Secure Network

by Student Researcher

Course	Network security
Institution	Academic Institution
Date	February 2025

Table of Contents

1 Introduction

1.1 Project Objectives

The primary objective of this project is to set up a network with end devices as securely as possible for a small startup company called LNXE-Corp. The company has currently ten employees in the development department and must support both stationary workstations and laptops in a client-server relationship while making robust security measures to protect the company's assets.

The report will demonstrate configuration setups for a local Area Network (LAN) in Cisco's simulation program Packet Tracer (PT) with focus on security. Access to network devices will be restricted to only allow cryptographic Secure Shell (SSH) for remote login. Routers and switches will also be restricted to only allow connection from the IP address of the administrator's PC to ensure only authorized access. Authentication, authorization, and accounting (AAA) will be implemented on the network devices together with Access Control Lists (ACLs) for control and to only allow necessary access.

The project will also include a secure wireless network connection for employees using laptops. The corporate wireless network will be restricted to company-owned devices, while a separate VLAN will be setup to allow guest access to the Internet.

Network segmentation with Virtual Local Area Networks (VLANs) will be implemented using Router-on-a-Stick Inter-VLAN routing to secure and allow separation between different departments. Access Control Lists (ACLs) will be applied to limit traffic between VLANs. This will ensure that only necessary communication is permitted. Furthermore, Port security will be configured on switches to prevent unauthorized devices from connecting to the network and Port Address Translation (PAT) will be used to hide internal IP addresses from external networks as every security measures are part of building a layered defense.

Using virtual machines (VMs) in VMware, a web server and a backup server will be installed with Linux Ubuntu Server operative systems (OSs) and configured with strict security measures. Only non-root users will be allowed to log in to the web server using SSH key-based authentication and password-based authentication will be disabled.

OS hardening will be implemented by without unnecessary services and applying best security practices to minimize vulnerabilities. For demonstration, OS hardening is mainly applied and discussed on the web server through the server setup and in a dedicated section called OS Hardening.

Data backup and restoration will be demonstrated with rsync between the web server and the backup server followed by setting up automatic backups of critical company files.

While firewall implementation is not included in this demonstration due to limited expertise in Cisco firewall configurations, other security mechanisms such as ACLs, VLAN segmentation, and port security will be used to mitigate potential threats. Redundancy is neither implemented in this project to maintain clarity, though its importance is widely acknowledged to ensure high availability and network fault tolerance.

Aligned with the CIA triad, the project focuses on confidentiality by restricting unauthorized access, integrity by implementing secure authentication and controlled access, and availability is demonstrated by ensuring critical data protection through backups.

1.2 Structure

The report is divided into four parts; an introduction, a main part and solution, a conclusion, and a part for references. The main part and solution is further divided into a head sections for the network design demonstrated with Cisco's Packet Tracer, a head section for Server configurations, a head section to demonstrate backup and restoration of the Web Server, followed by a head section for OS hardening. Commands executed on Cisco network devices are written in bold and the router and switch configurations are distinguished by shaded backgrounds, orange for Router1 and green for Switch1. Command executions done on end devices in VMware are shaded with a gray background for clarity.

1.3 Assumptions

It is assumed the name of the company is LNXE-Corp as there are a mismatch between the project name and the description of this project. However, this would be resolved in real life for clarity.

The router implemented in this setup is assumed either a Cisco ISR 4000-series or newer to handle modern service implementations like Cisco Umbrella.

It is also assumed that all employees work on the same floor and within tolerable distance to the two Wireless Access Points (WAPs) and an additional WLAN controller (WLC) is not needed. Bring Your Own Device (BYOD) is not implemented in the production environment and devices not own by the company are only allowed on the segmented wireless guest network. The Admin PC and the Client PCs is updated and has SSH installed and enabled.

All critical data for backup is stored recursively in the /var/dev/lnxecorp directory on the Web Server.

1.4 Expected Challenges

One major expected challenge is the limited features in Packet Tracer (PT). It is therefore some configurations, like dns server on a router and implementing services like Cisco Umbrella, that are described out from Cisco's configuration guides.

There is also a challenge with limited expertise in how a company's client-webserver relationship works as an platform for the developers in the backend.

Another expected challenge is how to balance the demonstration between Packet Tracer and client-servers in VMware. It might be easy to add a server for every service you want in PT since the services are pre-installed, however it could turn out to be to demanding to showcase the same setup with virtual machines, if necessary. It will also be more challenging with proof-of-concept in PT than with virtual machines in VMware, and most proof has to be viewed through the commands in the setup.

Balancing usability and security may also be challenging. If securing the network with too strong values on the like of port-security maximum mac-address es and shutdown violation, the usability might suffer.

Finally, it may be challenging to only cite highly credible sources for this report since there are some amazingly great communities and tutorials on the Internet where you may learn much more hands-on Linux configurations than many journals and books.

2 Main Part and Solution

2.1 Network Design

This section provides a comprehensive description of the network architecture and device configuration for the start-up company LNXE-Corp.

It is attempted to follow Cisco's Network Foundation Protection (NFP) framework as a strategic approach and break down the network device configurations and security implementations into the management plane, control plane, and data plane. To establish network functionality and practically coherent configurations, not all configuration steps are lined within the designated plane, however NFP is used for threat control and as a mitigation strategy.

2.1.1 Topology and End Device Initiations

2.1.1.1 Network Topology:

Topology of a Router-on-a-Stick network for Inter-VLAN Routing in Packet Tracer (see Figure 1).

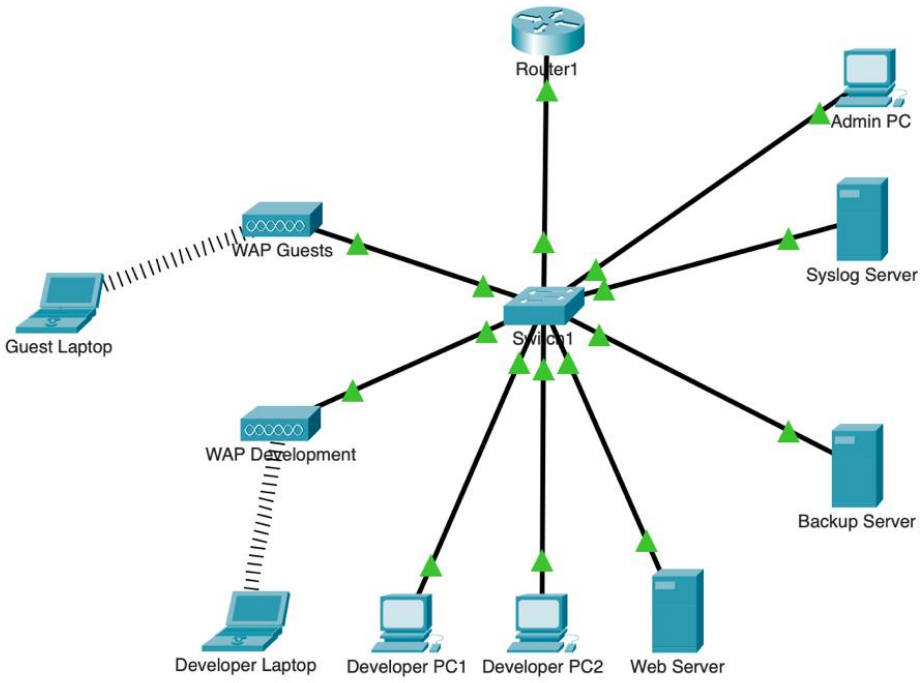

Figure 1: Network topology for LNXE-Corp (Bakke, 2025).

2.1.1.2 IP addressing and VLAN table:

IP Address Table

Device	Interface	IP Address	Subnet Mask	Gateway
Router1	Gig0/0	N/A	N/A	N/A
	Gig0/1.10	192.168.10.1	255.255.255.0	N/A
	Gig0/1.20	192.168.20.1	255.255.255.0	N/A
	Gig0/1.30	192.168.30.1	255.255.255.0	N/A
	Gig0/1.40	192.168.40.1	255.255.255.0	N/A
Switch1	VLAN 20	192.168.20.2	255.255.255.0	192.168.20.1
WAP Development	NIC	192.168.10.3	255.255.255.0	192.168.10.1
WAP Guests	NIC	192.168.40.3	255.255.255.0	192.168.40.1
Web Server	NIC	192.168.10.10	255.255.255.0	192.168.10.1
Developer PCs	NIC	192.168.10.11 - 192.168.10.20	255.255.255.0	192.168.10.1
Developer Laptops(DHCP)	NIC	192.168.10.100 - 192.168.10.199	255.255.255.0	192.168.10.1
Admin PC	NIC	192.168.20.10	255.255.255.0	192.168.20.1
Syslog Server	NIC	192.168.20.11	255.255.255.0	192.168.20.1
Backup Server	NIC	192.168.30.10	255.255.255.0	192.168.30.1

Table 1: IP Address Table (Bakke, 2025).

Switch1 VLAN Table

VLAN ID	VLAN Name	Purpose	Interface Assigned
10	Development-VLAN	Developer network	Fa0/5 - 16
20	Management-VLAN	Management network	Fa0/1 - 2
30	Backup-VLAN	Data backup network	Fa0/3
40	Guest-VLAN	Guest network	Fa0/4
177	Native-VLAN	For untagged traffic	
999	Unused-VLAN	For unused ports	Fa0/17 -24

Table 2: VLAN Table (Bakke, 2025).

2.1.1.3 Configure End Devices:

The IP Address table is used to assign static IP addresses and default gateways to the Web Server, Backup Server, System Server, Admin PC, and Developer PC1 and PC2. The same IP address as the default gateway should be used for the DNS address.

Laptops are configured to acquire dynamic IP addresses by enabling DHCP.

2.1.2 Router and Switch Configurations

2.1.2.1 Basic Setup:

Router1 and Switch1 are configured with usernames and a MD5 encrypted password for enabling configurations of the Terminal. Enforced password policy for security passwords are set to minimum 12 characters and all locally service passwords are obscured with a Type 7 encryption. A banner, signaling authorized personnel only, is added together with the domain of the company. An user for the administrator is created with full administrative access and a MD5 encrypted password.

```
Router>enable
Router#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#hostname Router1
Router1(config)#enable secret ep1enpa$$
Router1(config)#security passwords min-length 12
Router1(config)#service password-encryption
Router1(config)#banner motd #Unauthorized Access is Prohibited#
Router1(config)#ip domain-name lnxecorp.com
Router1(config)#username morpheus privilege 15 secret nap0130npa$$
```

```
Switch>enable
Switch#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)#hostname Switch1
Switch1(config)#enable secret ep1enpa$$
Switch1(config)#service password-encryption
Switch1(config)#banner motd #Unauthorized Access is Prohibited#
Switch1(config)#ip domain-name lnxecorp.com
Switch1(config)#username morpheus privilege 15 secret nap0130npa$$
```

2.1.2.2 Implement AAA Service:

It is considered adequately to configure local databases on Router1 and Switch1 for authentication, authorization, and accounting (AAA) for this project based on the company's small size and limited administrators. Role-based access control (RBAC) will be handled with privilege levels between 1 and 15 where the latter is the highest privilege. AAA will be used in this project as a security measure for logging and access. By implementing AAA, it is also possible to avoid brute-force attacks by configuring a maximum number of failed login attempts before a user gets locked out (Cisco, 2021).

```
Router1(config)#aaa new-model
Router1(config)#aaa local authentication attempts max-fail 5
Router1(config)#aaa authentication login default local
```

```
Switch1(config)#aaa new-model
Switch1(config)#aaa local authentication attempts max-fail 5
Switch1(config)#aaa authentication login default local
```

2.1.2.3 Configure Service Line Security:

The service lines passwords are by now enforced with an Type 7 algorithm, however, this encryption is not considered secure and can be decrypted easily. It is therefore recommended to enhance security for the service lines further with AAA login. To secure the physical out-of-band management ports for the console line on Router1 and Switch1 and the auxiliary (aux) line on Router1, they are first configured with passwords and then login authentication using the local AAA database. By implementing AAA login for the service lines, the MD5 encrypted user password is required instead of the weaker line password. The aux line is also disabled for CLI session access for now as it will not be in use.

To further secure and restrict in-band management on the VTY lines and to only allow SSH access from the Admin PC, a standard access list is configured and deployed to the line so only the Admin PC's IP address is permitted and any other IP address is denied. An executive timeout is also configured for security so the SSH session is terminated after five minutes of inactivity.

```
Router1(config)#line console 0
Router1(config-line)#password ep1conpa$$
Router1(config-line)#login authentication default
Router1(config-line)#line aux 0
Router1(config-line)#no exec
Router1(config-line)#password ep1auxpa$$
Router1(config-line)#login authentication default
Router1(config-line)#exit
Router1(config)#ip access-list standard SSH_ONLY
Router1(config-std-nacl)#remark Admin Management ACL
Router1(config-std-nacl)#permit 192.168.20.10
Router1(config-std-nacl)#deny any
Router1(config-std-nacl)#line vty 0 4
Router1(config-line)#access-class SSH_ONLY in
Router1(config-line)#transport input ssh
Router1(config-line)#exec-timeout 5 0
Router1(config-line)#exit
```

```
Switch1(config)#line console 0
Switch1(config-line)#password ep1conpa$$
Switch1(config-line)#login authentication default
```

```
Switch1(config-line)#exit
Switch1(config)#ip access-list standard SSH_ONLY
Switch1(config-std-nacl)#remark Admin Management ACL
Switch1(config-std-nacl)# permit 192.168.20.10
Switch1(config-std-nacl)# deny any
Switch1(config-std-nacl)#line vty 0 15
Switch1(config-line)#access-class SSH_ONLY in
Switch1(config-line)#transport input ssh
Switch1(config-line)#exec-timeout 5 0
Switch1(config-line)#exit
```

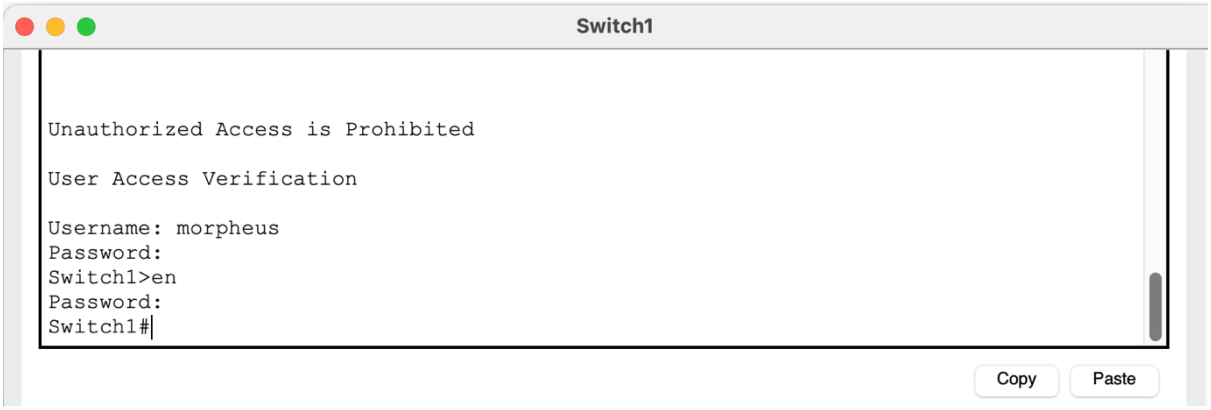

Figure 2: Verification of login security (Bakke, 2025)

2.1.2.4 Generate Encrypted SSH keys:

It is necessary to generate an encrypted key to enable SSH login through the VTY lines. The key is generated with the largest size of 4096 bits for the strongest security. Furthermore, the encryption used in SSH version 1 is not regarded as secure anymore and it is best security practice to ensure that version 2 is used. The SSH authentication is limited to be completed within 60 seconds with three attempt retries to protect against brute-force attacks.

```
Router1(config)#crypto key generate rsa
The name for the keys will be: Router1.lnxe-corp.local
Choose the size of the key modulus in the range of 360 to 4096 for your
General Purpose Keys. Choosing a key modulus greater than 512 may take
a few minutes.

How many bits in the modulus [512]: 4096
% Generating 1024 bit RSA keys, keys will be non-exportable...[OK]

Router1(config)#ip ssh version 2
*Mar 1 2:45:23.480: %SSH-5-ENABLED: SSH 1.99 has been enabled

Router1(config)#ip ssh authentication-retries 3
Router1(config)#ip ssh time-out 60
```

```
Switch1(config)#crypto key generate rsa
```

```
The name for the keys will be: Switch1.lnxecorp.com
Choose the size of the key modulus in the range of 360 to 4096 for your
General Purpose Keys. Choosing a key modulus greater than 512 may take
a few minutes.

How many bits in the modulus [512]: 4096
% Generating 1024 bit RSA keys, keys will be non-exportable...[OK]

Switch1(config)#ip ssh version 2
Switch1(config)#ip ssh authentication-retries 3
Switch1(config)#ip ssh time-out 60
```

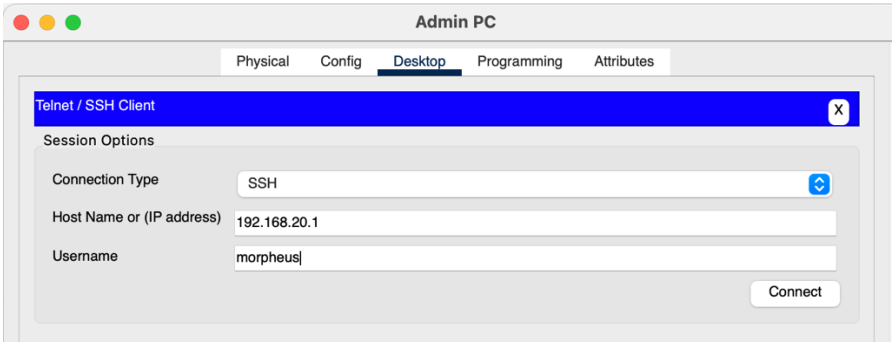

Figure 3: SSH login from Admin PC: (Bakke, 2025).

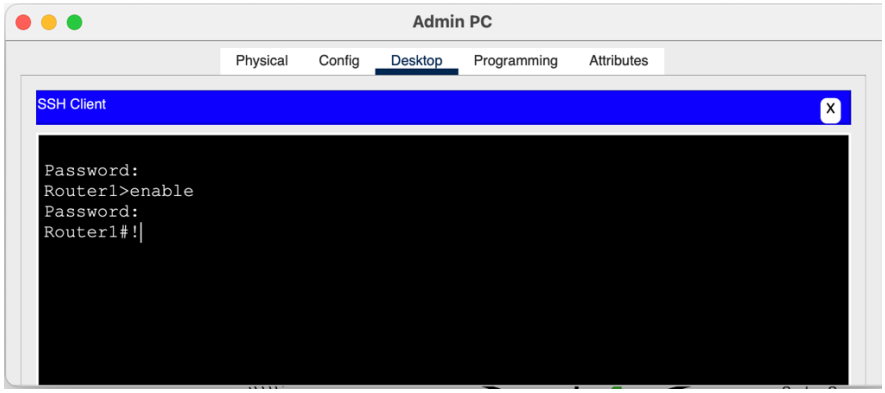

Figure 4: Verification of successful SSH login security: (Bakke, 2025).

2.1.2.5 Create sub-interfaces:

The interface connecting Router1 and Switch1 is segregated into sub-interfaces with VLAN tagging using 802.1Q encapsulation to improve security, access control, and management of network traffic.

```
Router1(config)#interface GigabitEthernet0/0/1.10
Router1(config-subif)#description Development VLAN
Router1(config-subif)#encapsulation dot1Q 10
Router1(config-subif)#ip address 192.168.10.1 255.255.255.0
Router1(config-subif)#interface GigabitEthernet0/0/1.20
Router1(config-subif)#description Management VLAN
```

```
Router1(config-subif)#encapsulation dot1Q 20
Router1(config-subif)#ip address 192.168.20.1 255.255.255.0
Router1(config-subif)#interface GigabitEthernet0/0/1.30
Router1(config-subif)#description Backup VLAN
Router1(config-subif)#encapsulation dot1Q 30
Router1(config-subif)#ip address 192.168.30.1 255.255.255.0
Router1(config-subif)#interface GigabitEthernet0/0/1.40
Router1(config-subif)#description Guest VLAN
Router1(config-subif)#encapsulation dot1Q 40
Router1(config-subif)#ip address 192.168.40.1 255.255.255.0
Router1(config-subif)#interface GigabitEthernet0/0/1
Router1(config-if)#description Trunk to Switch1
Router1(config-if)#no shutdown
Router1(config-if)#exit
```

2.1.2.6 Configure DHCP:

While it is decided to configure all stationary devices on the company's network with static IP addresses for management control, portable devices connected wireless will take use of dynamic hosting. Two wireless access points (WAP) are planned. One for company owned laptops for developers using VLAN10, and one WAP for guests and non-business traffic isolated on VLAN40.

```
Router1(config)#ip dhcp excluded-address 192.168.10.1 192.168.10.99
Router1(config)#ip dhcp excluded-address 192.168.10.200 192.168.10.255
Router1(config)#ip dhcp pool VLAN10_POOL
Router1(dhcp-config)#network 192.168.10.0 255.255.255.0
Router1(dhcp-config)#default-router 192.168.10.1
Router1(dhcp-config)#dns-server 192.168.10.1
Router1(dhcp-config)#exit
Router1(config)#ip dhcp excluded-address 192.168.40.1 192.168.40.100
Router1(config)#ip dhcp pool VLAN40_POOL
Router1(dhcp-config)#network 192.168.40.0 255.255.255.0
Router1(dhcp-config)#default-router 192.168.40.1
Router1(dhcp-config)#dns-server 192.168.40.1
Router1(dhcp-config)#exit
```

2.1.2.7 Configure DNS:

Packet Tracer does not have the DNS server features for router integration implemented, it is however a functionality in the real-world and it is decided to setup Router1 as a caching and forwarding DNS server with recursive DNS resolution for the servers in the network. The external DNS server is linked up to Cisco's Umbrella DNS resolvers for secure inspections. To enable the DNS service and domain lookup on Router1:

```
Router1(config)#ip dns server
Router1(config)#ip domain lookup
Router1(config)#ip name-server 208.67.222.222
Router1(config)#ip name-server 208.67.222.220
```

```
Router1(config)#ip host webserver 192.168.10.10
Router1(config)#ip host backupserver 192.168.30.10
```

2.1.2.8 Secure DNS using Umbrella:

The Umbrella Integration feature, available on Cisco 4000 Series ISRs, offer a cloud-based security service by inspecting the DNS query that is sent to the DNS server. Policies are defined in the Cisco Umbrella portal to either allow or deny traffic towards your domain. DNS packets sent between the router and the Umbrella DNS resolver may also be encrypted (Cisco, 2018).

Import CA certificate to the trust pool of the router.

```
Router1(config)#crypto pki trustpool import url
http://www.cisco.com/security/pki/trs/ios.p7b
```

A validation message is shown when the import succeeded. An alternatively method is to download the certificate and copy it to the router's flash if direct download to the router causes trouble. Log into the Umbrella portal https://login.umbrella.com/ to generate the API token from the Cisco Umbrella registration server. Navigate to Admin on the left sidebar, select API Keys, and choose Legacy Network Devices if the router is a 4000 series, or Umbrella Network Devices for non-legacy devices. Generate and copy the Token for use in configuring the router.

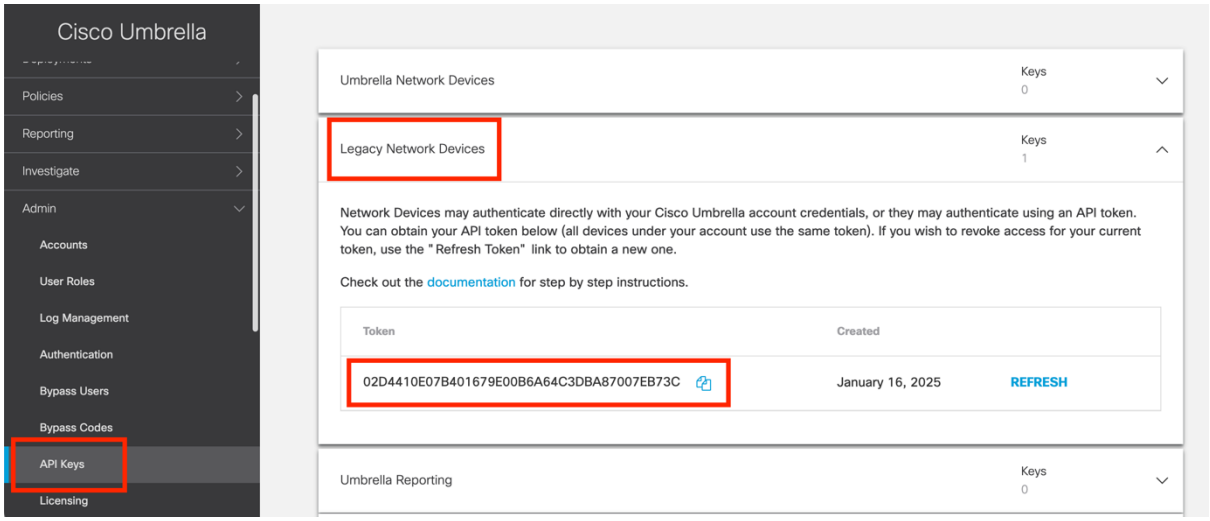

Figure 5: Token generated in Umbrella Portal (Bakke, 2025).

Back on Router1, it is optional to configure local domains as these DNS queries will not be forwarded to Umbrella DNS resolvers.

```
Router1(config)#parameter-map type regex dns_bypass
Router1(config)#pattern www.lnxecorp.com
Router1(config)#pattern *.lnxecorp.com
```

Now you need to set up a global Umbrella parameter map with the token that you got from the Umbrella portal and add the optional local domain bypass configured in the previous step. When using the global command, values for dnscrypt, resolver IP, and a public key are auto-populated for enhanced security.

```
Router1(config)#parameter-map type umbrella global
Router1(config-profile)#token 02D4410E07B401679E00B6A64C3DBA87007EB73C
Router1(config-profile)#local-domain dns_bypass
Router1(config-profile)#exit
```

Assign `umbrella out` on the ISP directed interface and enable `umbrella in` on all the interfaces that needs redirecting DNS traffic to the Umbrella cloud. Each interface should include a tag so policies can be configured based on the tag in the Umbrella portal. Tags are, in this case, demonstrated to correspond to the different sub-interfaces for each VLAN.

```
Router1(config)#interface GigabitEthernet0/0/0
Router1(config-if)#umbrella out
Router1(config-if)#interface GigabitEthernet0/0/1.10
Router1(config-if)#umbrella in Development
Router1(config-if)#interface GigabitEthernet0/0/1.20
Router1(config-if)#umbrella in Management
Router1(config-if)#interface GigabitEthernet0/0/1.40
Router1(config-if)#umbrella in Guests
Router1(config-if)#exit
```

2.1.2.9 Configure NTP:

Using pool.ntp.org as a Network Time Protocol (NTP) server does not really secure the company's network, however it helps ensure reliable time synchronization between devices, which then indirectly supports security and analysis. Adding a VPN to the boarder edge of the network for secure data-in-transit over WAN should render this configuration more secure.

```
Router1(config)#ntp server 23.150.41.122
Router1(config)#ntp server 69.89.207.199
```

2.1.2.10 Configure Logging:

Logging may not be considered as a methodic to secure a network, however it may be crucial for investigation and audit in security breaches. It is therefore decided to demonstrate and centralize logs on a dedicated server for network analysis. This implementation will be limited to the network topology in PT and not demonstrated in VMware.

Logging is enabled on Router1 and Switch1 in Packet Tracer and sent to the Syslog Server. Setting the trap level to 6 limits messages sent to the Syslog Server to informational and more severe logs and leave debugging logs. Timestamps are set to milliseconds and directed to the Syslog Server.

```
Router1(config)#logging on
Router1(config)#logging trap 6
Router1(config)#service timestamps log datetime msec
Router1(config)#logging host 192.168.20.11
```

```
Switch1(config)#logging on
Switch1(config)#logging trap 6
Switch1(config)#service timestamps log datetime msec
Switch1(config)#logging host 192.168.20.11
```

The syslog traffic is not considered necessary of encryption in this setup, however it is secured by using the Management VLAN just for management traffic. Additionally to the network devices, the Web Server and the Backup Server should also send logs to the Syslog Server and this can be done using the rsyslog service on the Ubuntu servers. Simple Network Management Protocol (SNMP) or more advanced monitoring services are not included in this report as it is not directly within the projects objective, however it should be used alongside the detailed syslogs to enable alerts and real-time tracking of system events and security incidents.

2.1.2.11 Configure Access Control:

Segmenting the network into VLANs enable defined filtering of communication between the different VLANs. Access Control Lists (ACLs) can be enforced on individual interfaces to restrict unwanted access or allow necessary traffic.

First of all, in this setup, devices with wireless connection on VLAN 10 and 40 need access to communicate with the DHCP server on Router1 via the UDP ports 67 and 68 to obtain dynamic IP addresses. Two ACLs are configured for these VLANs.

The first ACL is named VLAN10_RESTRICTIONS for control of communication on VLAN 10. The Web Server and the Backup Server respectively, will communicate using SSH, and two entries are added only for SSH communication both ways. SSH communication is also allowed between the Web Server and all other devices on VLAN 10 for the developers in the company. Any other communication from VLAN 10 to the other VLANs are then denied, while Internet connection is allowed using HTTPS. Any other traffic is denied at the end of the ACL. Finally, the ACL is placed on the incoming sub-interface GigabitEthernet0/0/1.10.

```
Router1(config)#ip access-list extended VLAN10_RESTRICTIONS
Router1(config-ext-nacl)#permit udp any host 192.168.10.1 eq 67
Router1(config-ext-nacl)#permit udp any host 192.168.10.1 eq 68
Router1(config-ext-nacl)#permit tcp host 192.168.10.10 host 192.168.30.10
eq 22
Router1(config-ext-nacl)#permit tcp host 192.168.30.10 host 192.168.10.10
eq 22
Router1(config-ext-nacl)#permit tcp 192.168.10.0 0.0.0.255 host
192.168.10.10 eq 22
Router1(config-ext-nacl)#permit tcp host 192.168.10.10 192.168.10.0
0.0.0.255 eq 22
Router1(config-ext-nacl)#deny ip 192.168.10.0 0.0.0.255 192.168.20.0
0.0.0.255
Router1(config-ext-nacl)#deny ip 192.168.10.0 0.0.0.255 192.168.30.0
0.0.0.255
Router1(config-ext-nacl)#deny ip 192.168.10.0 0.0.0.255 192.168.40.0
0.0.0.255
Router1(config-ext-nacl)#permit ip 192.168.10.0 0.0.0.255 any eq 443
Router1(config-ext-nacl)#deny ip 192.168.10.0 0.0.0.255 any
Router1(config-ext-nacl)#interface GigabitEthernet0/0/1.10
Router1(config-if)#ip access-group VLAN10_RESTRICTIONS in
Router1(config-if)#exit
```

The second ACL called GUEST_RESTRICTIONS is created here to filter communication on the untrusted VLAN 40. All communication is denied to the other VLANs, while only HTTPS traffic for the Internet is allowed. Any other communication is denied and the ACL is placed on the GigabitEthernet0/0/1.40 sub-interface.

```
Router1(config)#ip access-list extended GUEST_RESTRICTIONS
Router1(config-ext-nacl)#permit udp any host 192.168.40.1 eq 67
Router1(config-ext-nacl)#permit udp any host 192.168.40.1 eq 68
Router1(config-ext-nacl)#deny ip 192.168.40.0 0.0.0.255 192.168.10.0
0.0.0.255
Router1(config-ext-nacl)#deny ip 192.168.40.0 0.0.0.255 192.168.20.0
0.0.0.255
Router1(config-ext-nacl)#deny ip 192.168.40.0 0.0.0.255 192.168.30.0
0.0.0.255
Router1(config-ext-nacl)#permit ip 192.168.40.0 0.0.0.255 any eq 443
Router1(config-ext-nacl)#deny ip 192.168.10.0 0.0.0.255 any
Router1(config-ext-nacl)#interface GigabitEthernet0/0/1.40
Router1(config-subif)#ip access-group GUEST_RESTRICTIONS in
Router1(config-subif)#exit
```

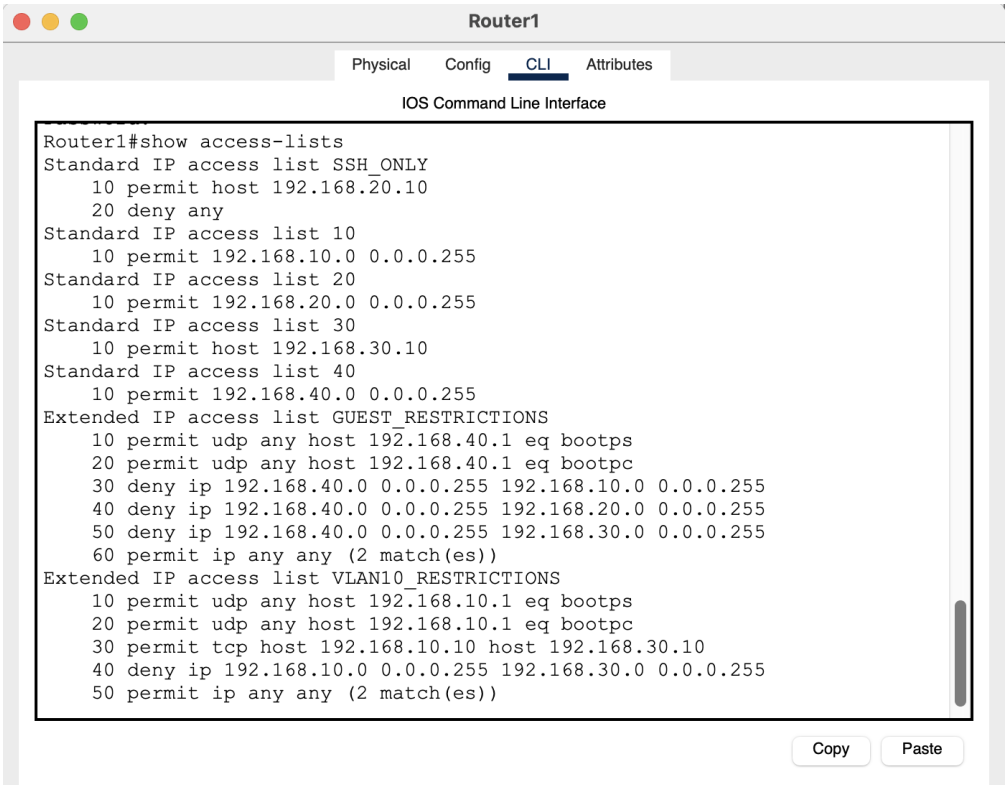

Figure 6: Verification of ACLs on Router1 (Bakke, 2025).

2.1.2.12 Create VLANs:

Virtual Local Area Networks (VLANs) are configured on Switch1 to separate network traffic. The VLANs are configured accordantly to the VLAN table in section 2.1.1.2. VLAN 10 isolates critical development team traffic between the Web Server and the developers PCs and reducing the risk of any unauthorized access to their data. VLAN 20 is restricted for management and syslog and VLAN 30 is solitary for backup traffic between the Web Server and the Backup Server. VLAN 40 is set up to provide isolated Internet access for untrusted traffic. The Native VLAN 177 is used as a default VLAN for untagged traffic to ensure that management and control traffic remains secure. Finally, VLAN 999 is created to store any ports that are not in use.

```
Switch1(config)#vlan 10
Switch1(config-vlan)#name Development-VLAN
Switch1(config-vlan)#vlan 20
Switch1(config-vlan)#name Management-VLAN
Switch1(config-vlan)#vlan 30
Switch1(config-vlan)#name Backup-VLAN
Switch1(config-vlan)#vlan 40
```

```
Switch1(config-vlan)#name Guest-VLAN
Switch1(config-vlan)#vlan 177
Switch1(config-vlan)#name Native-VLAN
Switch1(config-vlan)#vlan 999
Switch1(config-vlan)#name Unused-VLAN
Switch1(config-vlan)#exit
```

2.1.2.13 Configure Management VLAN:

The switch is also configured with an IP address for management.

```
Switch1(config)#interface vlan 20
Switch1(config-if)#ip address 192.168.20.2 255.255.255.0
Switch1(config-if)#no shutdown
Switch1(config-if)#exit
Switch1(config)#ip default-gateway 192.168.20.1
```

2.1.2.14 Configure Trunk Ports:

A trunk port is configured to allow multiple VLANs to use the same physical interface facing Router1. The "nonegotiate" command prohibits automated trunk negotiation so only defined ports is allowed to trunk.

```
Switch1(config)#interface GigabitEthernet0/1
Switch1(config-if)#description Trunk to Router1
Switch1(config-if)#switchport mode trunk
Switch1(config-if)#switchport trunk native vlan 177
Switch1(config-if)#switchport trunk allowed vlan 10,20,30,40
Switch1(config-if)#switchport nonegotiate
Switch1(config-if)#exit
```

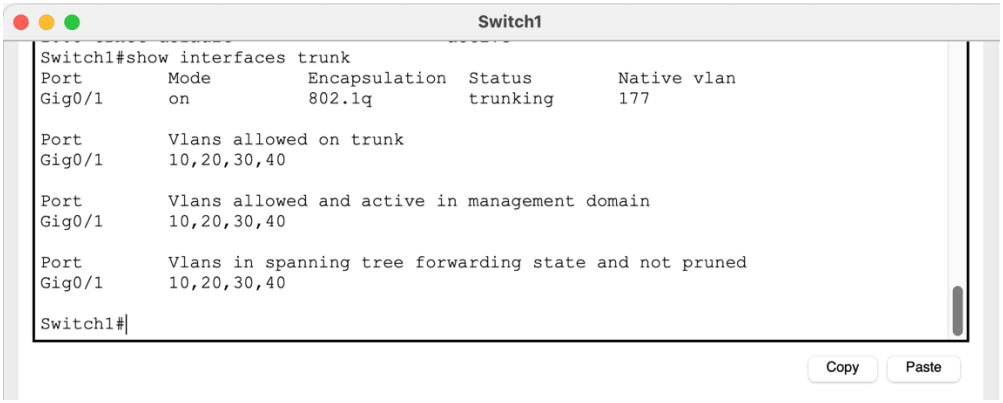

Figure 7: Verification of trunk ports on Switch1 (Bakke, 2025).

2.1.2.15 Configure Access Ports:

All ports set to carry single VLAN traffic are configured to access mode allowing its respectively VLAN.

```
Switch1(config)#interface FastEthernet0/1
Switch1(config-if)#description Admin PC
Switch1(config-if)#switchport mode access
Switch1(config-if)#switchport access vlan 20
Switch1(config-if)#interface FastEthernet0/2
Switch1(config-if)#description Syslog Server
Switch1(config-if)#switchport mode access
Switch1(config-if)#switchport access vlan 20
Switch1(config-if)#interface FastEthernet0/3
Switch1(config-if)#description Backup Server
Switch1(config-if)#switchport mode access
Switch1(config-if)#switchport access vlan 30
Switch1(config-if)#interface FastEthernet0/4
Switch1(config-if)#description Guest-WAP
Switch1(config-if)#switchport mode access
Switch1(config-if)#switchport access vlan 40
Switch1(config-if)#interface FastEthernet0/5
Switch1(config-if)#description Development-WAP
Switch1(config-if)#switchport mode access
Switch1(config-if)#switchport access vlan 10
Switch1(config-if)#switchport nonegotiate
Switch1(config-if)#interface FastEthernet0/6
Switch1(config-if)#description Web Server
Switch1(config-if)#switchport mode access
Switch1(config-if)#switchport access vlan 10
Switch1(config-if)#interface range FastEthernet0/7 - 16
Switch1(config-if-range)#description Development PC
Switch1(config-if-range)#switchport mode access
Switch1(config-if-range)#switchport access vlan 10
Switch1(config-if-range)#exit
```

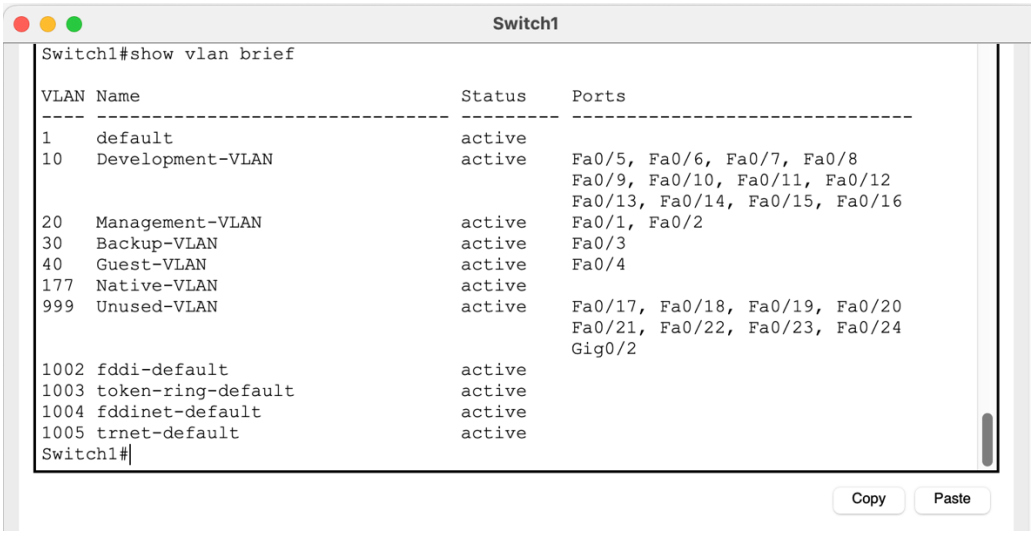

Figure 8: Verification of VLANs on Switch1 (Bakke, 2025).

2.1.2.16 Secure Switch Ports:

Any unused ports on Switch1 are potentially entry points for exploitation and a security risk to the network. Unused ports are therefore placed in VLAN 999 and manually shut down.

```
Switch1(config)#interface range FastEthernet0/17 - 24, GigabitEthernet0/2
Switch1(config-if-range)#switchport mode access
Switch1(config-if-range)#switchport access vlan 999
Switch1(config-if-range)#shutdown
Switch1(config-if-range)#exit
```

Port security is a security measure that identifies devices on the switch's interfaces based on the source MAC address of Ethernet frames that is sent from a connected device. If a new device attempts to send frames to the switch interface, based on the configurations, the switch can take different actions from simply discarding the traffic as offending to shutting down the interface and putting it in an err-disabled state (Odom, 2016).

Here, Port Security is enabled on the access ports with a maximum of two MAC addresses which is dynamically saved when a new device is sending Ethernet frames on the interface. The maximum value could be configured to one single address for maximum security, however two is used to balance functionality through the network setup. Shutdown of the interface is used if a violation occur, while the protect value could be used if the network was monitored with SNMP. Additionally, Spanning Tree Protocol (STP) security configurations are configured to secure the network from rouge switches.

```
Switch1(config)#interface range FastEthernet0/1 - 16
Switch1(config-if-range)#switchport port-security
Switch1(config-if-range)#switchport port-security maximum 2
Switch1(config-if-range)#switchport port-security mac-address sticky
Switch1(config-if-range)#switchport port-security violation shutdown
Switch1(config-if-range)#spanning-tree portfast
Switch1(config-if-range)#spanning-tree bpduguard enable
Switch1(config-if-range)#exit
```

To secure the switch ports with DHCP Snooping prevents man-in-the-middle and IP spoofing attacks with rogue DHCP servers from providing various misleading IP configurations to the legitimate clients on the network. Dynamic ARP Inspection (DAI) is dependent on DHCP Snooping and blocks Address Resolution Protocol (ARP) spoofing attacks by validating ARP packets against trusted DHCP bindings.

Global DHCP Snooping is enabled by default on routers and switches tested in Packet Tracer and this is preventing DHCP traffic from being inspected across any VLANs. It is therefore required to disable DHCP Snooping before re-enabling it to inspect the desired VLANs.

Router1 on the GigabitEthernet0/1 interface serves as the DHCP server and should be the only interface to be trusted for DHCP replies.

```
Switch1(config)#no ip dhcp snooping
Switch1(config)#interface GigabitEthernet0/1
Switch1(config-if)#ip dhcp snooping trust
Switch1(config-if)#ip arp inspection trust
Switch1(config-if)#interface range FastEthernet0/1 - 24
Switch1(config-if-range)#ip dhcp snooping limit rate 6
Switch1(config-if-range)#exit
Switch1(config)#ip dhcp snooping vlan 10,40
Switch1(config)#ip arp inspection vlan 10,20,30,40
Switch1(config)#ip arp inspection validate src-mac dst-mac ip
```

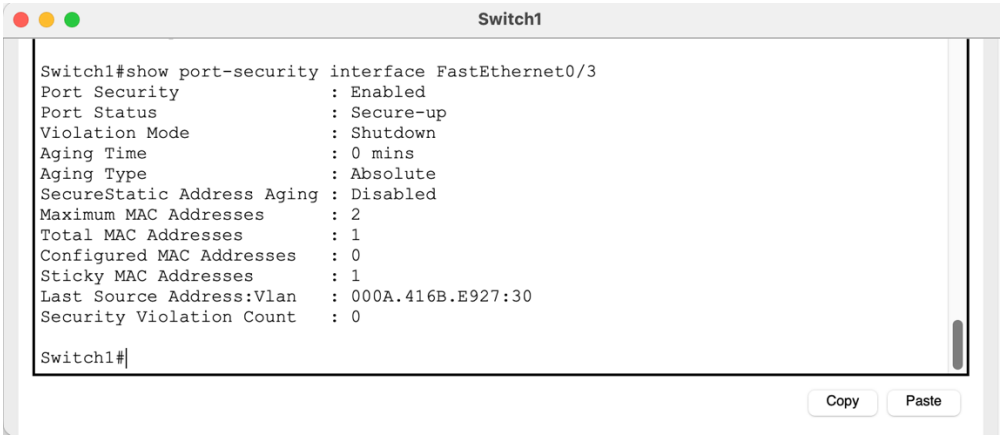

Figure 9: Verification of Port Security on Switch1 (Bakke, 2025).

2.1.2.17 Configure PAT:

PAT is configured to provide future Internet connection to multiple end devices in the company and PAT's function of converting public IP address to several private IP addresses helps conceal the private IP address of the internal devices from public.

```
Router1(config)#ip nat inside source list 1 interface GigabitEthernet0/0/0
overload
Router1(config)#access-list 10 permit 192.168.10.0 0.0.0.255
Router1(config)#access-list 20 permit 192.168.20.0 0.0.0.255
Router1(config)#access-list 30 permit 192.168.30.10 0.0.0.0
Router1(config)#access-list 40 permit 192.168.40.0 0.0.0.255
Router1(config)#interface GigabitEthernet0/0/1
Router1(config-if)#ip nat inside
Router1(config-if)#interface GigabitEthernet0/0/0
Router1(config-if)#ip nat outside
Router1(config-if)#exit
```

2.1.3 Wireless Access Point Configuration

It is decided to set up two separate wireless points (WAPs) in this project. One for developers using VLAN 10 and one for guests using VLAN 40. Setting up a guest network with strict network filtering is considered a security measure because would often like to provide guests or employees with non-company end devices to access the Internet without being able to interfere with sensitive data and compromise internal resources.

The wireless access points are configured with mostly the same setup. The WAP Guests is connected to the FastEthernet0/4 interface for VLAN 40 on Switch1, while WAP Development is connected to FastEthernet0/5 dedicated for VLAN 10. For demonstration in packet Tracer (see Figure), configurations for the WAP for VLAN 10 are done on Port 1 under Interface in the Config tab. To avoid broadcasting the company name, obfuscated SSID names are used as a low level security measure together with a strong Wi-Fi Protected Access 2 with pre shared key (WPA2-PSK) pass phrases used for authentication with Advanced Encryption Standard (AES) encryption.

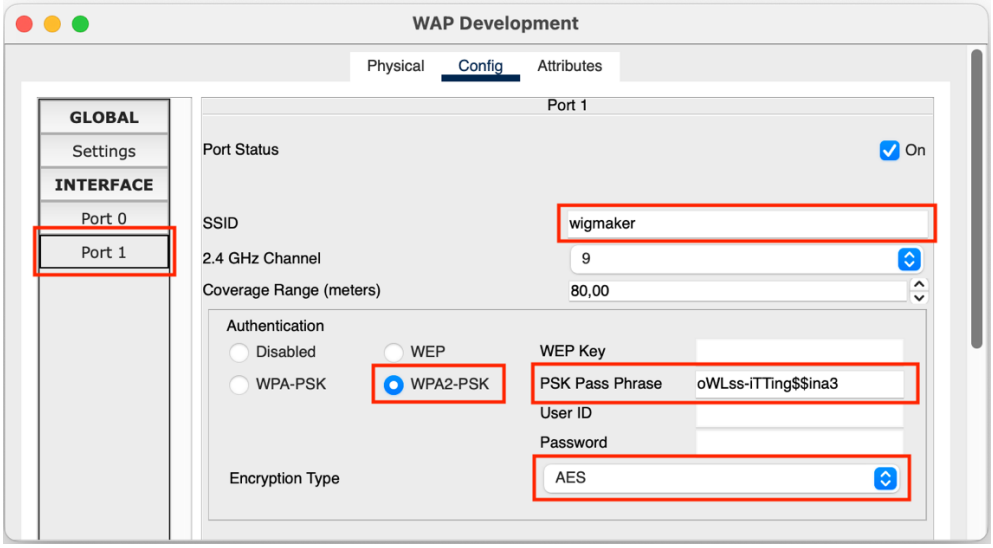

Figure 10: Wireless AP configuration (Bakke, 2025).

To demonstrate a successful network connection, a Developer Laptop equipped with a wireless NIC is selecting the SSID of the WAP Development as presented in Figure ... and entering the pre-shared pass phrase created in the previous step (see Figure).

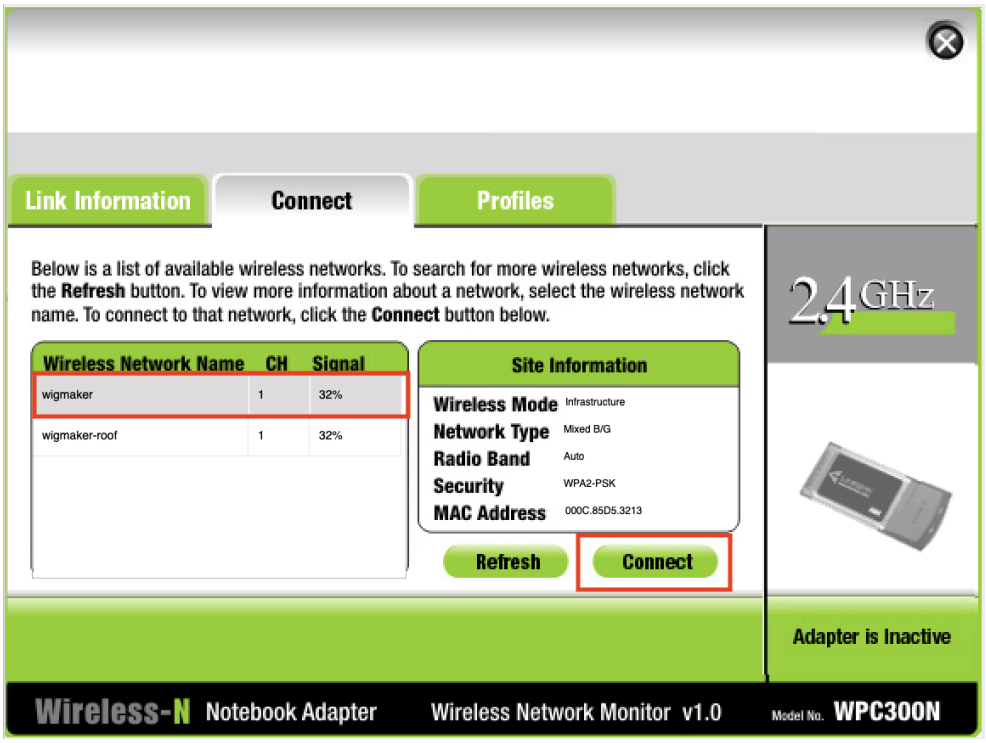

Figure 11: Wireless AP configuration (Bakke, 2025).

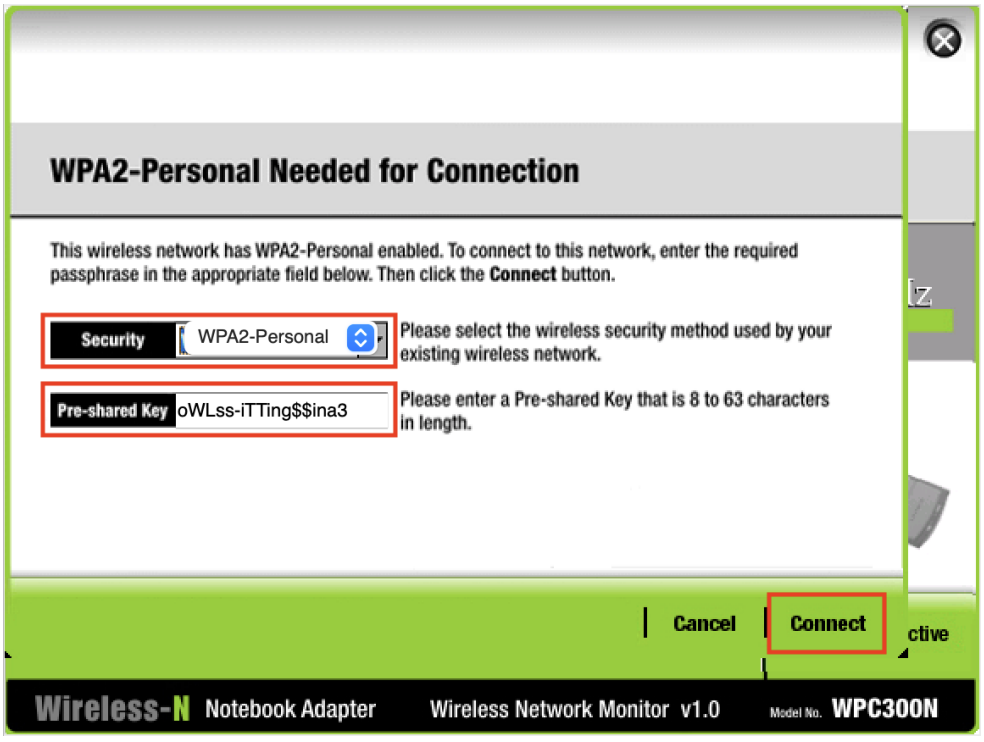

Figure 12: Wireless AP configuration (Bakke, 2025).

An encrypted connection is made and ensure DHCP is enabled on the Developer Laptop to acquire a dynamic IP address from the VLAN10_POOL as demonstrated in Figure and Figure .

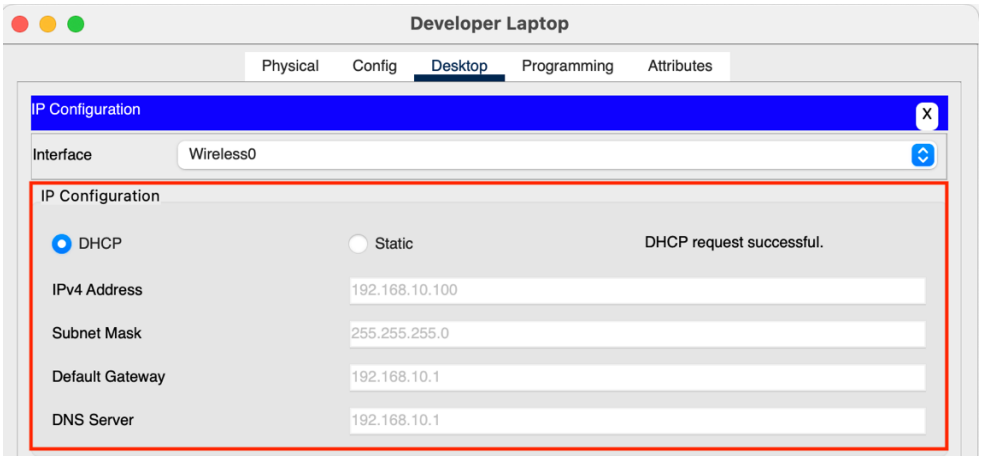

Figure 13: DHCP configuration on laptop (Bakke, 2025).

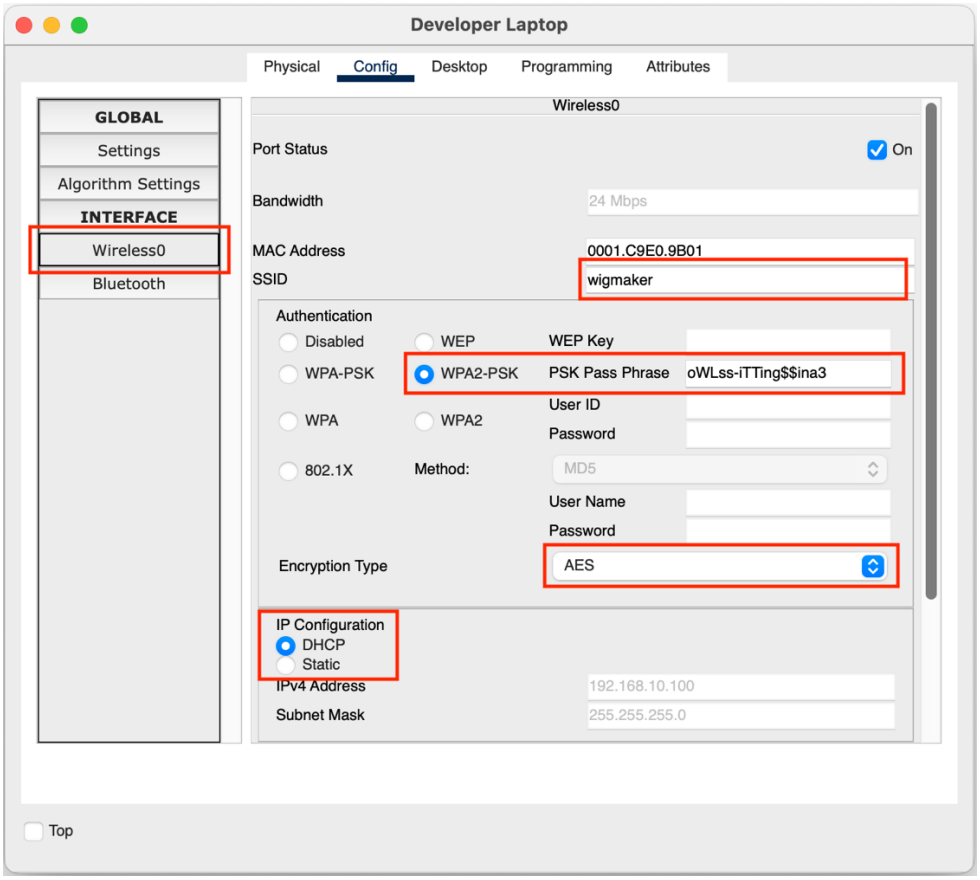

Figure 14: Setup for WAP connection on laptop (Bakke, 2025).

In a real-world scenario, wireless access points with the newer and more secure WPA3 protocol would be used for enhanced security. With WPA3, passwords can for instance not be cracked offline, meaning an attacker have to interact with the company's Wi-Fi connection for every password guess, which makes it much harder and time-consuming to crack (Gordon and Cohen, 2023).

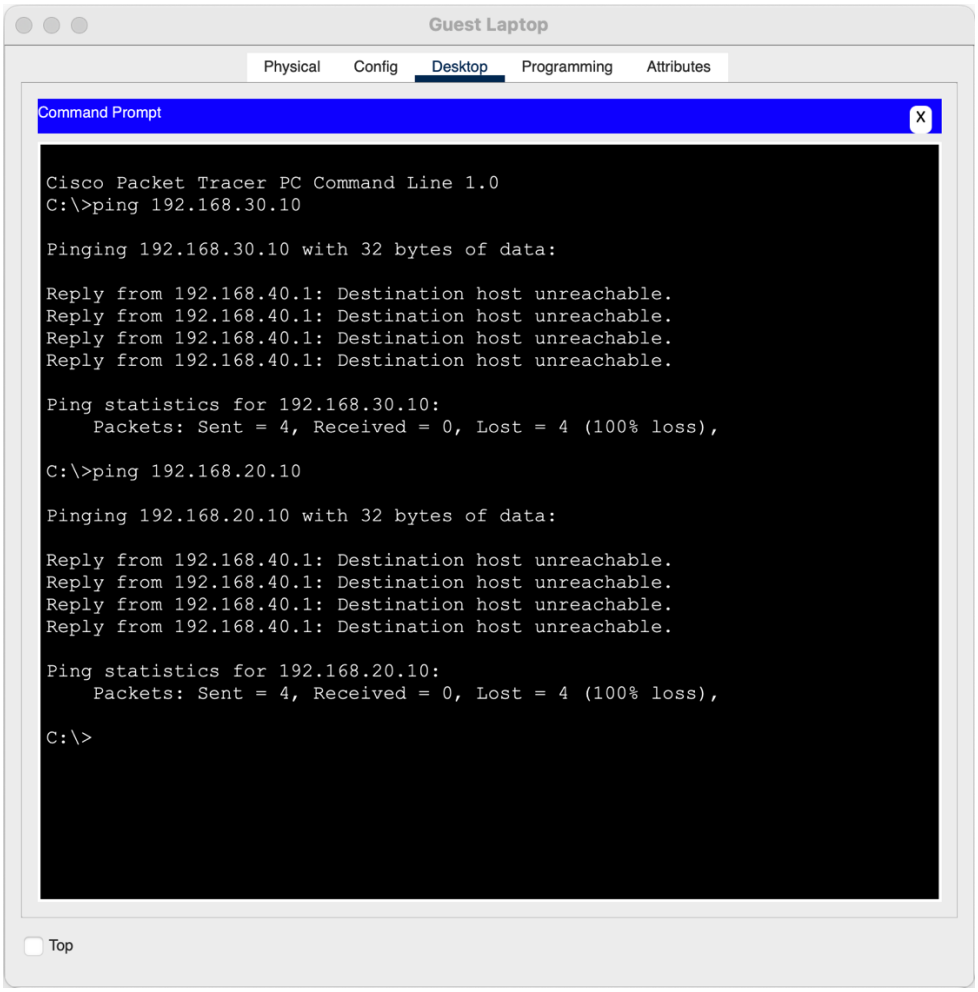

Figure 15: Verification of restricted access on guest VLAN (Bakke, 2025).

2.1.4 Firewall implementation and redundancy

A firewall at the border of our network would act as the first line of defense against external threats to the company's assets. A firewall monitors and controls incoming and outgoing network traffic and block any suspicious activity. Modern firewalls are capable of a great variety of security measures and should be part of any network. Many of the security implementations performed on the router in this project could be transferred to a firewall for centralized control and leave the router to route traffic.

Redundancy is also important for a healthy network infrastructure and redundancy would have been applied to every critical device in this design to preserve availability. Santos and Stuppi (2015) explain redundancy where if the network is poorly designed without any fault

tolerance and a device fails because of a mechanical or software failure and you do not have the failovers in place to continue to move traffic, your data plane is going to suffer.

2.2 Server Configurations

In this section, it is focused on demonstrating the setup and configurations for the Web Server and the Backup Server in VMware. The server names remain the same, however the IP addresses are changed as a result of the address pool available through PAT distribution on the host machine.

The new IP addresses are as follow:

Web Server : 192.168.130.220

Backup Server : 192.168.130.221

2.2.1 Web Server Setup

2.2.1.1 Ubuntu Server Installation:

When installing an Ubuntu server for specific services, it is best practice to choose the minimized version in the installation setup for easier OS hardening and reduced attack surface since just a set of specific services are installed and running by default (see Figure).

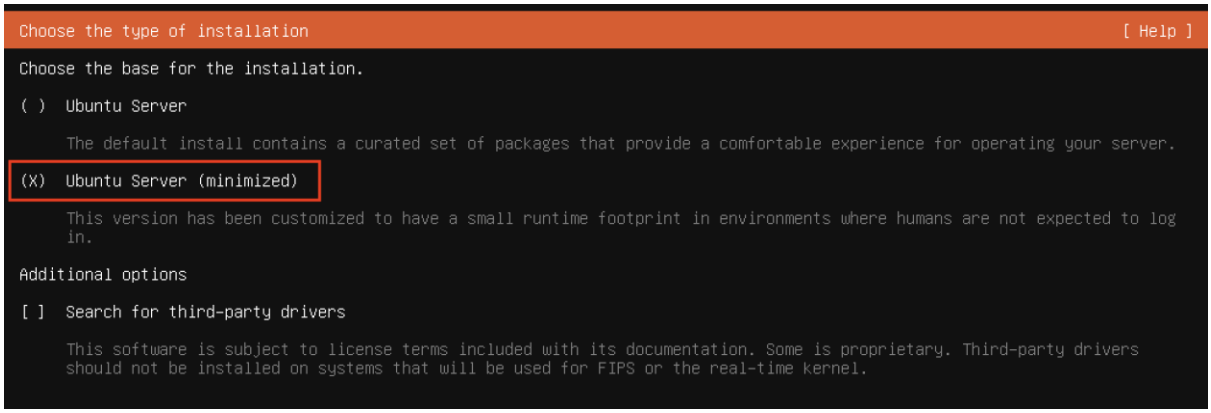

Figure 16: Server installation minimized (Bakke, 2025).

Set up the disk as an LVM group for logical partitioning and initiate encryption with Linux Unified Key Setup (LUKS) as highlighted in Figure .

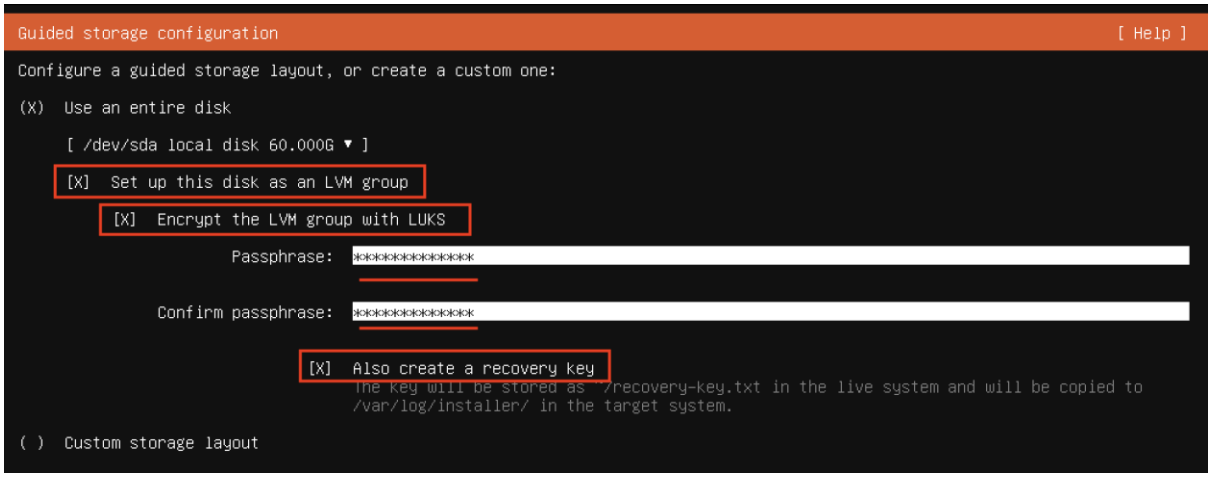

Figure 17: Server installation disk encryption (Bakke, 2025).

Fill in the storage and profile configurations after finishing the encryption setup. The username "morpheus" is used in this demonstration as a homage to a last century classic movie, however, best practices is to name users with admin privileges with uncommon names to avoid it becoming an easy target for brute-force or other credential attacks. Always use a strong password for authentication (see Figure).

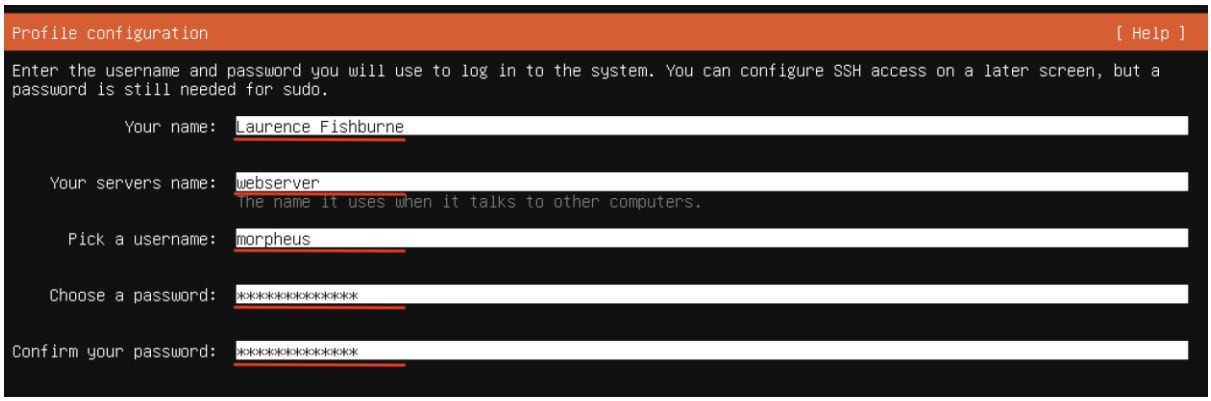

Figure 18: Server installation profile configuration (Bakke, 2025).

After user configurations, it is decided to sign up for a Ubuntu Pro Enterprise version for extended security updates and security features. Ubuntu Pro is further described in 2.4.7 Automated Security.

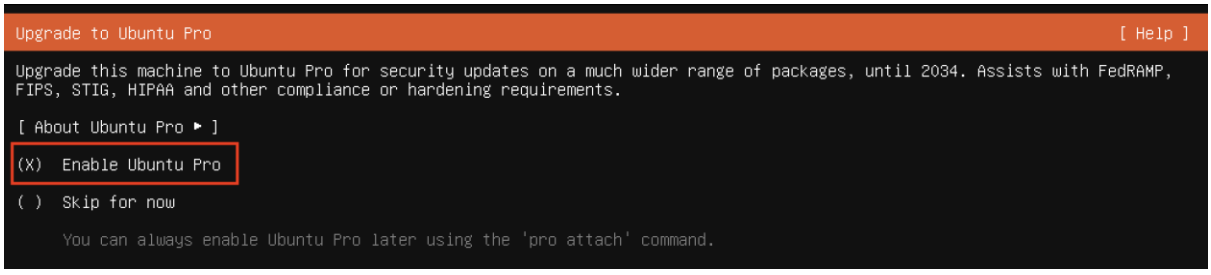

Figure 19: Server installation Ubuntu Pro upgrade (Bakke, 2025).

To register the server, enter the code via ubuntu.com/pro/attach or add the token generated from ubuntu.com/pro/dashboard to the configuration and proceed.

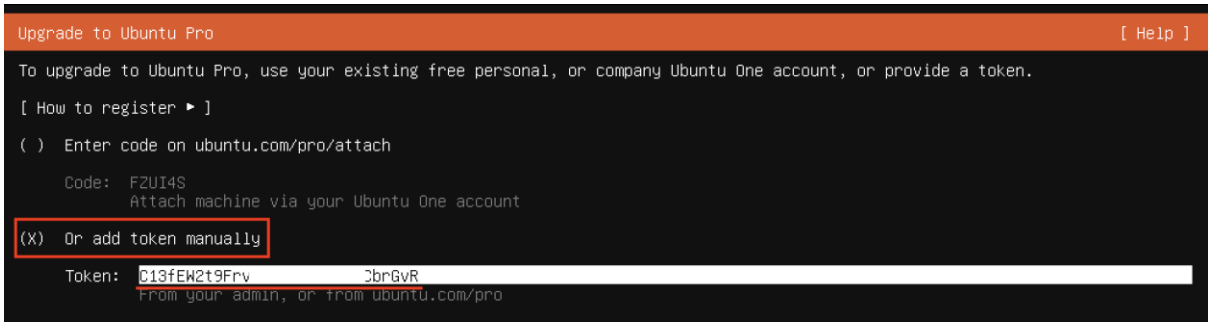

Figure 20: Server installation Ubuntu Pro token (Bakke, 2025).

When questioned, install OpenSSH server during the installation without importing any key at this moment. No additional server snaps is needed for this setup and you are done with the configurations and need to reboot.

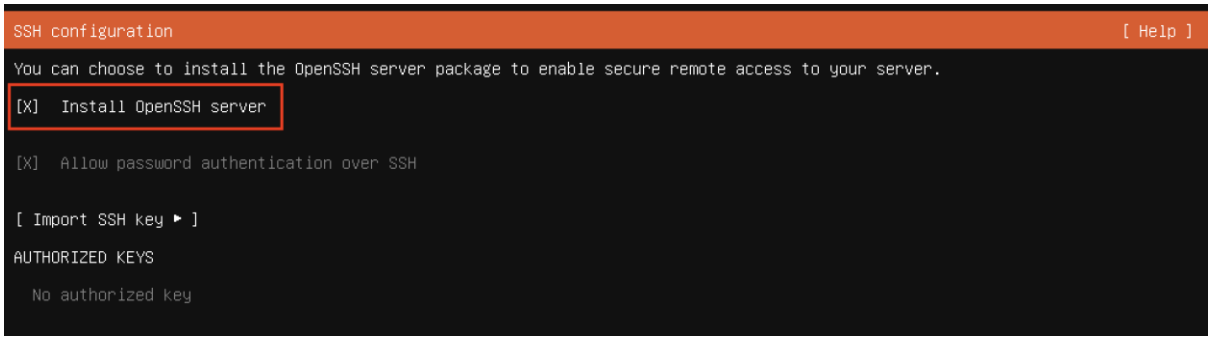

Figure 21: Server installation OpenSSH (Bakke, 2025).

2.2.1.2 Basic server initialization:

Using the minimize installation for the server helps control what services are running on the system and you have to add the necessary services rather than uninstall any unneeded programs for OS hardening.

Run this command to view all units that is now enabled in systemd on startup:

```
systemctl list-unit-files --state=enabled
```

It is also notable to view how few services are active and running from start with the minimized install version (see Figure):

```
systemctl --type=service --state=running
```

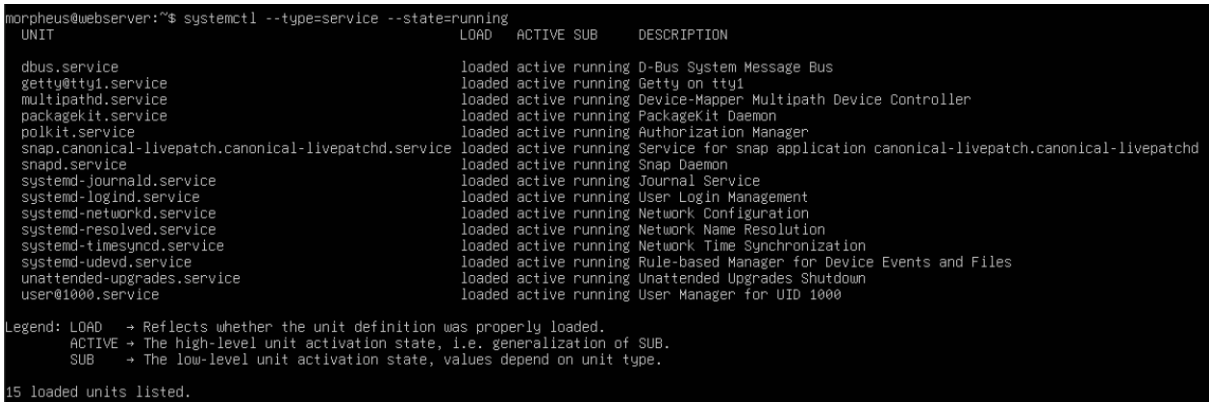

Figure 22: Running services on Web Server (Bakke, 2025).

Now, begin the initialization of the server by updating and upgrading all system packages installed on the server to their latest versions by running the following command:

```
sudo apt update && sudo apt upgrade -y
```

This ensures the server to run with the newest software and patches for known vulnerabilities and should be performed periodically and implemented in the company's security policy.

This is also a good moment to set the fully-qualified domain name (FQDN) for the server:

```
sudo hostnamectl set-hostname webserver.lnxecorp.com
```

To edit configuration files, install a text program, like nano with the command:

```
sudo apt install nano -y
```

2.2.1.3 Install and Configure Apache2:

When it comes to protecting the transfer of sensitive data, even if the web server in this project is configured for internal use and not facing the public Internet at the moment, it is still preferred to demonstrate a secure connection between a client and the Apache web server for layered defense.

Install Apache2 with the following command for web services on the server:

```
sudo apt install apache2 -y
```

For demonstration, a directory is created for the company's website in the default www directory:

```
sudo mkdir /var/www/lnxecorp
```

Set ownership and permissions for the root directory of the website to the Apache application. Setting the permission to 750 restricts access to only the owner and group:

```
sudo chown -R www-data:www-data /var/www/lnxecorp/
sudo chmod -R 750 /var/www/lnxecorp/
```

Create a simple index file for HTTPS demonstration:

Add the following text and save:

```
<html>
<head>
  <title> LNXE-Corp Web Server </title>
</head>
<body>
  <p> Up and running!
</body>
</html>
```

Set up a new virtual host for the company's domain by creating a new VirtualHost configuring file:

```
sudo nano /etc/apache2/sites-available/lnxecorp.conf
```

Write the following configurations to the file:

```
<VirtualHost *:80>
    ServerName lnxecorp.com
    ServerAlias www.lnxecorp.com
    DocumentRoot /var/www/lnxecorp

    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

Save and close the file when finished. It should also be noted that the important configuration in this enclosed project is the DocumentRoot path, while additional lines will be entered to secure the connection later.

To enable the website, the default Apache site should be disabled and the new virtual host configuration needs to be activated and reloaded with the following commands:

```
sudo a2dissite 000-default.conf
sudo a2ensite lnxecorp.conf
```

```
sudo systemctl reload apache2
```

Verify the configuration by visiting the site in a browser on a machine connected to the same network (see Figure).

Figure 23: Verified Apache2 index (Bakke, 2025).

2.2.1.4 Set Up HTTPS for Apache with OpenSSH:

For a website to be trusted by most modern browsers, you have to secure the Configuring the HTTP connection with encryption using SSL/TLS is essential to protect sensitive data traffic from being intercepted by rouge actors. HTTPS also improves trust as certificates to generate a secure connection is issued by trusted authorities. Let's Encrypt is a brilliant free tool to generate trusted certificates, however it demand that you own the fully-qualified domain name (FQDN) and the website is verified with an established IP address.

It is therefore decided to assign a self-signed certificate to demonstrate SSL/TLS implementation in this project.

To do this, begin with creating a directory in the Apache2 directory to store SSL certificates:

```
sudo mkdir /etc/apache2/ssl
```

To create a strong and secure private key and a self-signed certificate, run the following command:

```
sudo openssl req -x509 -nodes -days 365 -newkey rsa:4096 -keyout
/etc/apache2/ssl/lnxecorp.key -out /etc/apache2/ssl/lnxecorp.crt
```

You will now be prompted to fill in information that will be incorporated into your certificate request.

After entering the requests, the generated certificate have to be added to the VirtualHost config file created when testing the Apache application:

```
sudo nano /etc/apache2/sites-available/lnxecorp.conf
```

Change the configurations of the file as follow:

```
<VirtualHost *:80>
    ServerName lnxecorp.com
    ServerAlias www.lnxecorp.com
    Redirect permanent / https://lnxecorp.com
</VirtualHost>

<VirtualHost *:443>
    ServerName lnxecorp.com
    ServerAlias www.lnxecorp.com
    ServerAdmin morpheus@lnxecorp.com
    DocumentRoot /var/www/lnxecorp

    SSLEngine on
    SSLCertificateFile /etc/apache2/ssl/lnxecorp.crt
    SSLCertificateKeyFile /etc/apache2/ssl/lnxecorp.key

    <Directory /var/www/lnxecorp>
        AllowOverride All
    </Directory>

    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

Now, enable SSL mode for Apache and restart:

```
sudo a2enmod ssl
sudo systemctl reload apache2
```

A secure connection to the website is now verified (see Figure) by running the Web Server's IP address in a browser again.

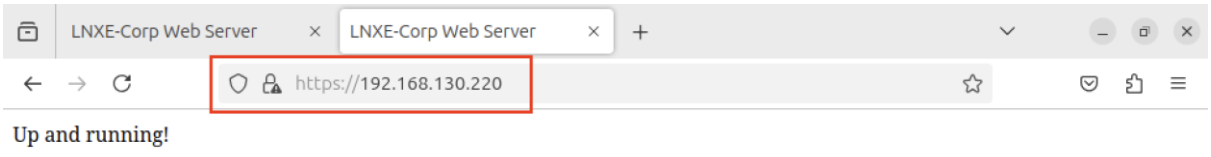

Figure 24: Verified Apache2 index with SSL/TLS (Bakke, 2025).

2.2.1.5 Create Groups and Users with Ownerships and Permissions:

One objective in this project is to demonstrate users connecting securely and working with files on the Web Server. This means the server has to be populated with a couple of user accounts and files.

Add two users from Development:

```
sudo adduser trinity
sudo adduser neo
```

Create a group for Development and add the two users crated in the previous step:

```
sudo groupadd devgroup
sudo usermod -aG devgroup trinity
sudo usermod -aG devgroup neo
```

`-aG` is used to append the users to the new group without losing their previous group memberships.

Add directories for the developers to the Web Server together with some test files for further demonstration of the project:

```
sudo mkdir -p /var/dev/lnxecorp/{devprojects,devshares,devcontracts}
sudo touch /var/dev/lnxecorp/devprojects/file{1..50}
sudo touch /var/dev/lnxecorp/devshares/file{51..90}
sudo touch /var/dev/lnxecorp/devcontracts/file{91..100}
```

Assign ownership of the directories to root and give access to the devgroup you created for the developers:

```
sudo chown -R root:devgroup /var/dev/lnxecorp/
```

Then, permissions has to be set to allow devgroup full access to the lnxecorp directories:

```
sudo chmod -R 770 /var/dev/lnxecorp/
```

Finally, ensure all files and directories created in these directories will inherit its parent permissions with `g+s`:

```
sudo chmod g+s /var/dev/lnxecorp
```

It should be noticed that access control lists could be applied for files and directories with the setfacl command and it would be strongly considered when the company expands and improved control is needed.

2.2.1.6 Enable and configure OpenSSH:

SSH is a cryptographic network protocol and has been the preferred way of securely connecting and login users on remote servers and machines. Using SSH key-based authentication, disable root login, and only allow group members to login instead of password-based authentication will mitigate the risk of brute-force attacks and credential theft.

Begin by enable the ssh.service with the command:

```
sudo systemctl enable ssh
```

Verify OpenSSH is enabled with the command:

```
systemctl list-unit-files | grep ssh
```

Verify the sshd configuration file by running:

```
sudo sshd -t -f /etc/ssh/sshd_config
```

In this case a message saying missing privilege separation directory: /run/sshd is displayed and needs to be created manually. OpenSSH has a security mechanism called privilege separation. Privilege Separation is a security feature in OpenSSH that is limiting the privileges of the SSH daemon process to minimize the attack surface and privilege escalation.

To create the missing directory and set the privilege and ownership to root, use the following commands and restart the service to assure it is running properly and finish off with the previous command to verify the sshd configuring file again:

```
sudo mkdir -p /run/sshd
sudo chmod 0755 /run/sshd
sudo chown root:root /run/sshd
sudo systemctl restart sshd
sudo sshd -t -f /etc/ssh/sshd_config
```

Before further configurations, it is good practice to backup and write protect the original configure file to not lose any default settings by running these commands:

```
sudo cp /etc/ssh/sshd_config /etc/ssh/sshd_config.original
sudo chmod a-w /etc/ssh/sshd_config.original
```

For better control and an additional security measure, SSH access should be restricted to only user accounts that should have it. Create a group called sshlogin and add users allowed to connect using SSH to the group:

```
sudo groupadd sshlogin
sudo usermod -aG sshlogin trinity
sudo usermod -aG sshlogin neo
sudo usermod -aG sshlogin morpheus
```

Now it is time to configure the sshd.config file by allowing only members of the sshlogin group, disable root login, and let the non-root users to login without using passwords.
Open the sshd_config file with nano using:

```
sudo nano /etc/ssh/sshd_config
```

Begin by uncomment and set `PermitRootLogin` to `no` for disabling root login through the SSH connection. If not present, add the line `AllowGroups sshlogin` for additional control of ssh login. To enable login without passwords, set `PasswordAuthentication` to `no` (see figure).

There are several security measures that could be implemented for SSH login in the configuration file, like changing the default SSH port number, however the measures done here is considered sufficient to demonstrate security judgement for the project.

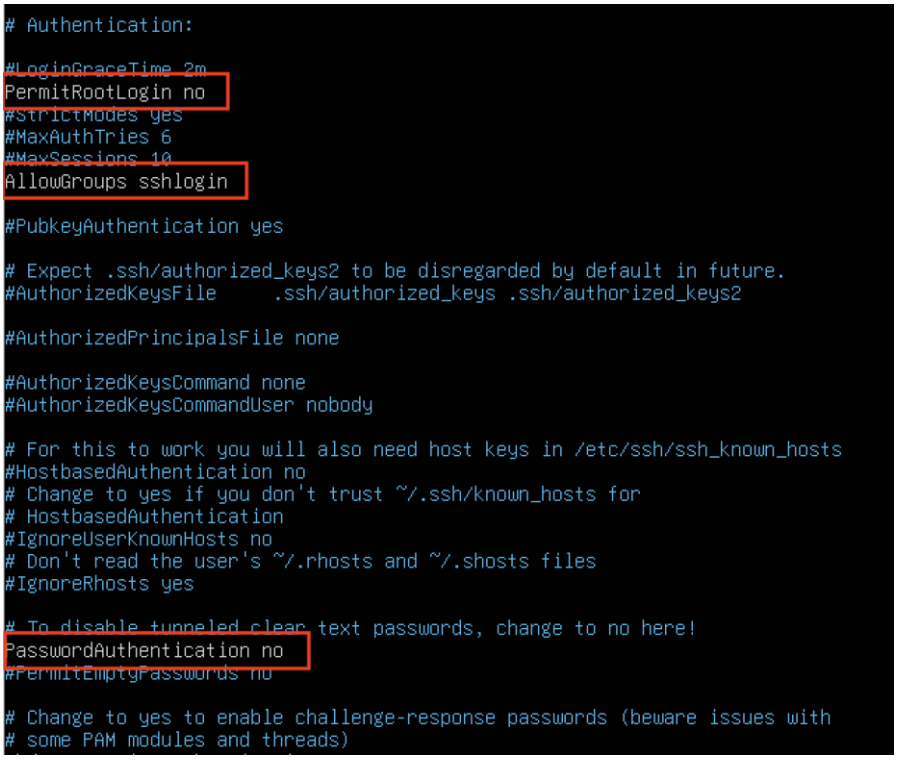

Figure 25: SSH configure file (Bakke, 2025).

Now we will login on the Admin PC and the Developer PC and generate SSH keys for each user:

```
ssh-keygen -t ed25519 -C "trinity@lnxecorp"
ssh-keygen -t ed25519 -C "neo@lnxecorp"
ssh-keygen -t ed25519 -C "morpheus@lnxecorp"
```

`-C` is added as a parameter to provide a comment to the key for easier removal of SSH accesses on the Web Server.

Leave the passphrase empty when prompted.

The two main options to generate a secure SSH key are either Ed25519 or RSA. RSA has been used for encryption of key pairs for a long time, and with 4096-bits signature algorithm it is very robust and absolutely a secure choice. Ed25519, on the other hand, is a newer EdDSA (Edwards-curve Digital Signature Algorithm) variation that is able to achieve the same level of security with much smaller key sizes than RSA and is generally recommended for its security and performance benefits (Colombier, 2024).

To copy the clients public keys to the server for this demonstration, it could either be done by the IP address or populating the host file `/etc/hosts` with `192.168.130.220 weberver.lnxecorp.com` on the client's machine. Both methods are presented here, where the host file is edited on the Admin PC:

```
ssh-copy-id trinity@192.168.130.220
ssh-copy-id neo@192.168.130.220
ssh-copy-id morpheus@webserver.lnxecorp.com
```

Restart the SSH service with `sudo systemctl restart sshd` on the Web Server if necessary and test the connection from the Admin PC. There is not prompt for a password as you can see in Figure :

```
ssh morpheus@webserver.lnxecorp.com
```

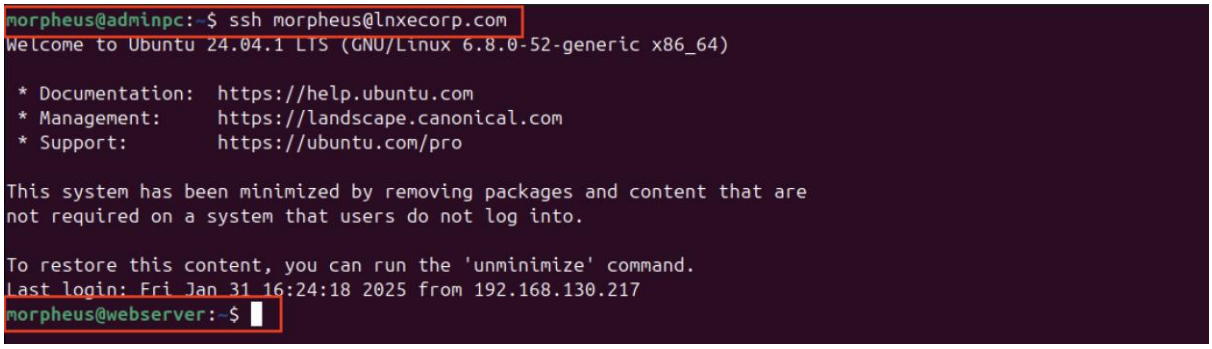

Figure 26: Verified successful SSH login without passphrase (Bakke, 2025).

Login with one of the developer accounts using the secure connection from the Developer PC and test permissions to the dev directories:

```
ssh neo@192.168.130.220
```

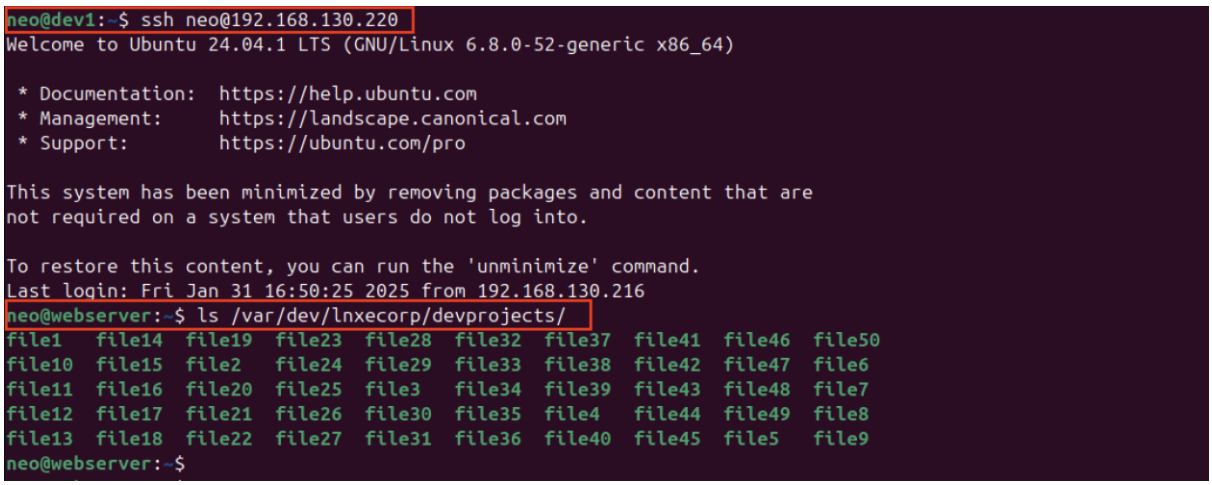

Figure 27: Verified successful SSH login and file permissions (Bakke, 2025).

Using SSH key authentication without passwords significantly enhances the security of the Web Server as passwords are much more vulnerable to password leaks and brute-force attacks than the strongly encrypted SSH key.

OpenSSH 8.2 has also support for the open standard U2F/FIDO for two-factor authentication hardware and could be used to provide an extra layer of security on top of the existing key-based authentication.

2.2.1.7 Install rsync

Before configuring the Backup Server, install rsync on the Web Server with the command:

```
sudo apt install rsync -y
```

Enable and start the rsync service once it is installed:

```
sudo systemctl enable rsync
sudo systemctl start rsync
```

2.2.2 Backup Server Setup

The configuration of the Backup Server focuses solitary on setting up an environment to demonstrate backups and restorations. Rsync has to be installed on both the source and destination system to get rsync to transfer files between them. You also need a method to access one system from the other. While, rsync has a default port of 873 for TCP traffic, it is not secure as it transmit data unencrypted (Ward, 2015). The rsync command has also the ability to use the SSH connection between hosts and this is what we will do to secure data-in-transit when copying from one host to the other.

2.2.2.1 Ubuntu Server Installation:

Follow the same steps as when installing the Web Server in 2.2.1.1 Ubuntu Server Installation with minimized server installation, LUKS encryption, Ubuntu Pro token, and OpenSSH. For this project, the username "lnxe-backup" is created as a user to perform backup demonstrations.

Update packages on the server and install nano for file configurations:

```
sudo apt update && sudo apt upgrade -y
sudo apt install nano -y
```

2.2.2.2 Install rsync

Install rsync on the Backup Server with `sudo apt install rsync -y`.

Once installed, start and enable the rsync service.

```
sudo systemctl start rsync
sudo systemctl enable rsync
```

2.2.2.3 Add User and Group for the Backup Server:

On the Web Server, create an user for the Backup Server. Add the user to the groups devgroup, and sslogin so the Backup Server can access the demonstrated data for backup and restoration:

```
sudo adduser lnxe-backup
sudo usermod -aG devgroup lnxe-backup
sudo usermod -aG sshlogin lnxe-backup
```

Create the same user on the Backup Server:

```
sudo adduser lnxe-backup
```

Change user on the Backup Server to lnxe-backup before you generate a SSH key:

```
su - lnxe-backup
```

2.2.2.4 Generate and Copy SSH key:

On the Backup Server, generate a SSH key par with a comment linking the key to its owner and copy it to the Web Server for data-in-transit encryption:

```
ssh-keygen -t ed25519 -C "lnxe-backup@lnxecorp.com"
```

```
ssh-copy-id lnxe-backup@192.168.130.220
```

2.2.2.5 Test Secure Connection and Permission:

To test access to the Web Server, be sure to log in as lnxe-backup on the Backup Server, login using SSH and test access to the lnxecorp.com directories (see Figure):

```
ssh lnxe-backup@192.168.130.220
ls -l /var/dev/lnxecorp/
```

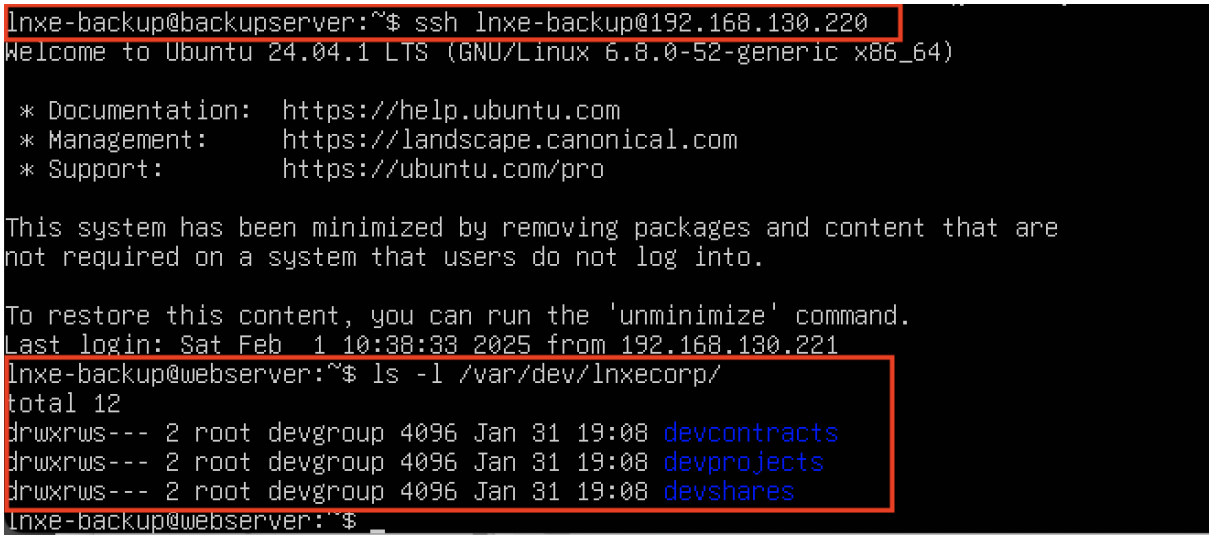

Figure 28: Verified SSH login from Backup to Web Server (Bakke, 2025).

Since the Backup Server is only backing up and restoring with rsync using SSH login, it is also decided to restrict the lnxe-backup user for shell access on the Web Server to limit the attack surface of the server:

```
sudo usermod -s /bin/rbash lnxe-backup
```

2.3 Data Backup and Restoration

Rsync and Cron Jobs will be performed from the Backup Server to pull backups from the Web Server. This let the Web Server focus on its primary role.

2.3.1 Create a directory for backups:

Login as the non-sudoer lnxe-backup on the Backup Server and create a directory for backups and remove permissions for others:

```
mkdir -p /home/lnxe-backup/backup/lnxecorp/
chmod -R 750 /home/lnxe-backup/backup/
```

2.3.2 Perform Backup from the Backup Server:

Perform a verbose "dry run" to test if the backup process works:

```
rsync -av --dry-run lnxe-backup@192.168.130.220:/var/dev/lnxecorp/
/backup/test/
```

The process works if the result looks like Figure below.

Figure 29: Verified rsync test (Bakke, 2025).

If the test is positive, a full backup of the website can be done:

```
rsync -aX lnxe-backup@192.168.130.220:/var/dev/lnxecorp/ /home/lnxe-backup/backup/lnxecorp/0001/
```

Using the `-a` and `-X` argument is meant to preserve all the attributes of the files so owner attributes or permissions will not be modified during the backup process.

In Figure , you can see that the Backup Server has performed a successful backup pull from the Web Server.

Figure 30: Verified backup process (Bakke, 2025).

2.3.3 Perform Restoration to the Web Server:

To demonstrate restoration of files on the Web Server, one of the directories in /var/dev/lnxecorp/ is deleted as shown in Figure .

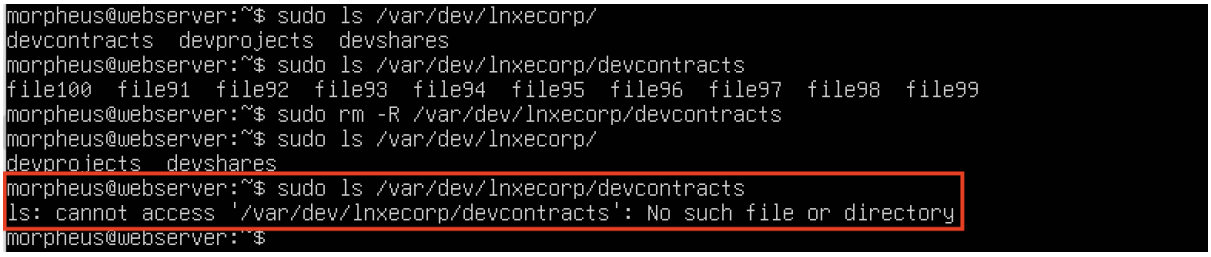

Figure 31: Deletion of a directory on the Web Server (Bakke, 2025).

From the Backup Server, run the following command to restore missing files on the Web Server:

```
rsync -aX /home/lnxe-backup/backup/lnxecorp/0001/ lnxe-
backup@192.168.130.220:/var/dev/lnxecorp/
```

Back on the Web Server, you can see the deleted files are restored (see Figure).

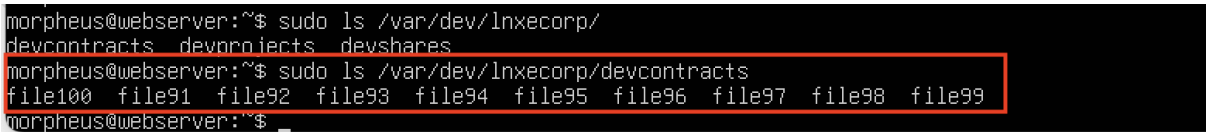

Figure 32: Verified restoration of deleted directory (Bakke, 2025).

2.3.4 Create a Cron Job for Periodically Backup:

Some automation should now be implemented to the process as pulling backups from the Web Server are demonstrated and working. The rate of backups vary depending on the company's need and individual data may have different demands. Based on the assumption that LNXE-Corp is a software firm, it is decided that a daily schedule should be sufficient to secure critical data for now.

The Cron daemon is a highly used system process to perform scheduled tasks in the background on Linux systems. In our setup, we have to install the Cron package logged in as the sudo user created during installation:

```
sudo apt install cron
```

A sudoer can edit and control other users' crontab jobs by adding `-u` and the username of the user to the command (Fox, 2021). So still logged in as the sudoer, open the crontab file to add a periodically backup task for the lnxe-backup user:

```
sudo -u lnxe-backup crontab -e
```

Specifying the recurrences of Cron Jobs are a bit complicated. The schedule is indicated with five values with spaces and is best explained with an illustration (see Figure).

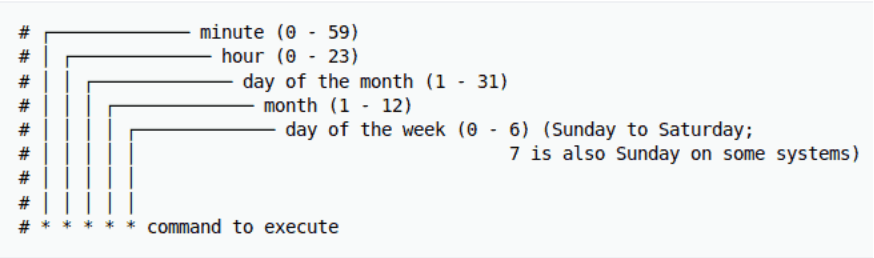

Figure 33: Explanation of crontab values (Linas and S, 2019).

Each of these values can be a specific integer or the wildcard *, which indicates every unit of the time period (Fox, 2021). Using 0 for minute and 0 for hour and leave rest of the values to indicate every unit will set the rsync task to be performed at midnight every day.
Enter the following text in the crontab to introduce a scheduled task to be run by Cron:

```
0 0 * * * rsync -avX lnxe-backup@192.168.130.220:/var/dev/lnxecorp/
/home/lnxe-backup/backup/lnxecorp/cron-job/
```

To verify the automatically backup done by Cron, change the time value to match a nearby instant and list files afterwards (see Figure):

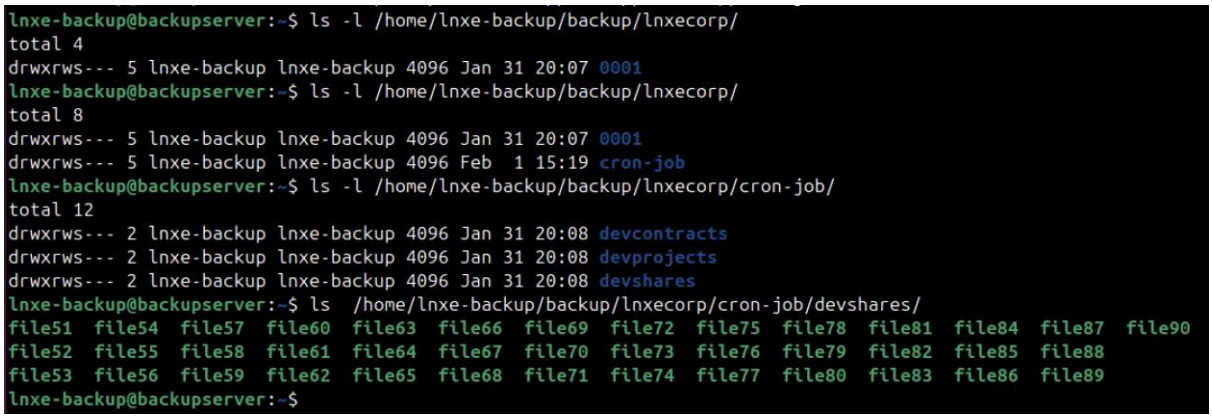

Figure 34: Verified automated Cron Job (Bakke, 2025).

2.4 OS Hardening Techniques

OS hardening is the process of making an operative system more secure by limiting its attack surface. This is done by deleting software packages that are not needed, tighten default settings, and configuring the system to only run programs and services that are required. System hardening should also be approached with implementing layered defenses so if a

weakness is present and found, it doesn’t lead to compromise an entire system. Using additional security measures as firewalls for both the OS and web applications, secure login without passwords, intrusion protection software, and encryption of data are all crucial to build a defense in depth.

OS hardening has so far, in this report, been demonstrated through the 2.2.1 Web Server Setup section by installing the minimized server version and initiated full disk encryption (FDE) using LUKS. There are other common open source options for encryption like eCryptfs and VeraCrypt with respectively file level encryption and cross-platform encryption, however, LUKS is chosen for the built-in convenience during the OS installation and since all machines are running on the same platform, together with an appeal to use block encryption for the whole disk. Using FDE protects all data-at-rest on the servers and ensures that even if the physical server is stolen, the data cannot be accessed without the encryption key.
Further, by upgrading all system packages, the OS is hardened with the latest software and security patches, and with securing access to the Web Server with SSH and configure a defined group allowed to login is further securing the data-in-transit.

2.4.1 Disable Unused Services and Ports:

In this minimized installation, there shouldn't be any unused services, however, to verify and check the listening ports of each service and service name, use the ss command below:

```
sudo ss -tulnp | grep LISTEN
```

Stop and disable any undesired services listening on ports by their service name:

```
sudo systemctl stop <service name>
sudo systemctl disable <service name>
```

2.4.2 Install and Configure a Host-based Firewall:

Install the Uncomplicated Firewall (ufw) service to manipulate the Linux kernel's Netfilter subsystem and manage the network traffic going in and out of the server with the command:

```
sudo apt install ufw -y
```

This installation will also install iptables, which is the main utility interacting with the kernel, however, ufw provides a more user-friendly approach to manage simple rules (Ubuntu, 2024). Activate ufw with the command:

```
sudo systemctl start ufw
```

Use ufw to ensure you have the basic rules to the firewall by denying incoming traffic and allow outgoing traffic:

```
sudo ufw default deny incoming
sudo ufw default allow outgoing
```

List the network reliable applications that have registered profiles in ufw and allow OpenSSH and Apache Secure:

```
sudo ufw app list
sudo ufw allow OpenSSH
sudo ufw allow 'Apache Secure'
```

The Apache application has different rules available added to ufw and by using the Apache Secure command, it will make sure the firewall only allow secure HTTP using port 443. There are also rules for IPv6 since IPv6 enabled by default in resent Ubuntu distributions and it is decided to keep this enabled so equal rules will be applied for IPv4 and IPv6 if the latter protocol will be used in the future.

Enable logging and limit the rate of log in attempts on port 22 for added security measures and preventing SSH brute-force attacks:

```
sudo ufw logging medium
sudo ufw limit 22/tcp
```

Ufw is disabled by default after installation and is enabled with the command:

```
sudo ufw enable
```

You can verify the firewall rules by checking the ufw status (see Figure):

```
sudo ufw status
```

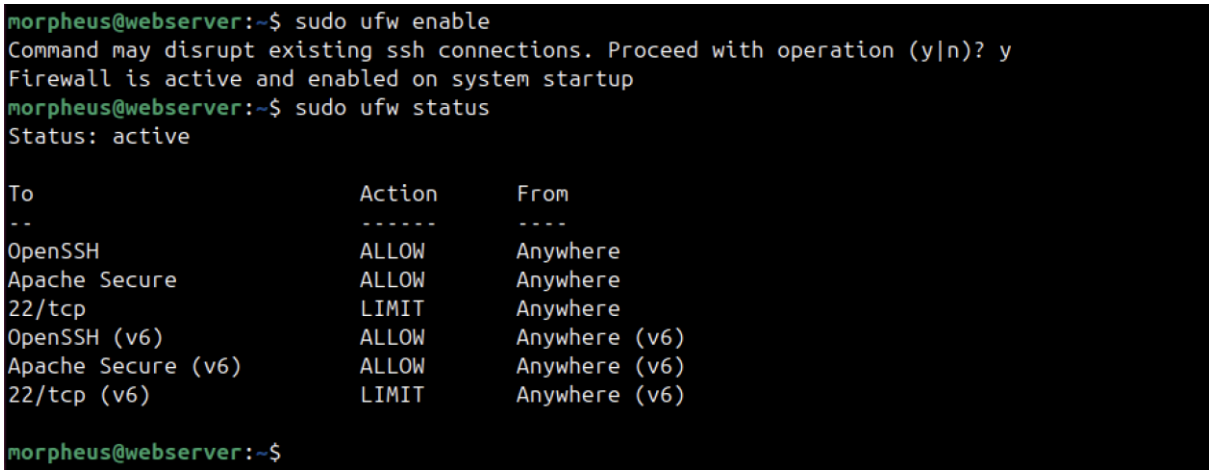

Figure 35: Verified firewall rules (Bakke, 2025).

These configurations was performed with a SSH connection from the Admin PC and as you can see in Figure , the OpenSSH rule worked and did not disrupt the connection.

2.4.3 Configure an Intrusion Detection System:

Fail2Ban is a widely acknowledged intrusion prevention system for Linux and protects your virtual server hosts against many security threats, such as DoS, DDoS, dictionary, and brute-force attacks. It works by scanning the system logs for any malicious activity and for any entries matching identified patterns. File2Ban integrates perfectly with iptables by updating the systems firewall with configured rules and reject new connections if it detects a certain amount of failed login attempts by blocking the source IP address for a certain time or indefinitely.

Fail2Ban is installed withe the following command:

```
sudo apt install fail2ban -y
```

Once the installation is complete, verify that Fail2Ban is installed correctly and enabled by checking its status:

```
sudo systemctl status fail2ban
```

When the Fail2Ban installation is verified, copy and rename the newly installed jail.conf file to jail.local:

```
sudo cp /etc/fail2ban/jail.conf /etc/fail2ban/jail.local
```

Fail2Ban works by reading its original files first and then overrides the settings with any .local files. Changes done in the original file will probably be overwritten via updates in the future so it is best practice to leave the original file untouched.

File2Ban uses several configuration files to customize and set up separate security rules, actions, and filters for different services on the server.

In this demonstration, File2Ban will be used to deploy additional login security and prevent malicious SSH logins. To do so, open the jail.local file and navigate to the JAILS section and configure the following lines (see Figure):

```
sudo nano /etc/fail2ban/jail.local
```

```
[sshd]
enabled = true
port = ssh
filter = sshd
maxretry = 3
bantime = 3600
```

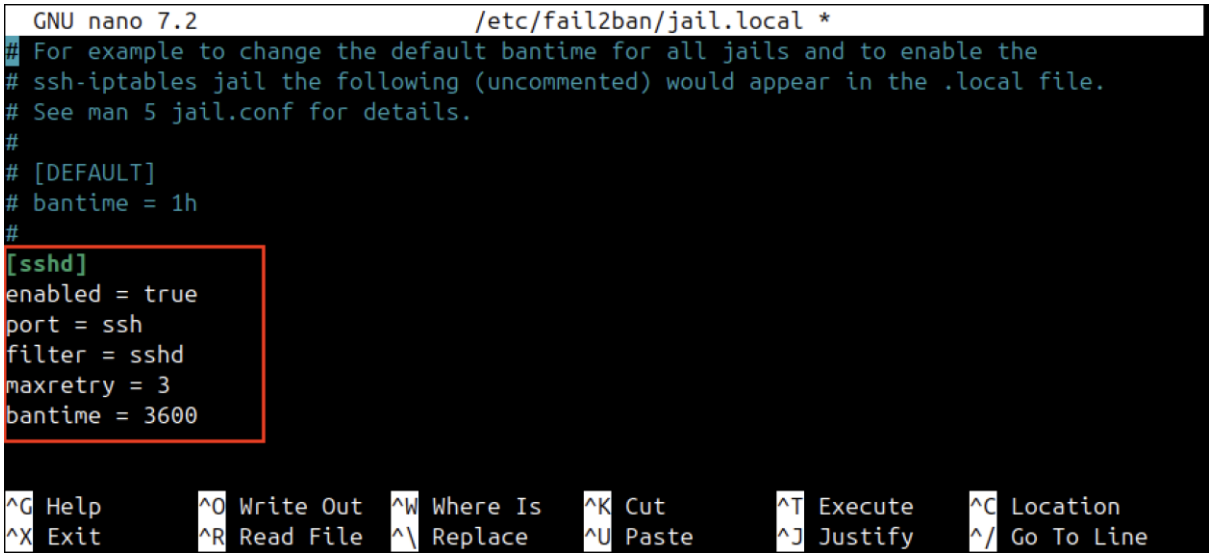

Figure 36 : Fail2Ban jail configuration (Bakke, 2025).

This configuration enables SSH protection through File2Ban, sets the maximum number of failed attempts to 3, and bans the offending IP for 1 hour.

Restart fail2ban with:

```
sudo systemctl restart fail2ban
```

2.4.4 Verify AppArmor:

AppArmor is a pre-installed security module that comes with Ubuntu distributions. AppArmor provides the OS with mandatory access control (MAC) and lets you set specific restrictions for any app on your machine. AppArmor makes sure that applications are tightly controlled and only allows minimal privileges as part of the OS hardening process.

Verify if AppArmor is active and enabled:

```
sudo systemctl status apparmor
```

2.4.5 Performed Regular Updates and Apply Security Patches:

It is crucial to stay up to date with updating software and system processes running on your devices. Updates and security patches should be implemented in the company's security policy and be performed regularly for the infrastructure to stay safe and healthy.

2.4.6 Automated Security:

As part of the Ubuntu Pro Enterprise subscription, Canonical, who distribute Ubuntu, has provided the automated cyber security tool Ubuntu Security Guide (USG) for system hardening, remediation and auditing.

The Center for Internet Security (CIS) publishes hardening benchmarks for many common software applications and operating systems, such as Ubuntu. USG performs automation and implement the hundreds of rules within the CIS benchmarks to your system as part of Ubuntu Pro and ensure a comprehensive level of security.

While not executed in this report, with a Ubuntu Pro subscription and USG installed, hardening your Ubuntu system to the CIS standards is as simple as running the command:

```
usg fix cis_level1_server
```

3 Conclusion

This project successfully establishes a secure and functional network infrastructure for the LNXE-Corp company and addressing important security measures while maintaining an

operational and competence network. The network was designed using a Router-on-a-Stick topology with VLAN segmentation for logical isolation between the various operative departments. Security was prioritized through the configuration of AAA authentication, access control lists, port security, and encryption protocols such as SSH and HTTPS.

The server configurations were done with security as the first priority with minimal attack surface. This included OS installing without unnecessary services, user access controls via groups and ownership, and encrypting both data-in-rest and data-in-transit. The Web Server was configured with Apache2 and secured with self-signed SSL certificates and critical data was securely backed up and restored by the Backup Server using rsync over SSH connections. Backups were also introduced with automation tasks through Cron Jobs to preserve high availability and reduced risk of data loss. OS hardening was further improved using a software-based firewall with few and strict rules, and an intrusion prevention system.

The project encountered mostly challenges with defining permissions and ownership for the different groups and applications in Linux. This involving access to files and directories where access was dismissed to some users because a parent folder has been granted strict access control and this followed through the hierarchy. Solving these obstacles did highlight the build-in security and effectiveness of the Linux computer system. The limitations of Packet Tracer did also provide a constraint in demonstrating certain advanced security configurations, such as DNS security and theoretical validation had to be done instead of practical demonstration.

In conclusion, the project was able to effectively demonstrate a layered security approach aligned with the CIA triad to ensure confidentiality through data encryption and restricted access, the integrity was controlled through authentication and permissions, and availability was secured with scheduled backups. While additional security measures, like multi-factor authentication and advanced firewall configurations on the network boarder could further strengthen the system, the security measures implemented in this project provides a strong foundation for a secure and scalable network infrastructure.

4 References

Cisco (2018). *Security Configuration Guide: Cisco Umbrella Integration On Cisco 4000 Series ISRs*. [online] Cisco Product Support. Available at: https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/sec_data_umbrbran/configuration/xe-16/sec-data-umbrella-branch-xe-16-book/sec-data-umbrella-bran.html [Accessed 30 Jan. 2025].

Cisco (2020). *Cisco Learning Network*. [online] Cisco.com. Available at: https://learningnetwork.cisco.com/s/article/configure-cisco-router-as-dns-server [Accessed 14 Jan. 2025].

Cisco (2021). *Cisco IOS Security Command Reference: Commands A to C - aaa accounting through aaa local authentication attempts max-fail [Support]*. [online] Cisco Support. Available at: https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/security/a1/sec-a1-cr-book/sec-cr-a1.html#wp2933959514 [Accessed 28 Jan. 2025].

Colombier, C. (2024). *How to generate a secure and robust SSH key in 2024*. [online] DEV Community. Available at: https://dev.to/ccoveille/how-to-generate-a-secure-and-robust-ssh-key-in-2024-3f4f [Accessed 20 Jan. 2025].

Fox, R. (2021). *Linux with Operating System Concepts*. 2nd ed. Boca Raton, FL: CRC Press.

Gordon, W. and Cohen, J. (2023). *More Secure Wi-Fi: What Is WPA3, and How to Set it Up on Your Router*. [online] PCMAG. Available at: https://www.pcmag.com/explainers/what-is-wpa3-secure-wifi-how-to-set-it-up-on-your-router [Accessed 23 Jan. 2025].

Linas, L. and S, A. (2019). *Cron Job: a Comprehensive Guide for Beginners 2020*. [online] Hostinger Tutorials. Available at: https://www.hostinger.com/tutorials/cron-job [Accessed 25 Jan. 2025].

Odom, W. (2016). *CCENT/CCNA ICND1 100-105 Official Cert Guide, Academic Edition*. Indianapolis, IN: Cisco Press.

Santos, O. and Stuppi, J. (2015). *CCNA Security 210-260 Official Cert Guide*. Indianapolis, IN: Cisco Press.

Ward, B. (2015). *How Linux Works : What Every Superuser Should Know*. 2nd ed. San Francisco, CA: No Starch Press.