Week 09: System Security and Hardening

Learning Objectives

- Conduct security assessments using automated tools
- Implement security hardening measures based on audit findings
- Analyse vulnerability reports systematically

Pre-Lab Preparation

Watch the following video lectures before attending the lab session:

- Security Auditing with Automated Tools
- Network Security Assessment Principles
- Security Hardening Strategies
- Review: System Security Concepts

Lab and Coursework Activity: Security Audit and System Hardening

Coursework Phase: Security Audit and System Evaluation

What you will do today:

- Conduct initial security audit with Lynis
- Perform network security assessment with nmap
- Implement security hardening measures
- Re-audit system to measure improvements
- Document complete security audit report
- Verify all security controls are functioning

Part 1: Initial Security Assessment with Lynis

Task 1.1: Installing and Running Lynis

1. On your server via SSH, install Lynis:

   ```
   sudo apt update
   sudo apt install lynis
   ```

2. Check Lynis version:

   ```
   lynis show version
   ```

3. Run comprehensive security audit:

   ```
   sudo lynis audit system
   ```

   This takes several minutes. Review the output carefully.

4. Save the initial audit report:

   ```
   sudo cp /var/log/lynis.log ~/lynis-initial-$(date +%Y%m%d).log
   sudo chown $USER:$USER ~/lynis-initial-*.log
   ```

For your journal: Screenshot showing Lynis audit in progress. Take note of the final hardening index score.

Task 1.2: Analysing Lynis Results

1. Extract the hardening index:

   ```
   grep "Hardening index" ~/lynis-initial-*.log
   ```

2. List all warnings:

   ```
   grep "Warning:" ~/lynis-initial-*.log | tee lynis-warnings.txt
   wc -l lynis-warnings.txt
   ```

3. List all suggestions:

   ```
   grep "Suggestion:" ~/lynis-initial-*.log | tee lynis-suggestions.txt
   wc -l lynis-suggestions.txt
   ```

4. View specific test categories:

   ```
   grep "Test:" ~/lynis-initial-*.log | head -20
   ```

5. Create initial audit summary:

Metric	Initial Value
Hardening Index	/100
Tests Performed	
Warnings Found	
Suggestions Made	

For your journal: Complete audit summary table. List the 10 most critical warnings identified by Lynis with explanations of what each means.

Task 1.3: Prioritising Security Improvements

1. Review warnings and categorise by severity:

   ```
   nano security-priorities.txt
   ```

2. Create prioritisation matrix:

   ```
   HIGH PRIORITY (Security critical, easy to fix):
   1. [Issue]: [Brief description]
      Impact: [Security impact]
      Effort: [Implementation difficulty]

   MEDIUM PRIORITY (Important, moderate effort):
   1. [Issue]: [Brief description]

   LOW PRIORITY (Nice to have, or difficult to implement):
   1. [Issue]: [Brief description]
   ```

3. Identify quick wins (high security impact, low implementation effort)

For your journal: Security prioritisation matrix with justifications for priority levels.

Part 2: Network Security Assessment

Task 2.1: Port Scanning with nmap

IMPORTANT: Only scan systems you own within your isolated VirtualBox environment. Never scan systems you do not own or networks outside your control.

1. On your workstation, ensure nmap is installed:

   ```
   sudo apt install nmap
   ```

2. Perform basic port scan of your server:

   ```
   nmap server_ip
   ```

3. Scan with service version detection:

   ```
   nmap -sV server_ip
   ```

4. Scan common ports with detailed output:

   ```
   nmap -p 1-1000 -sV server_ip
   ```

5. Perform comprehensive scan of all ports:

   ```
   nmap -p- server_ip -oN nmap-all-ports.txt
   ```

   This takes longer. Save results to file.

6. Attempt OS detection (requires sudo):

   ```
   sudo nmap -O server_ip
   ```

7. Review the scan results and identify all open ports

For your journal: Create port inventory table:

Port	State	Service	Version	Justification for Being Open
22	open	ssh	OpenSSH x.x	Required for remote administration

Document why each open port is necessary or should be closed.

Task 2.2: Verifying Firewall Effectiveness

1. On your server, review complete firewall configuration:

```
ssh username@server_ip 'sudo ufw status numbered'
ssh username@server_ip 'sudo ufw status verbose'
```

2. Verify default policies:

```
ssh username@server_ip 'sudo ufw show raw'
```

3. Check iptables rules directly:

```
ssh username@server_ip 'sudo iptables -L -n -v'
```

4. Document firewall configuration:

```
ssh username@server_ip 'sudo ufw status numbered' > firewall-config.txt
```

5. Test firewall from workstation by attempting connections to blocked ports:

```
telnet server_ip 23  # Should fail/timeout
nc -zv server_ip 3306  # MySQL, should fail
```

For your journal: Complete firewall ruleset documentation with justification for each rule. Evidence showing firewall successfully blocks unauthorised connections.

Task 2.3: Analysing Running Services

1. List all listening services:

```
ssh username@server_ip 'sudo ss -tulnp'
```

2. Identify services by process:

```
ssh username@server_ip 'sudo netstat -tulnp'
```

3. List all running services:

```
ssh username@server_ip 'systemctl list-units --type=service --state=running'
```

4. For each listening service, determine:

    - What is the service?
    - Why is it running?
    - Is it necessary?
    - Is it securely configured?

5. Create service inventory:

Service	Port	Protocol	Purpose	Necessary?	Security Measures
sshd	22	TCP	Remote admin	Yes	Key-based auth, no root login

For your journal: Complete service inventory with security justification for each service. Identify any unnecessary services that should be disabled.

Part 3: Security Hardening Implementation

Task 3.1: Implementing Kernel Security Hardening

1. Edit system control configuration:

    ```
    ssh username@server_ip 'sudo nano /etc/sysctl.conf'
    ```

2. Add or modify these kernel security parameters:

    ```
    # Disable IP forwarding (unless server is a router)
    net.ipv4.ip_forward = 0
    net.ipv6.conf.all.forwarding = 0

    # Enable TCP SYN cookie protection (anti-DoS)
    net.ipv4.tcp_syncookies = 1

    # Disable ICMP redirect acceptance
    net.ipv4.conf.all.accept_redirects = 0
    net.ipv4.conf.default.accept_redirects = 0
    net.ipv6.conf.all.accept_redirects = 0
    net.ipv6.conf.default.accept_redirects = 0

    # Disable source packet routing
    net.ipv4.conf.all.accept_source_route = 0
    net.ipv4.conf.default.accept_source_route = 0
    net.ipv6.conf.all.accept_source_route = 0
    net.ipv6.conf.default.accept_source_route = 0

    # Enable IP spoofing protection
    net.ipv4.conf.all.rp_filter = 1
    ```

```
net.ipv4.conf.default.rp_filter = 1

# Log suspicious packets (martians)
net.ipv4.conf.all.log_martians = 1
net.ipv4.conf.default.log_martians = 1

# Ignore ICMP echo requests (ping) - optional
# net.ipv4.icmp_echo_ignore_all = 1

# Ignore broadcast ICMP requests
net.ipv4.icmp_echo_ignore_broadcasts = 1

# Ignore bogus ICMP error responses
net.ipv4.icmp_ignore_bogus_error_responses = 1
```

3. Apply the changes:

   ```
   ssh username@server_ip 'sudo sysctl -p'
   ```

4. Verify settings are applied:

   ```
   ssh username@server_ip 'sudo sysctl net.ipv4.tcp_syncookies'
   ssh username@server_ip 'sudo sysctl net.ipv4.conf.all.accept_redirects'
   ```

For your journal: Document all kernel hardening parameters with explanations of what each does and why it improves security.

Task 3.2: File System Security Hardening

1. Secure shared memory (prevent execution):

   ```
   ssh username@server_ip 'echo "tmpfs /run/shm tmpfs defaults,noexec,nodev,nosuid 0 0" | sudo tee -a /etc/fstab'
   ```

2. Check for world-writable files (potential security risk):

   ```
   ssh username@server_ip 'sudo find / -xdev -type f -perm -0002 -ls 2>/dev/null | head -20'
   ```

3. Check for files with no owner:

   ```
   ssh username@server_ip 'sudo find / -xdev -nouser -o -nogroup 2>/dev/null'
   ```

4. Review SUID/SGID files (can be security risks):

   ```
   ssh username@server_ip 'sudo find / -xdev -type f \( -perm -4000 -o -perm -2000 \) -ls 2>/dev/null' > suid-files.txt
   ```

5. Disable USB storage (if recommended by Lynis and appropriate):

```
ssh username@server_ip 'echo "install usb-storage /bin/true" | sudo tee
/etc/modprobe.d/disable-usb-storage.conf'
```

For your journal: Document file system security measures. Explain the security risks associated with SUID files and world-writable files.

Task 3.3: Password and Authentication Hardening

1. Strengthen password quality requirements:

   ```
   ssh username@server_ip 'sudo nano /etc/security/pwquality.conf'
   ```

2. Add or modify:

   ```
   minlen = 12
   dcredit = -1
   ucredit = -1
   lcredit = -1
   ocredit = -1
   ```

   This requires:

   - Minimum 12 characters
   - At least 1 digit, 1 uppercase, 1 lowercase, 1 other character

3. Configure password ageing:

   ```
   ssh username@server_ip 'sudo nano /etc/login.defs'
   ```

4. Set appropriate values:

   ```
   PASS_MAX_DAYS 90
   PASS_MIN_DAYS 1
   PASS_WARN_AGE 7
   ```

5. Limit login attempts by setting account lockout:

   ```
   ssh username@server_ip 'sudo nano /etc/pam.d/common-auth'
   ```

6. Add this line after pam_unix.so:

   ```
   auth required pam_tally2.so onerr=fail audit silent deny=5 unlock_time=
   900
   ```

For your journal: Document password policy hardening. Discuss the balance between security and usability in password policies.

Task 3.4: Service Hardening and Minimisation

1. List all enabled services:

```
ssh username@server_ip 'systemctl list-unit-files --type=service --state=enabled'
```

2. Identify unnecessary services. Examples that may not be needed on a headless server:

   - bluetooth
   - cups (printing)
   - avahi-daemon (network service discovery)

3. Disable unnecessary services (example):

   ```
   ssh username@server_ip 'sudo systemctl stop bluetooth'
   ssh username@server_ip 'sudo systemctl disable bluetooth'
   ```

4. Verify service is disabled:

   ```
   ssh username@server_ip 'systemctl is-enabled bluetooth'
   ```

5. Create before/after service comparison:

   ```
   # Count enabled services
   ssh username@server_ip 'systemctl list-unit-files --type=service --state=enabled | wc -l'
   ```

For your journal: Document which services you disabled and why. Create table showing enabled services before and after hardening.

Task 3.5: Audit Logging Enhancement

1. Ensure auditd is installed (if recommended by Lynis):

   ```
   ssh username@server_ip 'sudo apt install auditd audispd-plugins'
   ```

2. Enable and start auditd:

   ```
   ssh username@server_ip 'sudo systemctl enable auditd'
   ssh username@server_ip 'sudo systemctl start auditd'
   ```

3. Configure log rotation to prevent disk space exhaustion:

   ```
   ssh username@server_ip 'sudo nano /etc/logrotate.d/rsyslog'
   ```

4. Ensure it includes appropriate rotation settings:

   ```
   /var/log/syslog
   {
       rotate 7
       daily
       missingok
   ```

```
    notifempty
    delaycompress
    compress
    postrotate
        /usr/lib/rsyslog/rsyslog-rotate
    endscript
}
```

5. Check log directory sizes:

```
ssh username@server_ip 'sudo du -sh /var/log/*'
```

For your journal: Document logging enhancements. Explain the importance of audit logs for security incident investigation.

Task 3.6: Re-running Lynis After Hardening

1. Run Lynis audit again:

```
ssh username@server_ip 'sudo lynis audit system'
```

2. Save the post-hardening report:

```
ssh username@server_ip 'sudo cp /var/log/lynis.log ~/lynis-hardened-$(d
ate +%Y%m%d).log'
ssh username@server_ip 'sudo chown $USER:$USER ~/lynis-hardened-*.log'
```

3. Extract new hardening index:

```
ssh username@server_ip 'grep "Hardening index" ~/lynis-hardened-*.log'
```

4. Compare warnings before and after:

```
# Copy both reports to workstation for comparison
scp username@server_ip:~/lynis-*.log .

# Count warnings in each
grep "Warning:" lynis-initial-*.log | wc -l
grep "Warning:" lynis-hardened-*.log | wc -l
```

5. Create comparison table:

Metric	Initial	After Hardening	Improvement
Hardening Index	/100	/100	+X points
Warnings	X	Y	Reduced by Z
Suggestions	X	Y	Reduced by Z
Tests Passed	X	Y	+Z tests

For your journal: Complete comparison table with analysis. Highlight which hardening measures had the most impact on Lynis score.

Task 3.7: Final Security Baseline Verification

1. Run your security baseline script from Week 06:

   ```
   ssh username@server_ip './security-baseline.sh'
   ```

2. Verify all security controls are functioning:

   - ✓ SSH key-based authentication enabled
   - ✓ Password authentication disabled
   - ✓ Root login disabled
   - ✓ Firewall active with appropriate rules
   - ✓ fail2ban running and protecting SSH
   - ✓ AppArmor/SELinux active
   - ✓ Automatic security updates enabled
   - ✓ Non-root administrative user configured
   - ✓ Kernel hardening parameters applied

3. Document any remaining issues or warnings

4. Save final security baseline output:

   ```
   ssh username@server_ip './security-baseline.sh' > final-security-baseline.txt
   ```

For your journal: Screenshot of final security baseline verification showing all controls functioning correctly.

Creating the Security Audit Report

Task 4.1: Compiling the Complete Security Audit Report

1. Create comprehensive security audit report:

   ```
   nano security-audit-report.md
   ```

2. Structure your report:

   ```
   # Security Audit Report

   **System:** [Hostname and IP]
   ```

```markdown
**Operating System:** [Distribution and version]
**Audit Date:** [Date]
**Auditor:** [Student ID]

## Executive Summary

This report documents a comprehensive security audit of [system description].
The audit included automated scanning with Lynis, network security assessment
with nmap, service inventory, and systematic hardening implementation.

**Key Findings:**
- Initial hardening index: [X]/100
- Final hardening index: [Y]/100
- Improvement: [Z] points ([percentage]% increase)
- Critical vulnerabilities addressed: [number]
- Services minimised: [number] disabled

## Audit Methodology

### Tools and Techniques
1. **Lynis** - Comprehensive system security audit
2. **nmap** - Network port and service scanning
3. **Manual review** - Configuration file analysis
4. **Service inventory** - Running service assessment
5. **Custom scripts** - Security baseline verification

### Scope
- Operating system configuration
- Network security
- Access controls
- Authentication mechanisms
- Service security
- Kernel hardening

## Initial Security Assessment

### Lynis Initial Audit Results
- Hardening index: [X]/100
- Total tests performed: [number]
- Warnings identified: [number]
- Suggestions made: [number]

### Critical Issues Identified
1. [Issue 1]: [Description and risk]
2. [Issue 2]: [Description and risk]
3. [Issue 3]: [Description and risk]
[Continue for top 5-10 issues]
```

```markdown
### Network Security Assessment
**Open Ports Identified:**
- Port 22 (SSH): Necessary for administration
- [Other ports]: [Justification or concern]

**Services Listening:**
[List all listening services with risk assessment]

## Security Hardening Implemented

### 1. SSH Hardening
- ✓ Key-based authentication enabled
- ✓ Password authentication disabled
- ✓ Root login disabled
- ✓ Default port maintained (22)

Impact: Eliminates brute-force password attack risk

### 2. Firewall Configuration
- ✓ UFW enabled with deny-by-default policy
- ✓ SSH restricted to specific workstation IP
- ✓ All unnecessary ports blocked

Impact: Reduces attack surface significantly

### 3. Intrusion Detection
- ✓ fail2ban installed and configured
- ✓ SSH jail active (maxretry=3, bantime=600s)
- ✓ Automatic banning of brute-force attempts

Impact: Active defence against automated attacks

### 4. Mandatory Access Control
- ✓ AppArmor/SELinux enabled
- ✓ [X] profiles in enforce mode
- ✓ Regular profile monitoring

Impact: Additional security layer beyond DAC

### 5. Automatic Security Updates
- ✓ unattended-upgrades configured
- ✓ Automatic security updates enabled
- ✓ Update logs monitored

Impact: Timely patching of security vulnerabilities
```

6. Kernel Hardening
Implemented kernel security parameters:
- IP forwarding disabled
- SYN cookies enabled (anti-DoS)
- ICMP redirects disabled
- Source routing disabled
- IP spoofing protection enabled
- Martian packet logging enabled

Impact: Protection against network-level attacks

7. File System Security
- Shared memory secured (noexec)
- World-writable files reviewed
- SUID files inventoried and justified
- USB storage disabled (if applicable)

Impact: Reduces privilege escalation opportunities

8. Password Policy Enhancement
- Minimum length: 12 characters
- Complexity requirements enforced
- Password ageing configured (90 day max)
- Account lockout after 5 failed attempts

Impact: Strengthens authentication security

9. Service Minimisation
Services disabled:
- [Service 1]: Not required for server function
- [Service 2]: Unnecessary attack surface
- [Service 3]: [Justification]

Before hardening: [X] enabled services
After hardening: [Y] enabled services

Impact: Reduced attack surface by [percentage]%

10. Audit Logging
- auditd installed and enabled
- Log rotation configured
- Failed authentication attempts logged

Impact: Improved incident detection and investigation capability

Post-Hardening Assessment

Lynis Final Audit Results

```markdown
- Hardening index: [Y]/100
- Improvement: +[Z] points
- Warnings resolved: [number]
- Remaining warnings: [number]

### Comparison Analysis
[Table showing before/after metrics]

**Most Impactful Improvements:**
1. [Improvement]: Increased score by [X] points
2. [Improvement]: Increased score by [X] points

## Remaining Risks and Limitations

### Accepted Risks
1. **[Risk]**: [Description]
   - **Justification**: [Why not addressed]
   - **Mitigation**: [Compensating controls]

2. **[Risk]**: [Description]
   - **Justification**: [Why not addressed]
   - **Mitigation**: [Compensating controls]

### Recommendations for Future Improvement
1. [Recommendation]: [Implementation guidance]
2. [Recommendation]: [Implementation guidance]
3. [Recommendation]: [Implementation guidance]

## Security Control Verification

All security controls verified functional as of [date]:
- [✓] SSH security (key-based auth, no root login)
- [✓] Firewall protection (UFW active)
- [✓] Intrusion detection (fail2ban active)
- [✓] Mandatory access control (AppArmor/SELinux)
- [✓] Automatic updates (unattended-upgrades)
- [✓] Kernel hardening (sysctl parameters)
- [✓] Service minimisation (unnecessary services disabled)
- [✓] Audit logging (auditd running)

## Conclusion

This security audit successfully identified and addressed [number] security
issues, improving the system's hardening index from [X]/100 to [Y]/100,
representing a [percentage]% improvement. The system now implements multiple
layers of security controls following defence-in-depth principles.
```

```
The most significant improvements were achieved through:
- [Key improvement area 1]
- [Key improvement area 2]
- [Key improvement area 3]

While [number] warnings remain, these have been evaluated and either ac
cepted
as reasonable risks or scheduled for future implementation. The system'
s
security posture is now significantly stronger and aligned with industr
y
best practices.

## Appendices

### Appendix A: Lynis Reports
- Initial audit: lynis-initial-[date].log
- Post-hardening audit: lynis-hardened-[date].log

### Appendix B: Network Scan Results
- nmap scan results: nmap-all-ports.txt

### Appendix C: Configuration Files
- SSH configuration: /etc/ssh/sshd_config
- Firewall rules: firewall-config.txt
- Kernel parameters: /etc/sysctl.conf

### Appendix D: Scripts
- Security baseline verification: security-baseline.sh
- AppArmor/SELinux reporting: apparmor-report.sh

## References

[1] Lynis Project, "Lynis - Security auditing tool," 2024. [Online].
Available: https://cisofy.com/lynis/ [Accessed: XX-Dec-2024]

[2] Ubuntu, "Security - Ubuntu Server Guide," 2024. [Online].
Available: https://ubuntu.com/server/docs/security [Accessed: XX-Dec-20
24]

[3] CIS, "CIS Benchmarks," 2024. [Online].
Available: https://www.cisecurity.org/cis-benchmarks [Accessed: XX-Dec-
2024]

[Add other references as appropriate]
```

3. Complete all sections with your specific data

4. Export as PDF if required or keep as markdown for GitHub Pages

For your journal: Include the complete security audit report. This is a major deliverable for Phase 07.

Journal Entry Requirements for This Week

Your journal entry must include:

Initial Security Audit Section:

- Lynis initial scan results with hardening index score
- Top 10 critical warnings identified with explanations
- Security prioritisation matrix
- nmap scan results with port inventory table
- Service inventory with security justifications

Security Hardening Implementation:

- Detailed documentation of ALL hardening measures implemented
- Before and after configuration file comparisons
- Kernel hardening parameters with explanations
- File system security measures
- Password policy enhancements
- Service minimisation documentation
- Audit logging configuration

Post-Hardening Assessment:

- Lynis post-hardening scan results
- Comparison table (initial vs hardened metrics)
- Analysis of which improvements had most impact
- Final security baseline verification output

Security Audit Report:

- Complete formal security audit report (as structured above)
- Executive summary

- Detailed findings
- Implementation documentation
- Remaining risks assessment
- Recommendations

Network Security:

- Complete port inventory with justifications
- Firewall effectiveness verification
- Evidence that only necessary services are exposed

Reflection:

- Challenges in security hardening
- Trade-offs between security and usability/performance
- Understanding of defence-in-depth principles
- Professional relevance of security auditing
- Connection to learning outcomes (LO3, LO5)

Technical Requirements:

- All screenshots show username@hostname
- Before/after comparisons for all configurations
- Quantitative data (Lynis scores, warning counts, etc.)
- Complete and professional security audit report
- Commit to GitHub: `git add . && git commit -m "Week 09: Security audit and hardening" && git push`
