Network Security Implementation Report

This report details the implementation of critical **network security fundamentals** and **access control measures** within the organization's infrastructure. It covers the configuration of:

1. **Firewall rule implementation**
2. **Intrusion Detection System (IDS) configuration**
3. **Intrusion Prevention System (IPS) configuration**
4. **Access control measures** including:
    - **Access Control List (ACL) configuration**
    - **Access control model** (e.g., DAC, MAC)
    - **User access levels**

1. Network Security Fundamentals Implementation

1.1 Firewall Rule Implementation

Objective:
The goal was to implement a firewall rule to protect the internal network from unauthorized access while allowing legitimate traffic for business needs.

Firewall Tool Used: Cisco ASA (Adaptive Security Appliance)

- **Firewall Rule Configuration:**
  The rule was designed to:
    - **Block** all inbound traffic from external sources trying to access vulnerable ports (e.g., Telnet port 23, RDP port 3389).
    - **Allow** traffic for HTTP (port 80) and HTTPS (port 443) to facilitate external users accessing the corporate website.

Firewall Rule Example:
arduino
Copy code

```
access-list ACL_BLOCK extended deny tcp any host 10.1.1.100 eq 23
access-list ACL_BLOCK extended deny tcp any host 10.1.1.100 eq 3389
access-list ACL_BLOCK extended permit tcp any host 10.1.1.100 eq 80
access-list ACL_BLOCK extended permit tcp any host 10.1.1.100 eq 443
```

-
    - **Explanation:**

- The first two rules **deny** inbound TCP traffic from any external source (any IP) to the internal server (IP `10.1.1.100`) on ports **23** (Telnet) and **3389** (RDP).
- The last two rules **permit** inbound HTTP and HTTPS traffic (ports 80 and 443) to the same server, allowing external users to access the website.

Detected Event Example:
Firewall Log:
css
Copy code

```
[Timestamp] DENY [Source IP: 203.0.113.45] [Destination IP: 
10.1.1.100] [Port: 23] [Protocol: TCP] - Telnet attempt blocked
```

-
  - **Event Details:**
    - A Telnet attempt was detected and blocked from an external IP address (203.0.113.45) targeting the internal server (10.1.1.100) on port 23. This event was captured in the firewall logs, confirming that the rule was working as intended.

1.2 Intrusion Detection System (IDS) Configuration

Objective:
The IDS was configured to detect potential malicious activity on the network, such as port scanning or unauthorized access attempts.

IDS Tool Used: Snort (Open-source IDS)

IDS Rule Configuration:
A rule was created to detect **port scanning** behavior, which occurs when a source IP address attempts to scan multiple ports on a target machine in a short period.
IDS Rule Example:
css
Copy code

```
alert tcp any any -> [Internal Network IP]/24 (msg:"Port scan 
detected"; flags: S; threshold: type threshold, track by_src, count 
20, seconds 10; sid:1000001;)
```

-
  - **Explanation:**
    - This rule triggers an alert if more than 20 SYN packets are sent from a single source IP address within 10 seconds, a common pattern of port scanning.

        - **msg**: Provides a message in the alert log indicating a port scan.
        - **threshold**: Triggers the alert after 20 connection attempts within 10 seconds.

Detected Event Example:
IDS Log:
css
Copy code

```
[Timestamp] ALERT [Source IP: 203.0.113.45] [Destination IP:
10.1.1.100] Port scan detected on ports 22, 80, 443
```

-
    - **Event Details:**
        - The IDS successfully detected a port scanning attempt from an external IP address (203.0.113.45) targeting multiple ports on the internal server (10.1.1.100). The scan included ports 22, 80, and 443. This event triggered an alert to notify the security team.

1.3 Intrusion Prevention System (IPS) Configuration

Objective:
The IPS was configured to prevent certain types of attacks, such as **SYN Floods**, which can overwhelm a target machine and disrupt normal network operations.

IPS Tool Used: Cisco FirePower (IPS module)

IPS Rule Configuration:
A **SYN Flood** prevention rule was configured to automatically block any traffic that exceeds a threshold of SYN packets from a single source IP within a short time window.
IPS Rule Example:
css
Copy code

```
alert tcp any any -> [Internal Network IP] (msg:"SYN Flood Detected";
flags: S; threshold: type limit, track by_src, count 100, seconds 10;
action: drop; sid:2000001;)
```

-
    - **Explanation:**
        - The rule detects a SYN flood by tracking the number of SYN packets from a single source IP.
        - If 100 SYN packets are sent in 10 seconds, the IPS automatically drops the traffic and generates an alert.
        - **action: drop**: Automatically drops the flood traffic to prevent disruption.

Detected Event Example:
IPS Log:
css
Copy code

```
[Timestamp] ALERT [Source IP: 198.51.100.23] SYN flood detected and traffic dropped.
```

-
    - **Event Details:**
        - A SYN flood attack was detected from the external IP address (198.51.100.23), targeting an internal server. The IPS automatically dropped the malicious packets, mitigating the attack.

---

2. Access Control Measures Implementation

2.1 Access Control List (ACL) Configuration

Objective:
To restrict access to sensitive internal resources based on IP addresses and network requirements.

Tool Used: Cisco ASA (Firewall)

ACL Configuration:
An ACL was configured to restrict access to a critical database server (10.1.1.100) such that only a specific subnet (192.168.10.0/24) could access it. All other traffic was denied.
ACL Rule Example:
arduino
Copy code

```
access-list ACL_DB permit ip 192.168.10.0 0.0.0.255 host 10.1.1.100
access-list ACL_DB deny ip any host 10.1.1.100
```

-
    - **Explanation:**
        - The first rule allows IP addresses from the trusted subnet `192.168.10.0/24` to access the database server (`10.1.1.100`).
        - The second rule denies all other access to the database server.

2.2 Access Control Model

Objective:
To establish a formal method for managing user permissions and access rights.

Access Control Model Used: Discretionary Access Control (DAC)

- **DAC Overview:**
  - In the DAC model, the owner of a resource (e.g., a file or database) controls who has access to it.
  - The resource owner can grant permissions (read, write, execute) to other users or groups.
- **Example Configuration:**
  - **Owner**: Database administrator (DBA)
  - **Permissions**: The DBA grants full access (read/write) to the finance team, read-only access to auditors, and no access to non-privileged users.

2.3 User Access Level

Objective:
To ensure that different users have appropriate levels of access to the organization's resources based on their roles.

- **Access Levels:**
  - **Admin User**: Full access to all systems and configurations. Can modify system settings, install software, and access all data.
  - **Standard User**: Limited access to applications and internal resources. Can only access data and services necessary for their job functions.
  - **Guest User**: Access only to public-facing systems (e.g., public web pages), no access to internal systems or data.

Example Configuration for Admin User (on a Linux-based server):
Copy code

```
useradd admin
usermod -aG sudo admin
```

-
  - The **admin** user is added and granted full sudo privileges to manage system configurations.

---

3. Conclusion

This report outlines the successful implementation of key network security measures, including:

- A **firewall rule** to block unauthorized access.
- An **IDS configuration** to detect and alert on suspicious activity like port scanning.
- An **IPS configuration** to prevent DoS attacks such as SYN floods.

- The application of **access control measures** to restrict access to sensitive resources and enforce security policies across the network.

These configurations work together to strengthen the overall security posture of the organization, preventing unauthorized access, detecting malicious activity, and ensuring that access to sensitive data is properly controlled.