Network Topology Report: LAN Configuration

1. Overview

A **Local Area Network (LAN)** is a network that connects computers and devices within a limited geographic area, such as a home, school, office building, or campus. LANs are typically used to share resources such as printers, files, and internet access, and are known for their high speed and low latency.

---

2. Selected Network Topology: Star Topology in a LAN

A **star topology** is one of the most commonly used configurations in LAN setups. In this configuration, all devices (nodes) are connected to a central device, usually a **switch** or **hub**.

---

3. Network Topology Diagram

lua
CopyEdit

```
                   +----------+
                    | Printer |
                    +----+----+
                         |
                         |
+---------+        +-----+-----+         +---------+
| Laptop |---------|           |---------| Server  |
+---------+        |   Switch  |         +---------+
+---------+        |           |         +---------+
|  PC 1  |---------|           |---------|  PC 2   |
+---------+        +-----------+         +---------+
                         |
                         |
                   +-----+-----+
                   | Router    |
                   +-----------+
                   |
               [Internet]
```

4. Secure Communication Support

LANs in a star topology support secure communication in several ways:

- **Centralized Management**: The use of a switch or router allows administrators to control and monitor traffic from a central point. This makes it easier to detect suspicious activity and apply security policies.
- **Segmentation**: VLANs (Virtual LANs) can be configured on managed switches to segment network traffic, reducing broadcast domains and isolating sensitive data or departments.
- **Access Control**: Network devices can be configured with MAC address filtering, port security, or 802.1X authentication to prevent unauthorized access.
- **Firewall & Intrusion Detection**: Routers or dedicated appliances at the edge of the LAN can be configured with firewalls or IDS/IPS systems to inspect and control traffic to and from the internet.
- **Encryption**: Internal LAN traffic can be encrypted using protocols like IPSec or by implementing secure tunnels (VPNs) for sensitive communications.

5. Network Management Benefits

- **Easy Troubleshooting**: If a device or cable fails, it can be easily identified and fixed without affecting the entire network.
- **Scalability**: Devices can be added or removed without disrupting the rest of the network.
- **Monitoring Tools**: SNMP (Simple Network Management Protocol) and centralized logging systems can be used to gather network performance and security data.
- **Patch Management**: Centralized servers can manage software updates across all connected devices.

6. Conclusion

A **LAN using star topology** is an effective and secure configuration for small to medium-sized environments. It supports robust security features and simplifies network management, making it an ideal choice for modern organizations seeking control, performance, and security within their local networks.

✅ Network Design Report

1. Overview

This report outlines a secure network design that includes:

- The **OSI and TCP/IP models** mapped to a **single device** (PC).
- **Proper subnetting** for one subnet.
- A **secure architecture** utilizing protocols such as **HTTPS, SSH, IPSec**, and **802.1X**.

---

2. Network Diagram and Architecture

pgsql
CopyEdit

```
         [Internet]
               |
         +-----------+
         |  Firewall |
         +-----+-----+
               |
          +----+----+
          |  Router  |     <== Gateway IP: 192.168.10.1
          +----+----+
               |
       +-------+-------+
       |   Managed     |
       |   Switch      |
       +---+---+---+---+
           |   |   |
    +-------+   |   +-------+
    |          |          |
+---+---+  +---+---+  +---+---+
|  PC 1  |  | Server|  |Printer|
+-------+  +-------+  +-------+
```

---

3. Subnetting Details

Network Address: `192.168.10.0/24`
Subnet Mask: `255.255.255.0`
Total IPs: 256
Usable Hosts: 254
Default Gateway: `192.168.10.1`

Device	IP Address	Subnet Mask
PC 1	192.168.10.10	255.255.255.0
Server	192.168.10.20	255.255.255.0
Printer	192.168.10.30	255.255.255.0

4. OSI & TCP/IP Model for PC 1

OSI Layer	TCP/IP Layer	Function in PC 1
7. Application	Application	Web browser using HTTPS, Email via SMTP/IMAP
6. Presentation	Application	SSL/TLS encryption of HTTP data
5. Session	Application	Establishes and manages sessions (SSH, RDP)
4. Transport	Transport	TCP for reliable data (HTTPS), UDP for DNS
3. Network	Internet	IP addressing and routing (IPv4)
2. Data Link	Network Access	Ethernet, MAC addressing, 802.1X authentication
1. Physical	Network Access	Ethernet cables, Wi-Fi interface

5. Secure Network Architecture & Protocols

Security Protocol	Purpose
HTTPS	Encrypts web traffic between PC and websites

SSH	Secure remote login to servers
IPSec	Encrypts network layer data for VPNs or internal comm
802.1X	Authenticates devices before network access
Firewall Rules	Blocks unauthorized incoming/outgoing traffic
Antivirus/EDR	Host-level protection against malware & threats
VLANs	Separate devices by department or function

---

6. Summary

This secure network setup ensures:

- **Proper segmentation** via subnetting and VLANs.
- **End-to-end encryption** using HTTPS, SSH, and IPSec.
- **Access control** using 802.1X.
- **Visibility and control** via firewall and managed switch.

It provides a **resilient, secure, and scalable** architecture for home or small office environments.

✅ Network Security Fundamentals Report

1. Objective

This report outlines the implementation of essential network security components:

- A **firewall rule** to control network traffic.
- An **Intrusion Detection System (IDS)** configuration to monitor threats.
- An **Intrusion Prevention System (IPS)** setup to block attacks.
- One **example of a detected event** for each system.

---

2. Firewall Rule Implementation

Tool Used: UFW (Uncomplicated Firewall) on Ubuntu Server

Rule: Block all incoming traffic except SSH (port 22), HTTP (port 80), and HTTPS (port 443).

bash
CopyEdit

```
sudo ufw default deny incoming
sudo ufw default allow outgoing
sudo ufw allow 22/tcp
sudo ufw allow 80/tcp
sudo ufw allow 443/tcp
sudo ufw enable
```

✅ **Purpose:**

This firewall rule reduces the attack surface by allowing only essential services and blocking unnecessary ports.

---

3. IDS Configuration

Tool Used: Snort (IDS Mode)

Configuration Snippet (Local Rule File - `local.rules`):

snort
CopyEdit

```
alert tcp any any -> 192.168.10.10 22 (msg:"Possible SSH brute force";
flags:S; threshold:type threshold, track by_src, count 5, seconds 60;
sid:1000001;)
```

✅ **Explanation:**

This rule raises an alert if 5 or more SSH connection attempts are made to the PC within 60 seconds — indicating a potential brute force attempt.

🛑 **Example Detected Event:**

css
CopyEdit

```
[**] [1:1000001:0] Possible SSH brute force [**]
[Classification: Attempted Administrator Privilege Gain]
[Priority: 1]
04/20-14:31:44.847584 192.168.1.25 -> 192.168.10.10
```

```
TCP TTL:64 TOS:0x0 ID:48997 IpLen:20 DgmLen:48
```

4. IPS Configuration

Tool Used: Snort (Inline IPS Mode with NFQUEUE)

Rule Example: Block Ping (ICMP) Flood
snort
CopyEdit

```
drop icmp any any -> any any (msg:"ICMP Flood detected"; dsize: >100;
detection_filter: track by_src, count 10, seconds 1; sid:1000002;)
```

✅ Explanation:

This IPS rule blocks ICMP echo requests (pings) larger than 100 bytes if 10 are detected within 1 second — a sign of a DoS attack.

🛑 Example Blocked Event:

less
CopyEdit

```
[Drop] [1:1000002:0] ICMP Flood detected [**]
04/21-10:03:21.122334 192.168.1.50 -> 192.168.10.10
ICMP TTL:64 Type:8 Code:0 ID:3333 Seq:1100 Len:120
```

5. Summary of Protection

Security Tool	Protection Type	Action Taken	Detected Example
Firewall	Access Control	Block all but essential ports	Unauthorized FTP blocked
IDS (Snort)	Threat Detection	Alert	SSH brute force detected
IPS (Snort)	Threat Prevention	Drop traffic	ICMP flood blocked

✅ Conclusion

This implementation demonstrates core security defenses by:

- **Limiting entry points** via firewall rules,
- **Detecting suspicious activity** using IDS,
- **Actively blocking threats** using IPS.

Together, these provide a layered approach to **defense-in-depth** and help maintain a secure network environment.

✅ Access Control Measures Implementation Report

1. Objective

This report demonstrates the implementation of access control within a secure network environment, including:

- One **Access Control List (ACL)** configuration.
- One **access control model** (DAC).
- One **user access level** definition.

---

2. Access Control List (ACL) Configuration

Scenario:

Restrict network access so that only the Admin PC (`192.168.10.10`) can connect to a web server on port 80 (`192.168.10.20`), and all other IPs are denied.

Device: Cisco Router or Layer 3 Switch

ACL Configuration Example:
cisco
CopyEdit

```
access-list 100 permit tcp host 192.168.10.10 host 192.168.10.20 eq 80
access-list 100 deny ip any any
interface GigabitEthernet0/1
 ip access-group 100 in
```

✅ **Purpose:**

This ACL restricts access to the internal web server, allowing only the Admin PC to communicate over HTTP and blocking all others. This minimizes unauthorized access attempts.

3. Access Control Model: Discretionary Access Control (DAC)

Definition:

DAC is an access control model where the **resource owner** (e.g., file or directory owner) determines who can access the resource and what permissions they have.

Implementation Example:

On a **Linux server**, the file `/srv/data/report.txt` is owned by `user1`. Using DAC, `user1` can control access using permissions:

bash
CopyEdit

```
# Only user1 can read and write; others denied
chmod 600 /srv/data/report.txt
chown user1:user1 /srv/data/report.txt
```

✅ **Purpose:**

This ensures that only the file owner can access or modify the file, and no other user on the system can read or write to it without explicit permission.

---

4. User Access Level Definition

Role-Based Example:

Username	Role	Access Level	Permissions
`admin1`	Administrator	Full access (root/UID 0)	Can configure firewall, manage users, install software
`user2`	Standard User	Restricted access	Can read/edit personal files, no system config rights
`guest`	Guest	Minimal access	Can browse network, no write or system permissions

✅ **Purpose:**

Defining access levels ensures users only access resources and perform actions necessary for their role — supporting the **principle of least privilege**.

---

5. Summary of Implementation

Component	Method Used	Result
ACL	Cisco ACL on router	Restricts web access to a single IP
Access Control Model	DAC (Discretionary Access Control)	User manages permissions on owned resources
User Access Level	Role-Based Permissions	Users assigned only necessary system rights

---

✅ Conclusion

This implementation shows how **ACLs**, **DAC**, and **user roles** combine to enforce a **layered and effective access control strategy**, ensuring data security and reducing insider and external threats.

✅ Secure Wireless Network Implementation Report

1. Objective

This report documents the security measures implemented on a wireless network, focusing on:

- **WPA2/WPA3 encryption configuration for one wireless network.**
- **The use of a Wireless Intrusion Prevention System (WIPS) to detect and block unauthorized access.**

---

2. Wireless Network Configuration (WPA3)

Network Name (SSID): `SecureNet_Office`

Access Point: Ubiquiti UniFi U6-Lite

(*Other routers such as Cisco, TP-Link, or ASUS can also be configured similarly.*)

Security Settings:

Setting	Value
SSID	SecureNet_Office
Security Type	WPA3-Personal
Encryption	AES-CCMP
Passphrase	`Str0ngPass!2025`
SSID Broadcast	Enabled (optional)
MAC Filtering	Enabled (whitelist)
Guest Network	Isolated VLAN

✅ Purpose:

WPA3 uses Simultaneous Authentication of Equals (SAE) to protect against brute-force attacks, offering stronger encryption and resilience over WPA2. AES-CCMP ensures robust data encryption.

---

3. Wireless Intrusion Prevention System (WIPS) Setup

Tool Used: Cisco Meraki AirMarshal / Ubiquiti UniFi IDS/WIPS

WIPS Functions Enabled:

- **Rogue AP Detection**: Detects access points that aren't part of the organization.
- Evil Twin Prevention: Alerts when a cloned SSID tries to mimic the real network.
- Deauthentication Flood Protection: Blocks denial-of-service attempts targeting client devices.
- MAC Spoofing Detection: Identifies devices attempting to change MACs to bypass filtering.

✅ Example Detection Event:

Event Type	Description
Rogue AP Found	"Unapproved AP with SSID 'Free WiFi'" detected at MAC `00:14:22:01:23:45`
Action Taken	Blocked client association and sent alert
Timestamp	2025-06-20 14:18:00

---

4. Summary of Implementation

Component	Tool/Configuration	Purpose
WPA3 Encryption	Enabled with AES-CCMP	Secures wireless communication
MAC Filtering	Allow-listed device access	Restricts access to known devices
WIPS	Ubiquiti / Cisco Meraki	Detects and prevents wireless threats
Guest Isolation	VLAN-separated SSID	Prevents guests from accessing internal LAN

✅ Conclusion

The combination of WPA3 encryption and an actively monitored WIPS creates a secure wireless environment by:

- Encrypting all communications to prevent eavesdropping,
- Allowing only approved devices to connect,
- Detecting and responding to rogue threats in real-time.

This wireless security implementation upholds confidentiality, integrity, and availability across the organization's Wi-Fi network.

✅ Network Security Tools Report

1. Objective

This report demonstrates the usage of three core network security tools:

- A Wireshark packet capture with analysis
- A network vulnerability scanner report (using Nessus or OpenVAS)
- A penetration testing tool output (using Metasploit)

2. Tool #1: Wireshark Packet Capture & Analysis

Scenario: Capturing HTTP traffic from a workstation browsing a non-HTTPS site.

Capture Setup:

- **Interface:** Ethernet0
- Filter Used: `http`

Key Findings:

- Captured several HTTP GET requests
- Detected unencrypted credentials during form login

✅ Sample Analysis:

Field	Details
Source IP	192.168.10.10
Destination IP	93.184.216.34 (example.com)
Protocol	HTTP
Info	GET /login.php
Payload	`username=admin&password=12345` (visible)

✅ Conclusion:

This shows why using HTTPS is essential — Wireshark easily captured plaintext login credentials.

---

3. Tool #2: Network Vulnerability Scan (OpenVAS)

Tool Used: Greenbone/OpenVAS

Target: Internal Web Server `192.168.10.20`

✅ Scan Summary:

Metric	Value
Total vulnerabilities	9
High severity	2
Medium severity	4
Low severity	3

✅ Sample High-Risk Vulnerabilities Found:

1. CVE-2021-41773 – Apache Path Traversal
   Risk: Allows remote attackers to access sensitive files.
   Solution: Patch to latest Apache version.
2. Outdated PHP Version Detected
   Risk: Contains known vulnerabilities.
   Solution: Update to PHP 8.x.

4. Tool #3: Penetration Test (Metasploit Framework)

Objective: Exploit unpatched vulnerability on the same web server (`192.168.10.20`)

Exploit Used: Apache 2.4.49 Path Traversal (CVE-2021-41773)

✅ Steps:

bash
CopyEdit

```
msfconsole
use exploit/multi/http/apache_path_traversal
set RHOSTS 192.168.10.20
set TARGETURI /
exploit
```

✅ Output:

css
CopyEdit

```
[*] Exploit completed, but no session was created.
[*] Retrieved /etc/passwd from target
```

✅ Conclusion:

The Metasploit exploit succeeded in reading sensitive files from the server, confirming the vulnerability reported by the scanner.

5. Summary of Tools Used

Tool	Purpose	Outcome
Wireshark	Network traffic capture and analysis	Detected plaintext credentials over HTTP
OpenVAS	Vulnerability scanning	Identified unpatched Apache and PHP vulnerabilities
Metasploit	Penetration testing & exploitation	Verified exploitability of Apache path traversal flaw

✅ Conclusion

Using Wireshark, OpenVAS, and Metasploit together provides a complete view of:

- Detection (via packet inspection)
- Assessment (via vulnerability scans)
- Exploitation (via real-world attack simulation)

This layered approach is critical to validating and strengthening network security posture.

✅ Network Security Monitoring & Incident Response Report

1. Objective

This report details the process of monitoring network security events, identifying a security incident, and documenting the incident response steps taken, with support from logs and screenshots.

---

2. Monitoring Setup

Tools Used:

- Security Information and Event Management (SIEM): ***Wazuh***
- Intrusion Detection System (IDS): *Snort*
- Log Sources: Syslog from firewall, server logs, and endpoint alerts

Monitored Assets:

- Internal Web Server (`192.168.10.20`)
- Endpoint Workstation (`192.168.10.10`)

---

3. Security Incident Identified

Type: Brute-force login attempt to SSH service

Detected By: Snort IDS + Wazuh SIEM

Timestamp: `2025-06-20 13:24:10`

Source IP: `192.168.1.25`

Target IP: `192.168.10.10` (PC 1)

---

4. Logs and Screenshot Evidence

Wazuh Alert Log Snippet:

json
CopyEdit

```json
{
  "rule": {
    "level": 10,
    "description": "Multiple SSH authentication failures"
  },
  "srcip": "192.168.1.25",
  "dstip": "192.168.10.10",
  "timestamp": "2025-06-20T13:24:10Z",
  "event": {
    "ssh_failures": 8,
    "interval_seconds": 60
  }
}
```

Snort Alert Log:

css
CopyEdit

```
[**] [1:1000001:0] Possible SSH brute force [**]
[Priority: 1] 06/20-13:24:10.482334
TCP 192.168.1.25:45781 -> 192.168.10.10:22
```

> 🖼️ (Insert screenshot here): Screenshot of Wazuh dashboard showing SSH brute-force alerts

---

5. Incident Response Steps

Step	Action Taken	Tool Used
1	Alert reviewed and verified	Wazuh Dashboard
2	Malicious IP `192.168.1.25` blocked	UFW Firewall
3	SSH service hardened (Fail2Ban installed)	Linux Server
4	System logs collected and archived	Syslog
5	Report generated and shared with admin team	Internal Report
6	Scheduled vulnerability assessment initiated	OpenVAS

Firewall Block Command:

bash

CopyEdit

```
sudo ufw deny from 192.168.1.25 to any port 22
```

Fail2Ban SSH Filter:

bash

CopyEdit

```
[sshd]
enabled = true
port    = ssh
filter  = sshd
logpath = /var/log/auth.log
maxretry = 5
```

---

6. Outcome

- Attack mitigated before access was gained
- Source IP blacklisted
- SSH brute-force protection now active via Fail2Ban
- No data loss or system compromise detected

✅ Conclusion

This report demonstrates a successful real-time security monitoring and incident response workflow, including:

- Detection using IDS and SIEM tools
- Verification and analysis of logs
- Swift mitigation using firewall and service hardening
- Documentation for compliance and follow-up

The organization now has stronger defenses against repeated brute-force attacks on remote services.