# System Security and Hardening

Security Auditing, Network Assessment, Hardening, and Best Practices

Auditing Tools | Assessment | Hardening | System Concepts | Best Practices

Instructor: Dr Shabih Fatima & Tanaya Bowade

# Introduction to Security Auditing

## What is a Security Audit?

A systematic evaluation of an organization's security infrastructure, policies, and practices to identify vulnerabilities before they can be exploited.

- Assesses controls and ensures compliance
- Detects breaches and recommends improvements
- Evaluates security across various aspects

## Security Audit vs. Others

| Feature | Security Audit | Vulnerability Assessment | Penetration Testing |
|---|---|---|---|
| Focus | Comprehensive evaluation | Identifying weaknesses | Exploiting vulnerabilities |
| Approach | Systematic review | Scanning for flaws | Active exploitation |
| Techniques | Inspection, testing | Automated scanning | Simulated attacks |

Security auditing is a critical component of organizational security

# Network Security Assessment Principles

## Assessment Methods

### Penetration Testing
Simulates real-world attacks to identify exploitable weaknesses

### Vulnerability Scanning
Systematically identifies known security flaws and misconfigurations

**Example: Port Scanning with Nmap**

```
$ nmap -sS 192.168.1.100
```

SYN scan to detect open ports without completing TCP handshake

## Assessment Tools

### Nmap
Network discovery and auditing

### Nessus
Vulnerability scanning

### Wireshark
Packet analysis

### Metasploit
Exploitation framework

## Assessment Workflow

1. **Reconnaissance**
   Gathering information about target network and systems (passive and active techniques)
2. **Enumeration**
   Identifying live hosts, open ports, services, and OS versions to map attack surface
3. **Analysis**
   Evaluating identified services for vulnerabilities, misconfigurations, and potential exploits

A systematic approach to identifying and addressing network security vulnerabilities.

### Asset Discovery
Identify all devices on the network.

```
$ nmap -sn 192.168.1.0/24
```

### Vulnerability Scanning
Identify security weaknesses in systems.

```
$ Nessus, Qualys, OpenVAS
```

### Port Analysis
Identify open ports and services.

```
$ nmap example.com
```

### Risk Prioritization
Prioritize risks by impact and likelihood.

```
Use risk matrix to plot vulnerabilities.
```

# Security Auditing with Automated Tools

**Automated security auditing** identifies weaknesses before exploitation.

## Key Auditing Tools

### Lynis
Comprehensive scanner for security hardening and vulnerabilities

### OpenSCAP
SCAP implementation for security policy auditing

### chkrootkit
Rootkit detection tool

## Lynis Audit Example

```
Lynis 3.0.8
---------------------------------------------------

System details
- Distribution: Ubuntu 22.04 LTS
- Kernel: Linux 5.15.0

Security Essentials
- Firewall: [V] UFW (active)
- SSH: [V] OpenSSH (running)
- Malware scanner: [X] Not found

Hardening Index : 75/255

Warnings (1)
- KRNL-5820: Found unbootable kernel entries

Suggestions (5)
- AUTH-9286: Enable two-factor authentication
```

## Audit Workflow

**Scan System**
Run audit tools

→

**Generate Report**
Review findings

→

**Implement Fixes**
Address issues

→

**Re-scan**
Verify fixes

# Security Hardening Strategies

**Security hardening** is the process of reducing a system's attack surface by eliminating potential vulnerabilities and strengthening defenses.

## Disable Unused Services

- Identify and disable unnecessary services
- Disable legacy protocols (Telnet, HTTP)
- Use systemctl to mask services not needed

## Keep Systems Updated

- Regularly apply security patches
- Enable automatic updates for critical packages
- Test updates in staging before production

## Implement Mandatory Access Control

- Configure SELinux with targeted policy
- Implement AppArmor profiles for applications
- Use least privilege for process permissions

## Enforce Strong Authentication

- Use SSH keys instead of passwords
- Implement two-factor authentication
- Configure password policies in PAM

## Principle of Least Privilege

Grant only minimum necessary permissions to users and processes.

Example:

A user should only have access to web server content directory, not system configuration files.

## Comprehensive Patch Management

Systematically identify, test, and apply security updates to all systems.

Linux Example:

```
sudo apt update && sudo apt upgrade
```

Regular monitoring and testing is essential

# Linux Security Hardening Best Practices

Linux powers a significant portion of the internet's infrastructure. Implementing proper security measures is critical.

## File Permissions

- Control access to files and directories
- Three categories: User, Group, Others
- Read (r), Write (w), Execute (x) permissions

Example:

```
chmod 600 sensitive_file.txt
```

## Advanced Attributes (chattr)

- Provides extra layer of security
- Immutable attribute (+i) prevents modification
- Effective against accidental changes

Example:

```
sudo chattr +i /etc/passwd
```

## SSH Hardening

- Disable root login
- Use key-based authentication
- Change default SSH port

Example:

```
PasswordAuthentication no
```

# File Permissions and Advanced Attributes

## Linux File Permissions

Linux permissions control access to files and directories.

### Permission Categories

- User (u) - Owner permissions
- Group (g) - Permissions for group
- Others (o) - Permissions for everyone else

### Permission Types

r Read (4)  w Write (2)  x Execute (1)

### Numeric Notation

6 User: Read (4) + Write (2)

0 Group: No permissions

## Advanced Attributes (chattr)

The chattr command provides advanced file attributes.

### Immutable Attribute (+i)

Prevents file modification, deletion, or renaming.

Even root cannot modify immutable files

### Example

```
$ sudo chattr +i /etc/passwd
```

```
$ sudo chattr -i /etc/passwd
```

Proper permission management is critical for system security

# User and Access Management

## Secure User Policies

### Password Aging
Enforces password expiration policies via chage command

### PAM Configuration
Flexible authentication framework for enforcing policies

### Restrictive Sudo Policy
Configured in /etc/sudoers via visudo

### Least Privilege Principle
Grant only minimum necessary permissions

## Access Control Examples

### Restricting Root Access via SSH

```
# Edit /etc/ssh/sshd_config
PermitRootLogin no
# Restart SSH service
sudo systemctl restart sshd
```

Prevents direct root login over SSH

### Reviewing Sudo Logs

```
# Check sudo logs
grep sudo /var/log/auth.log
```

Provides audit trail of administrative actions

### Roles vs. Privileges Example

| Role | Description | Sudo Privileges |
|---|---|---|
| Web Admin | Manages web server | Restart web service |

# Troubleshooting.....

**You're the new sysadmin for a small web development company. On your first day, you run Lynis and discover:**

- **Hardening score: 68/255**
- **Root login via SSH is enabled**
- **Password authentication is enabled**
- **The /etc/passwd file has 644 permissions (world-readable)**
- **No sudo logging is configured**
- **Three developers share the same 'devops' account with root privileges**

**Then you check the auth logs and see this:**

```
Failed password for root from 185.220.101.50 port 45123 ssh2
Failed password for root from 185.220.101.50 port 45124 ssh2
Failed password for root from 185.220.101.50 port 45125 ssh2
[repeats 47 more times in the last hour]
```

Using the concepts we've covered so far (file permissions, chattr, SSH hardening, sudo policies), **identify THREE most critical security issues here** and **propose a specific fix for each.**

# SSH Security Configuration

Secure Shell (SSH) is a primary method for remote access to Linux servers, making it a frequent target for attackers.

## Disable Root Login

Prevent direct root login to avoid immediate superuser superuser privileges upon successful compromise. compromise.

```
# /etc/ssh/sshd_config
PermitRootLogin no
```

Users should log in with a regular account and use sudo for sudo for administrative tasks.

## Use Key-Based Authentication

Cryptographic key pairs are more secure than password authentication, resistant to brute-force force attacks.

```
# /etc/ssh/sshd_config
PasswordAuthentication no
PubkeyAuthentication yes
```

Disable password authentication once key-based authentication is set up.

## Change Default SSH Port

Changing the default port (22) reduces the volume of automated attacks, though it does not eliminate determined attackers.

```
# /etc/ssh/sshd_config
Port 2222
```

Remember to update any firewall rules to allow traffic on the new port.

After making changes to sshd_config, restart the SSH service for changes to take effect: sudo systemctl restart sshd

Proper SSH configuration is critical for server security

# Firewall Configuration and Network Security

## Firewall Fundamentals

A network security system that monitors and controls incoming and outgoing traffic based on predefined security rules.

- Barrier between trusted/untrusted networks
- Filters traffic based on policies
- Controls access to resources
- Prevents unauthorized access

## Input/Output Chains

### INPUT Chain

Processes packets destined for local system

### OUTPUT Chain

Processes packets originating from local system

## Firewall Tools

### iptables

Traditional, flexible utility for Netfilter. Complex syntax.

### firewalld

Dynamic management with D-Bus interface. Uses zones and services.

### ufw (Uncomplicated)

User-friendly frontend for iptables. Intuitive commands.

## Intrusion Detection

### Fail2Ban

Scans logs for malicious patterns and updates firewall rules.

### Snort

Network intrusion detection system that performs traffic analysis.

# Monitoring and Incident Response

## Continuous Monitoring

### Log Analysis Tools

**journalctl**
Systemd journal logs

**syslog**
General system messages

**auditd**
System call auditing

### Real-time Alerting Systems

**OSSEC**
Host-based IDS

**Wazuh**
Security platform

## Incident Response

**Detect**
Identify incident

**Contain**
Limit impact

**Eradicate**
Remove cause

**Recover**
Restore service

### Log Snippet Example

```
Oct 26 10:30:05 server sshd[12345]: Failed password for invalid user guest from 192.168.1.100 port 54321 ssh2
Oct 26 10:30:08 server sshd[12346]: Failed password for root from 192.168.1.101 port 54322 ssh2
Oct 26 10:30:11 server sshd[12347]: Failed password for invalid user admin from 192.168.1.100 port 54323 ssh2
```

This snippet shows failed SSH login attempts, which could trigger an alert.

# Integrating Security into System Design

## Secure Architecture

- OS-level security is a foundational component of system architecture
- **Secure Boot**: Ensures only trusted software can load during boot process
- **Kernel Module Integrity**: Verifies loaded kernel modules haven't been tampered with

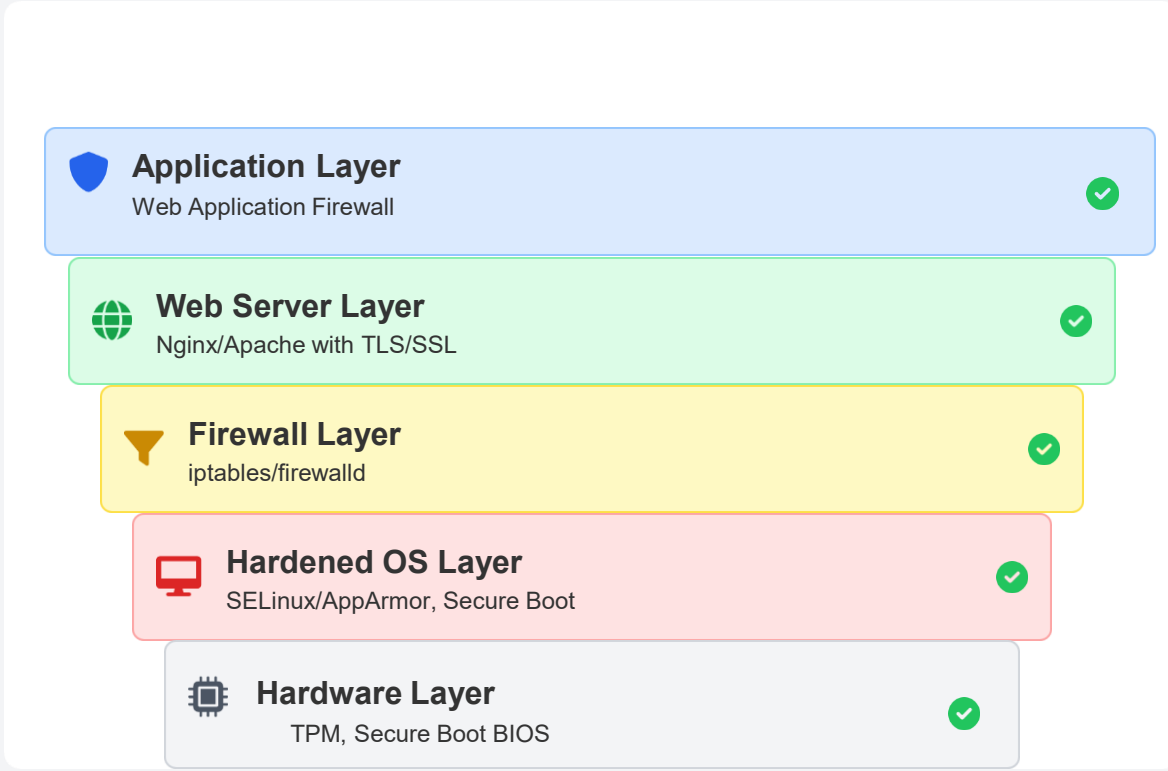
## Automation & Consistency

**Configuration Management**
Ansible, Puppet for uniform security policy enforcement

**Automated Backups**
Regular backups protect against data loss

**Automated Patching**
Ensures systems are promptly updated with security fixes

## Secure Deployment Example




**Key Takeaway:** Security is most effective when integrated into the system architecture from the ground up, not added as an afterthought.

# Review & Test Preparation

Test your knowledge of Linux security concepts with these key questions:

- **What is the primary function of Lynis?**
- **Explain the difference between SUID and SGID.**
- **Which command can display advanced file attributes?**
- **How does ufw simplify firewall configuration?**
- **Name two principles of security hardening.**
- **What's the purpose of auditd in Linux?**
- **How does SELinux enforce access control?**
- **Describe one method to detect rootkits.**

**Study Tip:** Review all slides and practice with these questions.