to configure.

The issue with this option is that freeRADIUS has to be installed on its own separate server and thus requires extra “hops” to be able to reach the Active Directory (user database) everytime it needs to perform AAA activities.

Another issue is that because the solution is an open source software, this means that everytime there's a new update out it may or may not tamper with the configuration. This gives more work to the designated employee in constantly making sure that the AAA solution is working fine after every update.

Network Policy Server (NPS)

The NPS is also a potentially good choice due to it being able to be installed in the same server as the Active Directory. This means that this solution requires 0 hops due to the fact that the user data is on the same server, which makes this the best performance-based solution. Aside from this, the cost is also low (free) as NPS is a built-in feature of the “Server Manager” on the Windows Server operating system.

The issue of this solution is the fact that it can be difficult to implement and integrate in an existing IT landscape.

RADIUS-as-a-Service

RADIUS-as-a-Service can also be a good choice. Due to not needing to have much knowledge on the subject as this solution would be installed and configured by an external company. And (depending on the need of Axians) there may be no need for any servers, this can save a lot on the total cost of the solution.

One of the issues with this solution is a majority lack of control over what goes into the RADIUS server since it is not self built and configured. Very stringent compliance requirements may also mandate an on-prem RADIUS server(s) for liability reasons. Another reason is that Axians would be very dependent on the service provider for anything relating to the RADIUS server (i.e. incidents, changes, updates, etc.).

2.2.2 DIAMETER solution

freeDIAMETER

The freeDIAMETER solution is a very nice choice due to the fact that it is very lightweight. This is because it runs on a Linux-based operating system which takes less resources compared to other operating system choices (e.g. Windows).

The issue with this solution comes from the knowledge needed in order to configure and maintain this solution. Besides that this solution also shares the same issues as the previously mentioned ***freeRADIUS*** in 2.2.1 RADIUS solution. This specific solution also suffers from having very limited information publicly available, this makes it an extremely unlikely choice.

2.2.3 TACACS+ solution

TACACS.net
TACACS+-as-a-Service
Cisco ISE

The TACACS+ solutions are strictly Cisco proprietary solutions. This means that they are only compatible with Cisco-related devices/systems. Because Axians is built mostly on HP switches and not Cisco, none of the TACACS+-related solutions mentioned above can be seen as a successful solution choice(s).

If this issue was not the case, then the TACACS+-related solutions would have been good potential solution choice(s).

2.2.4 ClearPass solution

ClearPass

As Aruba's ClearPass is a very nice choice due to the fact that it supports both TACACS+ and RADIUS protocols for authentication, authorization and accounting. Another good thing about ClearPass is that it has a nice overview on all of its features through its web interface.

The main issue with this solution is that it is an expensive solution, so expensive in fact that it can be considered an "unnecessarily" expensive solution for what Axians would need it to do.

2.3 The Proposal

One of the requirements from Axians for the current project was to come up with a proposal for the best-fitting solution and discuss this with the company (Axians). In order to provide a proper proposal, the idea is to establish a table with every potential solution which showcases their strengths (++) and weaknesses (--). Based on these strengths/weaknesses it can be found out whether the specific solution is or is not the best-fitting solution.

The potential solutions used in establishing the table were the ones discussed in 2.2 Potential AAA solutions. The table in question can be seen below.

Solution	Performance	Cost	Maintenance	Compatible
RADIUS				
freeRADIUS	++	++	--	Linux
Radius-as-a-Service	+-	+-	++	Windows Server
Network Policy Server	++	++	++	Windows Server
DIAMETER				
freeDIAMETER	++	++	--	Linux
TACACS+				
TACACS.net	+-	--	+-	Cisco
TACACS+-as-a-Service	+-	+-	++	Cisco
CISCO ISE	+-	--	++	Cisco
CLEARPASS				
ClearPass	+-	--	++	all

Legend: Good, Decent, Bad

As seen from the table above, every potential solution is checked on four important categories: their performance, the cost, maintenance and the system the solution is compatible with. While all 4 are important, the compatibility category might be the most important factor due to the fact that the specific solution would need to be compatible with the current network/systems of Axians. The next step is to narrow down the amount of possible solutions in order to come closer to a proper proposal.

As explained previously in 2.2.3 TACACS+ solution, the TACACS+-related solutions were all given “No” as potential choice(s) which means they will not be taken into consideration as a proposal at the end.

There are also two more solutions that can be removed, these solutions are *freeRADIUS* and *freeDIAMETER*. The reason for this decision for removal is because these are open source softwares and as mentioned previously in (and as seen on the table) these solutions require a lot of time/effort spent on maintenance. If the time spent on the maintenance is high then this can severely affect the cost.

The new table looks like the following:

Solution	Performance	Cost	Maintenance	Compatible with
RADIUS				
Radius-as-a-Service	+−	+−	++	Windows Server
Network Policy Server	++	++	++	Windows Server
CLEARPASS				
ClearPass	+−	−−	++	all

Please refer to the table above, there are now three possible solution choices that are potentially good choices for the AAA solution. From the table it is clear to see which solution has the most benefits compared to the other two, this solution is the ***Network Policy Server (NPS)*** solution. This will also be chosen as the proposed solution given to Axians.

The reason for choosing NPS is mainly because it provides more/better benefits to Axians than the other two solutions. Here is some of the reasons why NPS is the better option:

- NPS is better in performance due to it being installed in the same server as the Active Directory (stores user credentials) which makes the solution itself costless
- NPS is cheaper to build/implement due to already existing servers and knowledge
- NPS does not need to constantly be checked/monitored post-implementation, which saves time and cost on the maintenance
- Due to needing a second server as a secondary AAA (redundancy) server this would double the cost of the other solutions, while for NPS it still remains costless. This is because there are multiple servers(Domain Controllers) where the NPS can be installed.

2.4 Axians' Decision

After the meeting, the representatives of Axians discussed and agreed to the proposal given.

The project can now continue on to the future environment plan with the chosen AAA solution of the **RADIUS - Network Policy Server (NPS)** solution.

3. The Future Environment

This chapter will go over the future environment of the AAA project.

The future environment could also could be referred to as the “desired environment”, which is an environment where the AAA project will be implemented with little to no surprises as everything within that environment will be specifically chosen in order to implement and test the chosen AAA solution successfully.

Keep in mind that the future environment that will be planned here is strictly intended for the AAA project and not an improvement or to change in any way the current network of Axians, as this is not permitted nor allowed for an internship student to do so. The basic idea is that if the AAA solution works in this environment it is highly likely that it will also work in the live production environment.

3.1 Overview

The general idea is to create a network where the chosen AAA solution can be implemented and tested. This network must be within a virtual environment in order to avoid any harm that may or may not be brought to the live production environment.

While this network will be in a virtual environment, It must act as if this is the production environment. The idea behind this decision is if the AAA project can work within this virtual environment, then it would be safe to say that it will work in the live production environment. This way if the project runs successfully, it is possible to smoothly bring the AAA service from the virtual environment to the live production environment.

After discussing the idea with the project supervisor and the network employees, they were more than happy to supply a virtual environment where the project can be built, implemented and tested. They call this virtual environment “Zandbak”. This is Axians’ personal testing environment where they test certain systems or services before selling/bringing them to their clients.

An additional plan is to have the project be able to be (as close as possible) fully automatically deployed. This means that with the use of an automation software/tool the servers/switches will be automatically configured, for repetitive purposes. The software program that will be used and how this is going to be implemented will be explained later on.

The image below shows a network drawing on how the future environment will look like. The drawing consists of multiple components that will be necessary in order to build, implement and test the AAA project successfully.

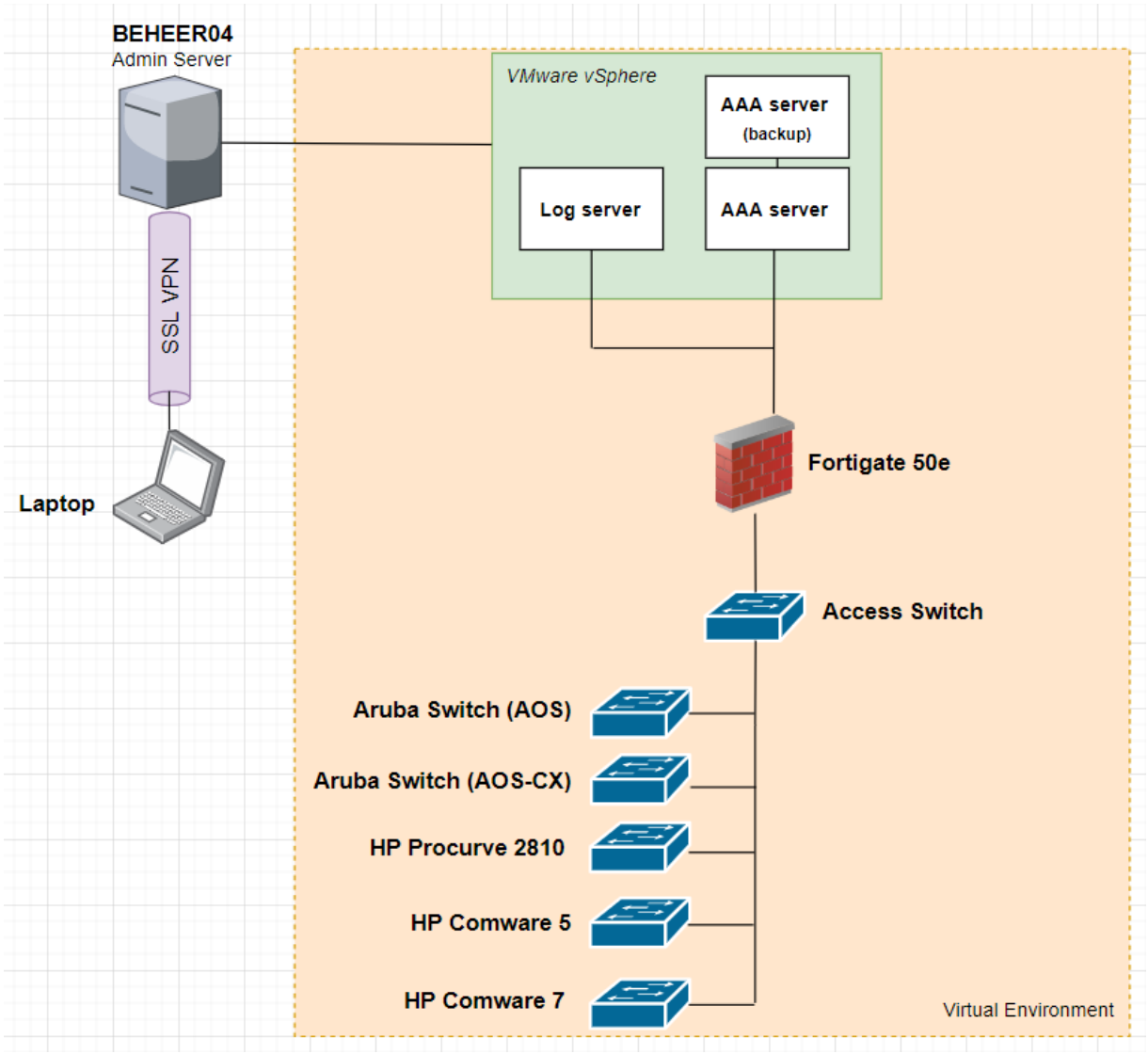


Figure 1: Future Environment Overview

The future environment will consist of 3 “sides” each of which consists of different component(s). These sides are: the server side, the network switch(es) side and the router side that interconnects them.

Server Side

Everything above the router will be known as the “server side”. There will be two servers that are going to be used, an AAA server and a Log server. The AAA server will be running the chosen AAA solution (RADIUS) and then sends the logs to the 2nd server, the Log server.

These servers will be set up as virtual machines (VMs), running on VMware Vsphere. Vsphere is a server virtualization software application from VMware which serves as a complete platform for implementing and managing VM infrastructure on a large scale.

The VMware Vsphere application runs on one of the admin servers of Axians (BEHEER04) and in order to access this server a laptop from Axians needs to be used with the pre-installed SSL VPN in order to connect securely to the admin server in question.

Switches Side

Everything below the router will be known as the “switches side”. One of the requirements in order for this project to be successful is that the chosen solution must work with the network switches that Axians currently use within their live production environment.

Because of Axians’ complex network and the current insufficient knowledge of the inner workings, a meeting was conducted with the senior network engineer. The plan is to summarize all of the switches that Axians uses into a specific set amount, enough to be able to use for testing the AAA service and verify that it works with the specified network switches in question.

The Router

The general idea is that any router can be used as long as they have the functionality of being able to use a RADIUS server for authentication in their settings. The reason for this is because when a user tries to log in (from the switches side) the router must be able to forward this “log-in-request” to the AAA server in order to authenticate the user.

These “sides” will be discussed in more detail in the next chapters in order to fully understand the plan on how the AAA project will be implemented.

3.2 The Servers

As mentioned before, the idea is to have 2 main servers: an AAA server and a Log server. These 2 servers are the foundation of the AAA project, without them the project would not be possible. Aside from the mains servers, there is also the possibility of having a 2nd AAA server for redundancy purposes. See image below.

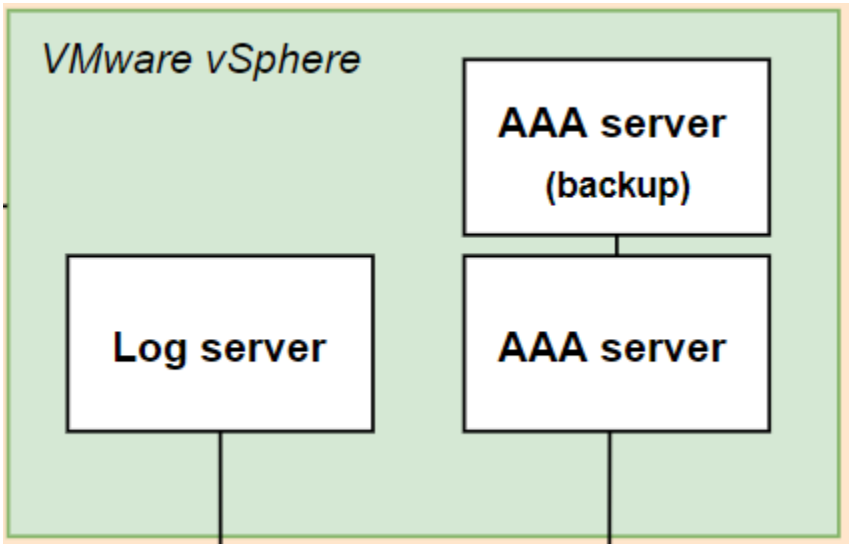


Figure 2: Server Side

The general idea behind this redundancy (backup) AAA server is if the primary AAA server goes down (i.e turns off or is unresponsive) or needs maintenance, the 2nd (backup) AAA server will take over the job of the primary AAA server. This is to minimize the downtime of the employee's work within a company.

3.2.1 The AAA Server

Since the chosen AAA solution turned out to be a Network Policy Server (NPS), this means that the AAA server will be running on the ***Windows Server*** operating system. The idea is that the NPS will be installed on a server that will function as the Domain Controller (DC), since this is where the NPS will be installed in the live production environment in the future.

In order to act like a Domain Controller, the server needs to have a number of services installed. These services are:

Active Directory Domain Services (ADDS)
A directory is a hierarchical structure that stores information about objects on the network. A directory service, such as Active Directory Domain Services (ADDS), provides the methods for storing directory data and making this data available to network users and administrators.

Domain Name Service (DNS)

Active Directory Domain Services (AD DS) uses Domain Name System (DNS) name resolution services to make it possible for clients to locate domain controllers and for the domain controllers that host the directory service to communicate with each other.

Network Time Protocol (NTP)
The Windows Time Protocol synchronizes the date and time for all computers managed by Active Directory Domain Services (AD DS). This is needed in order to configure the router and switches to sync with the time on the DC server, this makes all devices have the appropriate time so that the logs show the proper activities on the proper timeline.

After the DC is set up, the next step is to install the main component of the AAA project, the *Network Policy Server (NPS).* The NPS consists of 3 main components that must be configured in order for the AAA service to work properly. These components can be seen in the table below.

NPS	Description
RADIUS Clients	This is where you add all of the Network Access Servers (NAS). For example: switches, routers, APs (access points)
Policies	The policies are where the Authentication/Authorization part of the NPS happens. The policies either grants or denies access to the network depending if the accessing user complies with the specifically made policy
Accounting	This is where the Log Server will be specified in order for the NPS to be able to send the accounting data (logs)

3.2.2 The Log Server

After discussing the specifics with the senior network engineer, it was decided that the Log server will be running on a **CentOS 7** Operating System (OS). The reason for this decision is because the log server in the live production environment runs on CentOS 7.

The idea is that the log server will have the basic installation (up-to-date) as it is only needed to receive logs from the AAA server. In order to receive said logs there needs to be a specific software installed to act as a database where the logs will be stored for monitoring purposes. The database software that will be installed will be the ***Microsoft SQL Server.*** The reason for this decision is because the RADIUS NPS specifically works with SQL for its accounting (logging) service.

After the sql database is set up, the next step is to view the logs in a readable way. The reason behind this is to reduce the time spent trying to look for the specific information from a specific logging period. A good software application to have in this case would be **Microsoft SQL**

Server Management Studio. This is a software that will be installed on the DC server and can be linked to the Log Server in order to view the SQL Database and view more details.

This software is only going to be used to verify if the Log Server is logging the right information during the AAA project. Axians have no need for this in their systems. This is because Axians already have a monitoring software within their systems

3.3 The Switches

After a meeting with the senior network engineer, the decision was to summarize all of the switches they have in their current network into 5 switches for the AAA project, as shown in the image below.

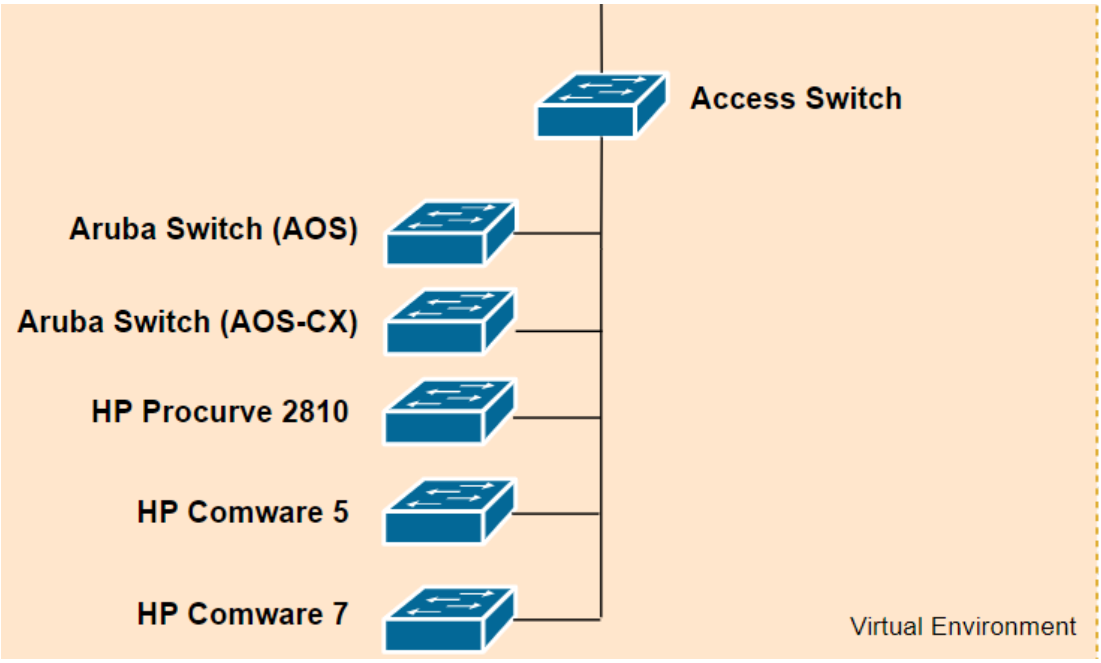


Figure 3: Switches side

Aside from the 5 switches mentioned above there will be an extra switch which will be between de 5 switches and the router. This switch will be known as the Access Switch. The only purpose of this Access Switch is to make the 5 physical connections coming from the 5 switches into 1 physical connection to the router.

3.3.1 Switch Configuration

The purpose of the switches is to provide a user who connects themselves to the switch access to the network, the system and anything they are authorized to access. However, in order for the user to gain this access they must first authenticate themselves with the AAA. This means the switches have to be configured in order to do so.

Before being configured for authenticating users with the AAA server, the switches will need a couple of necessary configurations. These configurations are the following:

Configuration	Description
Reset Switch	Ensures that no past configurations will be on the switch by resetting said switch
Console password	Set a password for the switch (console) to prevent being tampered by a 3rd party
Hostname	Give the switch a logical hostname
Banner	Banners provides a message when logging into the switch in order to warn the user about specific policies about the switch
Management IP	Configure a management IP address which can be used to logging into or troubleshooting the switch for future administrative tasks
Default Gateway	Configure the default gateway to the IP of the router that the switch will be connected to
Secure Shell (SSH)	Enable SSH in order to access the switches remotely. This removes the need to constantly needing to physically go to the switch for troubleshooting
Network Time Protocol (NTP)	Enable NTP in order to have the proper time on all the switches. This is especially important for having the right time on the log data of the switches
Interface configuration	Configure the interface(s) that connect the 5 switches to the access switch and from the access switch to the router

Aside from the basic configuration, the next step would be to configure VLANs (Virtual LANs). The reason for this is that VLANs provide flexibility in the network configurations and reduce administrative efforts. For this project the idea is to provide each switch (excl. Access Switch) with its own VLAN subnet, this will make it easier to manage as it will be easier to manage the switches based on their specific VLAN subnet.

The basic idea for the VLANs for each switch is shown in the following table.

Switch	VLAN	IP/Subnet
Aruba Switch (AOS)	10	x.x.10.0 /24
Aruba CX Switch (AOS-CX)	20	x.x.20.0 /24

ComWare 5 (COM5)	30	x.x.30.0 /24
ProCurve	40	x.x.40.0 /24
ComWare 7 (COM7)	50	x.x.50.0 /24

The basic idea is of course to use local (private) IP addresses, however, at the moment it is unclear how many available IP addresses exist within the test environment. This is what is meant by the x's in the table above. The order of the VLANs was only based on the switches being physically on top of each other. Lastly, the subnet (/24) decision was only based on the "standard" subnet used in most networks and can be later changed if necessary.

3.4 The Router

After discussing the necessity of a router with the project supervisor, the project supervisor offered a *FortiGate 50e* as a potential router choice for the project. The reason why this device was offered is because the network employees work a lot with FortiGate devices and thus have knowledge and experience which they are able to share. Another reason is that there is already one installed in the virtual environment which can be used for the project, this saves a lot of time in setting up a new router.

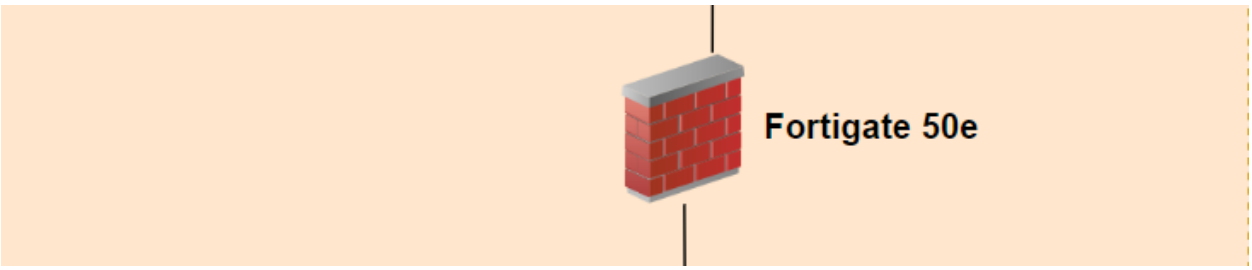


Figure 4: Router side

FortiGates are firewalls that provide high performance, multilayered security and deep visibility for end-to-end protection across enterprise networks. The FortiGate is therefore more than enough to act as the firewall/router for the AAA project. With this in mind, the FortiGate offered by the project supervisor was chosen as the acting router for the AAA project.

3.4.1 Router Configuration

The FortiGate will be the gateway between the switches and the servers, this means that the router has to be configured in order for the switches to be able to communicate with the AAA and the Log servers.

As the FortiGate is already installed and connected to the virtual environment, there is no need to perform any basic installations. The focus will now be in configuring the FortiGate with the necessary configurations needed for this project. These configurations can be seen in the table below.

Configuration	Description
Interface configuration	Configure the interface that is connected to the switches side
Static Route	Configure a static route to the server side and the switches side
VLAN configuration	Create VLANs (10,20,30,40,50) and assign them an IP address
Policy	Configure policies that allow the switches to reach the servers and the servers to reach the switches
AAA (optional)	If the FortiGate supports WiFi, configure FortGate as a RADIUS Client in order to authenticate WiFi users
DHCP configuration (optional)	Configure DHCP for the VLANs, in order for a connecting user to automatically receive the appropriate IP address

Aside from the configurations shown in the table above, another feature that FortiGates support is WiFi. This also means that (depending on the need) the WiFi can be configured to authenticate users that try to connect.

If the WiFi ends up as a needed portion for the project, then this needs to be established and configured. The necessary configurations for the creation of a WiFi network are seen in the table below.

WiFi Settings	Description
SSID	Name of the WiFi network
Security mode	This is the choice of wireless security protocols. The choice for this will depend on the needs of the company (Axians) and what the WiFi will be used for (e.g. WEP/WPA/WPA2)
Pre-shared key	Specify a password for the WiFi network
Client limit (optional)	Specify the total number of clients that are allowed to be connected on the specific WiFi network
DHCP configuration	Configure DHCP in order for the connecting users to automatically receive an appropriate IP address once connected

There are of course extra settings choices in configuring the WiFi network that were not listed in the table above. However, these choices are not relevant for the project or for making the WiFi network work.

3.5 Automation

As mentioned in 3.1 Overview, the idea is for the project to have an automated deployment feature which configures the AAA server(s) and the switches automatically with the necessary configurations with the use of a specific automation software.

The automation software that will be used is called Ansible. Ansible is an open source IT automation engine that automates provisioning, configuration management, application deployment, orchestration, and many other IT processes.

The reasons for choosing Ansible as the automation software instead of another option is because of the pre-existing knowledge of the software in question within Axians, specifically in the network department. Another reason is that since Ansible is an open source software, it is free to download and install.

A basic idea on how the Ansible automation process within the project will look like can be seen in the image below.

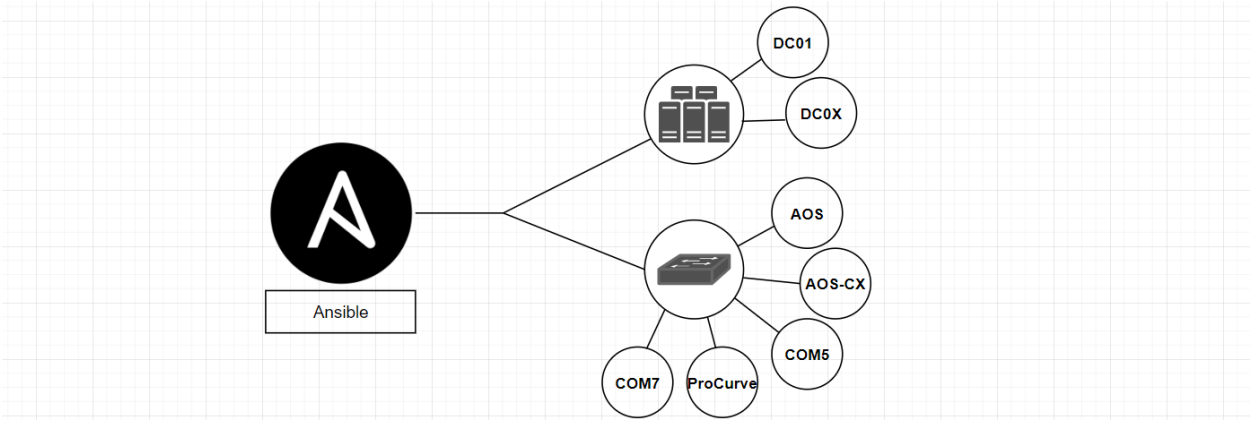


Figure 5: Ansible Overview

The image above shows the Ansible node (an Ansible server) which will be running on a Linux-based operating system. After talking to a network engineer on the idea of implementing Ansible, the recommendation is to install ansible on an AlmaLinux operating system as this is also used within the production environment. The main reason that Axians use Ansible is for Fortigate-specific devices.

Another thing to note is that the Ansible node is splitting up into two category types, one for servers and one for switches. This is needed to have the particulair configuration for the particulair device, in this case the servers will be the AAA server(s) running the Windows Server operating system and the network switches each running their own switch operating system.

The core components that Ansible needs to work are **playbooks** and the **inventory.**

Playbooks

Playbooks are essentially scripts written in YAML syntax which offer a repeatable, reusable simple configuration management and multi-machine deployment system. If a task needs to be executed more than once, a playbook is written and put under the source control which can then be used to push out new configurations or confirm the configuration of the particulair device it will be pushed onto.

Inventory

Ansible automates tasks on managed nodes or “hosts” in the infrastructure, using a list or group of lists known as inventory. The inventory defines which nodes (hosts) will be automated, with groups that can run automation tasks on multiple hosts at the same time. Once the inventory is defined, specific patterns can be used to select the host(s) or groups that are required for Ansible to run against.

4. Security

This chapter will be discussing the security measurements that are planned to be implemented with the AAA project.

4.1 AAA Server

As mentioned before, the NPS consists of 3 main components, 2 of which have security measurements that can be taken into consideration when implementing the NPS service. These 2 components are the RADIUS Clients and the Policies.

4.1.1 RADIUS Clients

The RADIUS NPS uses a *symmetric encryption* algorithm which encrypts/decrypts data sent between the server and client, as shown in the following diagram.

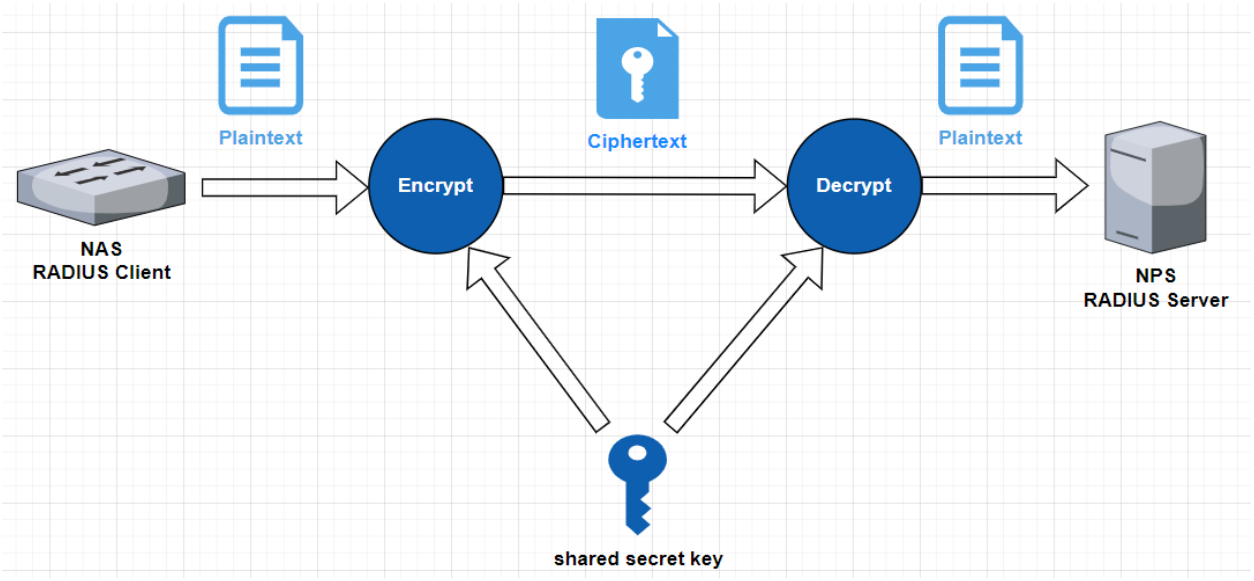


Figure 6: Symmetric Encryption Process

In this example the NAS encrypts its plaintext data (using a unique shared secret key) into ciphertext data, then proceeds to send this ciphertext over the network to the NPS. The NPS receives this ciphertext and decrypts it using the same shared secret key in order to get the original plaintext data.

The shared secret key needs to be manually created on the NPS. This means that the shared secret made on the NPS needs to be the same on the specific RADIUS-Client. As there is no specific password policy provided on how long/strong the password needs to be, the decision was left up to the internship student. With this in mind was the table below established with an example on how the shared secrets will be made.

RADIUS Clients	Shared secret
FortiGate 50e	*exampleSecret50e!*
AOS	*exampleSecretA0S!*
AOS-CX	*exampleSecretA0SCX!*
COM5	*exampleSecretCOM5!*
ProCurve	*exampleSecretPro!*
COM7	*exampleSecretCOM7!*

The idea behind the shared secret (example) passwords in the table above is that each device will have a strong password so that the devices cannot be easily tampered (breached) with. The following requirements is what is understood as a “strong” password:

- At least 12 characters long but 14 or more is better
- A combination of uppercase letters, lowercase letters, numbers and symbols
- Not a word that can be found in a dictionary or the name of a person or organization
- Significantly different from the previous passwords

The above list was made by Microsoft and will be used as an example/reference for future passwords made for the current project.

4.1.2 Policies

The NPS Policy is divided into 2 main components: the *Connection Request Policies* and the *Network Policies*.

- **Connection Request Policy** determines whether connection requests are processed locally for forwarded to remote RADIUS servers
- **Network Policy** determines who is authorized to connect to the network and the circumstances under which they can or cannot connect.

Each policy has ***Conditions*** and ***Constraints*** that have to be met in order to provide access to a connection request or a connecting user. When the conditions and constraints match, the connection request gets access granted via the policy.

Conditions

Conditions are a set of “prerequisites” that the requesting RADIUS Client needs to match when requesting access in order to gain access to the network. The more conditions implemented, the more secure the network will be. This makes it that only the desired RADIUS Clients will receive access to the network upon implementation.

The following table shows the conditions that will be in place when implementing the RADIUS server with an explanation why these conditions were chosen.

Condition	Description
Client Friendly Name	The Client Friendly Name condition specifies the name of the RADIUS client that forwarded the connection request to NPS.
Client IPv4 Address	The Client IP Address condition specifies the IP address of the RADIUS client that forwarded the connection request to the NPS
NAS Port Type	Ethernet

As you can see from the table above, only 3 conditions are going to be used. For the purpose of this test environment the main focus is to make sure that the RADIUS NPS works by accepting and granting access to the desired requesting device(s).

The amount of conditions can always be changed, which means if there is any need to add or remove certain conditions further in the project, then this can be easily done so.

For a full detailed list of every possible condition that is available in the NPS policies, please refer to *Appendix I: NPS Policy Conditions*.

Constraints

The *Constraints* component is where the “authentication methods” must be chosen/specified. The authentication method consists of authentication protocol choices that must match the one that the RADIUS Client will be using when communicating with the RADIUS NPS (server). Another thing to note is that more than one authentication method can be chosen.

The different authentication protocol choices can be seen below along with a short information on what they do and further down the reason if it will be considered to be chosen as one of the authentication method choice(s).

- **PAP (Password Authentication Protocol)**

When logging into a network resource (server), the user or device (NAS) is required to supply a username and password.

- The username and password are sent in cleartext format, so this method is considered insecure and should only be used as a last resort.

- **CHAP (Challenge Handshake Authentication Protocol)**

When logging into a network resource (server), the user or device (NAS) is challenged to supply a username and secret password and it authenticates through a 3-way handshake process.

- The resource (server) issues a challenge: what is the hashed value of the username and secret password?
- The user’s device (NAS) sends the hashed values to the resource device (server)

- The resource (server) evaluates the hashed values and either accepts or rejects the connection

- **MS-CHAP (Microsoft CHAP) or MS-CHAP-v2**

Functionally the same as CHAP, but is proprietary to Microsoft systems. Since its creation, Microsoft has updated MS-CHAP to MS-CHAP version 2. MS-CHAPv2 is more secure than MS-CHAP because it provides *mutual authentication* between the client and server. Therefore, when communicating with a server, one can be assured they are communicating with the correct server and not a rogue server impersonating the real one.

- **EAP (Extensible Authentication Protocol)**

The EAP protocol is a series of method interfaces that make up the framework known as EAP. EAP provides many options for authentication and is predominantly used on wireless networks. EAP also provides user authentication against a directory service, which allows administrators to track employee behavior or what they do with wireless access. There are currently many, many (around 100) different implementations of the EAP framework, but the RADIUS NPS only supports 2 specific ones: *EAP-MSCHAP-v2* and *Protected EAP*.

- EAP-MSCHAP-v2
    - EAP-MSCHAP-v2 is an EAP version of the common MSCHAPV2 authentication mechanism. The main difference in this method is that the server must have a digital certificate in order for this method to work.
- PEAP (Protected EAP)
    - PEAP is a protocol that encapsulates the EAP within an encrypted and authenticated Transport Layer Security (TLS) tunnel. It provides authentication by the use of *digital certificates* issued from a Certificate Authority (CA), both the servers and clients each have their own unique certificates which they use for authentication. This protocol uses primarily certificates as credentials and thus removes the need for password credentials.

With authentication method(s) layed out, the next step will be to decide the authentication method(s) that will be used to authenticate the RADIUS Clients. As a reminder the authentication method must match the one that will be used by the switch/router (RADIUS Client).

Switches

For the switches, the most secure authentication method would be the PEAP if they support this authentication method.

MS-CHAP/ MS-CHAP-v2/ EAP-MSCHAP-v2 are all Microsoft proprietary and thus can not be considered options for the switches, as the switches do not support them.

If none of the authentication methods above works, the last few choices would be either CHAP and if that isn't supported then PAP would be the choice as a last resort.

Router

For the router, the most secure authentication method would also be PEAP if they support this authentication method.

Since FortiGate supports the authentication methods of: MS-CHAP/ MS-CHAP-v2/ EAP-MSCHAP-v2, then these can be taken as a 2nd choice if the 1st choice isn't supported.

If none of the authentication methods above works, the last few choices would be either CHAP and if that isn't supported then PAP would be the choice as a last resort.

4.2 Log Server

The Log server that will be implemented in the test environment will not be used in the live production environment in the future. The reason for this is because Axians already has a Log server for their own system.

As the Log server only needs to receive logs from the AAA server, there are not many security measures that would need to be implemented, besides keeping it updated to the latest version. However, it is still necessary to mention what will be implemented.

4.2.1 Firewalld

Firewalld is a firewall management tool for Linux operating systems. It provides firewall features by acting as a front-end for the Linux-based operating systems. The reason why firewalld was chosen is purely because of it being the default firewall management tool. Having this firewall already built-in within the operating system saves time in finding a new one to install and also saves time in configuring said new firewall.

The reason why not another firewall was not chosen is due to the fact that the Log server will not be doing much, so the default firewall (firewalld) will be more than enough for the project. Another reason is that there is a small chance that the Log server will be implemented in the live production environment in the future, this is because Axians already has their own Log server where they store their respective logs.

The main purpose for using a firewall is to only allow the desired programs that will be used within the Log server to go through the firewall.

The next step now is to define the ports that are going to be used for the project for their associated program. The following table shows a list of the programs along, the ports they need and a reason why these programs are needed.

Program	Port	Reason
Microsoft SQL server	1433 / 1434	This creates the Log Database where all logs will be stored and is a necessary component for the project.
Secure Shell (SSH)	22	This program is useful in order to access the Log server remotely.
Network Time Protocol (NTP)	123	The NTP is extremely important in order to make sure that the logs are being received at the appropriate time.
RADIUS	1813	This is the Accounting port in which the AAA server will be sending the logs through.
Syslog (optional)	514	The Log server may also opt to receive logs from the network switches themselves (if configured).

4.3 Switches

The network switches are to be configured with some security measurements in order to minimize the chance of someone tampering with a specific switch.

The following table shows the security measurements that have to be taken into account when configuring the switches.

Configuration	Description
Console password	Every company has their own policy on how long/strong, for the purpose of the AAA project this will be small. The reason for this is because the switch resides in a testing environment and the fact that the switch needs to be consistently logged into for the duration of the project. Example of password: SWpass!!
Syslog (optional)	Enable syslog in order to send logs to the Log server. This might be optional since Axians already have this on their devices in the live environment
Disable unused ports	A good security measurement to take into account is to disable any unused ports, this prevents anyone from tampering with the switches. At least two ports on the 5 switches needs to be enables (one for the link to the Access Switch and one for a link to the user)
SSH	When generating RSA encryption keys for the SSH configuration, the minimum recommendation is 2048-bit. This can be changed if Axians have their own policy on how

	strong the keys need to be when implemented in their live production environment. * The recommendation is from the National Institute of Standards and Technology (NIST).

4.4 Router

As mentioned previously, the FortiGate that will be used for the project is already installed in the virtual environment. This of course means that certain security measures were taken when it was installed and configured.

For the project there are 2 main security measures that need to be taken into account on the FortiGate. The **WiFi security** for the potential wireless network and the **Firewall policies**. The policies of FortiGate are one of the most important aspects of the device (as is with any Firewall).

4.4.1 WiFi Security

In order to authenticate users that want to connect to the potential WiFi, the FortiGate needs to be configured with an AAA server. This configuration requires an authentication method to encrypt the communication between the FortiGate and the AAA server. The chosen authentication method will be based on the choice(s) made for the router in 4.1.2 Policies.

The following table shows the configurations needed in order to configure the AAA server(s) on the FortiGate.

Configuration	Description
Name	Name of the profile for the AAA configuration.
Authentication method	Specify the authentication method for the communication between the FortiGate and the AAA server.
NAS IP (optional)	Specify NAS IP of the source device. In this case this will be the IP address of the FortiGate. While this is an optional choice, it is beneficial to adding an NAS IP in order to receive more precise logging information
IP/Name	Specify the IP address of the AAA server
Secret	Specify the “shared secret” password. This password must be the same one as specified in 4.1.1 RADIUS Clients for the FortiGate
Secondary server	Specify the IP/secret of the secondary (backup) AAA server

4.4.2 Firewall policies

The FortiGate specific "**IPV4 policy**" includes the ability to accept or deny traffic, apply security profiles, shape traffic, log traffic and schedule a timeframe for a policy to apply.
The project will need a certain amount of policies in place in order for the FortiGate to allow specific traffic to go through, or else there can not be any communication between the FortiGate and the servers/switches. In order to get a better grasp on how the traffic will flow on the FortiGate was the following diagram made.

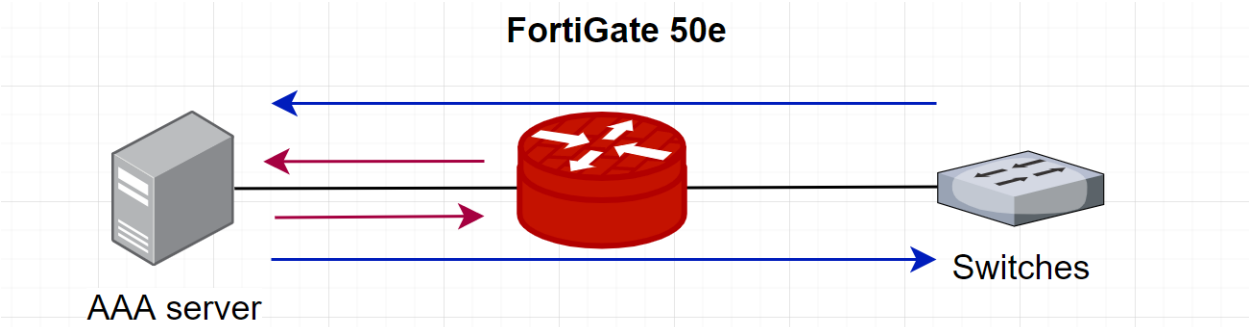


Figure 7: Traffic between FortiGate and servers/switches

As you can see from the image above, the arrows indicate the direction of the traffic flow. There needs to be a policy in place for every arrow in this case. This means that there needs to be 4 policies configured in total. Below is a list of the general idea on how the 4 policies will revolve around.

1. From FortiGate ➤ To AAA server
2. From AAA Server ➤ To FortiGate
3. From Switches ➤ via FortiGate ➤ To AAA server
4. From AAA server ➤ via FortiGate ➤ To Switches

In order to have these policies in place they must first be configured on the FortiGate. Below is a list of the configurations necessary in order to establish a policy.

Configuration	Description
ID	Policy identifying number. This is mostly used to refer to the specific policy within a log
Name	Name of the policy
Source	Source where the traffic will flow from
Destination	Destination on where the traffic will flow towards
Schedule (optional)	Defines a specific timeframe in which the policy will apply. This feature might not be used for the project, as it is not needed to have this time restriction during the project.

Service	This feature gives the option of choosing which specific traffic types (e.g RADIUS, ICMP, HTTP, DNS, etc.) is/are allowed through.
Action	Accept or Deny the specified action

For security purposes, there will also be an extra policy (e.g. called Implicit Deny) that denies any other services. This policy will deny everything that does not comply with the 4 main policies.

Conclusion

The design plan consisted of building the “best case scenario” for the project. This design report provides a concrete idea how the project will be implemented later on. The design report keeps in mind that the product will have to be implemented in the future production environment. This means that the test environment where this is to be implemented must act as if it is the production environment.

This document consists of listing a set amount of potential solution choices and through process of elimination, arriving at the best-fitting solution choice. This solution choice must be then presented to representatives of Axians as a proposal. The representatives will then discuss and agree or disagree to the proposed solution choice.

Afterwards, with the chosen solution now established, the future environment can be designed and established. The initial idea is to cover everything that needs to happen in order for the project to function without any unexpected events.

Lastly, a list of security measures has been established for every device that will be used when implementing the chosen solution. The security measures are there to provide a layer or layers of security for the network devices as they are to be implemented in a live production environment in the future. This reduces the risk of any potential security risks.

Sources and Literature

FORTINET | “FortiGate Handbook” | Retrieved 6 Oct 2022 from
https://docs.fortinet.com/document/fortigate/6.0.0/handbook/

Rodriguez, 2022a | Analysis Report - AAA Project | Retrieved 19-dec-2022, from
https://github.com/justRodriguez/NPS_Project/blob/main/Docs/2.%20Analysis%20Report%20(1).pdf

Appendix

Appendix I: NPS Policy Conditions

User Name	
User Name	The user name that is used by the access client in the RADIUS message. This attribute is a character string that typically contains a realm name and a user account name.
Connection Properties	
Access Client IPv4 Address	The Access Client IPv4 Address condition specifies the IPv4 address of the Access Client that is requesting access from the RADIUS client.
Access Client IPv6 Address	The Access Client IPv6 Address condition specifies the IPv6 address of the Access Client that is requesting access from the RADIUS client.
Framed Protocol	The Framed Protocol condition restricts the policy to only clients specifying a certain framing protocol for incoming packets, such as PPP or SLIP.
Service Type	The Service Type condition restricts policy to only clients specifying a certain type of service, such as Telnet or Point to Point Protocol connections.
Tunnel Type	The Tunnel Type condition restricts the policy to only clients that create a specific type of tunnel, such as PPTP or L2TP.
Day and Time Restrictions	
Day and Time Restriction	Day and Time restrictions specify the days and times when connection attempts are and are not allowed. These restrictions are based on the time zone where the NPS server is located.
RADIUS Client Properties	
Calling Station ID	The Calling Station ID condition specifies the network access server telephone number dialed by the access client.
Client Friendly Name	The Client Friendly Name condition specifies the name of the RADIUS client that forwarded the connection request to NPS.
Client IPv4 Address	The Client IP Address condition specifies the IP address of the RADIUS client that forwarded the connection request to the NPS
Client IPv6 Address	The Client IPv6 Address condition specifies the IP address of the RADIUS client that forwarded the connection request to the NPS
Client Vendor	The Client Vendor condition specifies the name of the vendor of the RADIUS client that sends connection requests to NPS.

Gateway	
Called Station ID	The Called Station ID condition specifies a character string that is the telephone number of the network access server (NAS). You can use pattern matching syntax to specify area codes.
NAS Identifier	The NAS Identifier condition specifies a character string that is the name of the network access server (NAS). You can use pattern matching syntax to specify NAS names.
NAS IPv4 Address	The NAS IP Address condition specifies a character string that is the IP address of the network access server (NAS). You can use pattern matching syntax to specify IP networks.
NAS IPv6 Address	The NAS IPv6 Address condition specifies a character string that is the IPv6 address of the network access server (NAS). You can use pattern matching syntax to specify IPv6 networks.
NAS Port Type	The NAS Port Type condition specifies the type of media used by the access client, such as analog phone lines, ISDN, tunnels or virtual private networks, IEEE 802.11 wireless and Ethernet switches.