# Penetration Testing - Student Worksheet

**Name:** **\_\_\_\_** **Date:** **\_\_\_\_**

**Lab Partner:** **\_\_\_\_** **Section:** **\_\_\_\_**

## ⚠️ Ethical Agreement

**I understand that all penetration testing techniques learned in this exercise are for educational purposes only. I agree to only use these techniques in authorized environments and will never attempt unauthorized access to systems I do not own or lack explicit permission to test.**

**Student Signature**: **\_\_\_\_**\_**\_\_\_\_** **Date**: **\_\_\_\_**\_**\_\_\_\_**

## 🔧 Pre-Exercise Setup Verification

**Before starting penetration testing, verify your environment is ready:**

### Step 1: Check Target Environment

# Verify Docker containers are running:  
cd docker && docker-compose ps

**Container Status Check:**

* ☐ cybersec\_sandbox - State: Up
* ☐ vulnerable\_web\_app - State: Up

### Step 2: Verify Target Applications

# Test Flask Application:  
curl -s http://localhost:5000 | head -3  
  
# Test PWA Application:  
curl -s http://localhost:9090 | head -3

**Target Accessibility Check:**

* ☐ Flask App responds at http://localhost:5000 ✅
* ☐ PWA App responds at http://localhost:9090 ✅

### Step 3: Verify All Analysis Tools

# Test SAST tool:  
python src/analyzer/analyze\_cli.py --help  
  
# Test DAST tool:  
python src/analyzer/dast\_cli.py --help  
  
# Test Network Analysis tool:  
python src/analyzer/network\_cli.py --help  
  
# Test basic network tools:  
nmap --version

**Tool Verification Checklist:**

* ☐ SAST CLI available ✅
* ☐ DAST CLI available ✅
* ☐ Network Analysis CLI available ✅
* ☐ Network scanning tools available ✅

### Step 4: Verify Sample Applications

# Test access to suspicious applications:  
ls samples/backdoor-apps/  
ls samples/suspicious-scripts/  
ls samples/resource-abuse/

**Sample Apps Check:**

* ☐ Backdoor applications accessible ✅
* ☐ Suspicious scripts accessible ✅
* ☐ Resource abuse samples accessible ✅

### Troubleshooting:

**If any verification fails:**

1. Restart environment: cd docker && docker-compose down && docker-compose up -d
2. Wait 60 seconds for full startup
3. Re-run verification steps
4. **Contact instructor if issues persist - do not proceed without working environment**

**⚠️ IMPORTANT: Complete ALL verification steps before beginning penetration testing activities.**

## 🎯 Learning Objectives Checklist

By the end of this exercise, I will be able to:

* ☐ Explain penetration testing methodology and ethical considerations
* ☐ Conduct systematic reconnaissance using multiple techniques
* ☐ Integrate SAST, DAST, Network Analysis, and Sandbox findings
* ☐ Perform controlled exploitation in a safe environment
* ☐ Document findings and create professional security reports
* ☐ Understand legal and ethical responsibilities of security professionals

## 📋 Phase 1: Reconnaissance (45 minutes)

### 1.1 Network Discovery

**Objective**: Identify active services and potential entry points

#### Commands to Execute:

# Start network monitoring  
python src/analyzer/network\_cli.py --monitor-connections --educational --duration 300 &  
  
# Discover active services  
python src/analyzer/network\_cli.py --scan-services localhost --educational

#### Documentation Section:

**Services Discovered**: | Port | Service | Version | Risk Level | |——|———|———|————| | | | | | | | | | | | | | | |

**Potential Entry Points Identified**:

### 1.2 Web Application Enumeration

**Objective**: Map web application structure and functionality

#### Commands to Execute:

# Enumerate web directories and files  
python src/analyzer/dast\_cli.py http://localhost:5000 --deep-scan --educational  
python src/analyzer/dast\_cli.py http://localhost:8080 --deep-scan --educational

#### Documentation Section:

**Web Application Structure**:

* **Application 1** (Port 5000):
  + Key endpoints: **\_\_\_\_**
  + Technologies identified: **\_\_\_\_**
  + Interesting files/directories: **\_\_\_\_**
* **Application 2** (Port 8080):
  + Key endpoints: **\_\_\_\_**
  + Technologies identified: **\_\_\_\_**
  + Interesting files/directories: **\_\_\_\_**

### 1.3 Technology Stack Analysis

**Objective**: Understand the underlying technologies and potential vulnerabilities

#### Commands to Execute:

# Analyze application code for technology insights  
python src/analyzer/analyze\_cli.py samples/vulnerable-flask-app --educational --verbose  
python src/analyzer/analyze\_cli.py samples/unsecure-pwa --educational --verbose

#### Documentation Section:

**Technology Stack Summary**:

* **Programming Languages**: **\_\_\_\_**
* **Frameworks**: **\_\_\_\_**
* **Dependencies**: **\_\_\_\_**
* **Potential Technology-Specific Vulnerabilities**: **\_\_\_\_**

### Knowledge Check 1:

**Q1**: What is the difference between active and passive reconnaissance? **A1**: **\_\_\_\_**\_\_\_\_\*\*

**Q2**: Why is it important to understand the technology stack before attempting exploitation? **A2**: **\_\_\_\_**\_\_\_\_\*\*

**Q3**: What ethical considerations apply during the reconnaissance phase? **A3**: **\_\_\_\_**\_\_\_\_\*\*

## 🔍 Phase 2: Vulnerability Assessment (60 minutes)

### 2.1 Static Analysis Integration

**Objective**: Identify code-level vulnerabilities that could be exploited

#### Commands to Execute:

# Comprehensive SAST analysis  
python src/analyzer/analyze\_cli.py samples/vulnerable-flask-app --educational --output reports/pentest\_sast\_flask.json --format json  
python src/analyzer/analyze\_cli.py samples/unsecure-pwa --educational --output reports/pentest\_sast\_pwa.json --format json

#### Vulnerability Classification Table:

| Vulnerability Type | Severity | Exploitable? | Location | Notes |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

### 2.2 Dynamic Testing Results

**Objective**: Identify runtime vulnerabilities and misconfigurations

#### Commands to Execute:

# Comprehensive DAST scans  
python src/analyzer/dast\_cli.py http://localhost:5000 --deep-scan --educational --output reports/pentest\_dast\_flask.json --format json  
python src/analyzer/dast\_cli.py http://localhost:8080 --deep-scan --educational --output reports/pentest\_dast\_pwa.json --format json

#### Runtime Vulnerability Assessment:

**High Priority Targets for Exploitation**:

1. **Vulnerability**: **\_\_\_\_**
   * **Location**: **\_\_\_\_**
   * **Exploitation Potential**: **\_\_\_\_**
   * **Expected Impact**: **\_\_\_\_**
2. **Vulnerability**: **\_\_\_\_**
   * **Location**: **\_\_\_\_**
   * **Exploitation Potential**: **\_\_\_\_**
   * **Expected Impact**: **\_\_\_\_**
3. **Vulnerability**: **\_\_\_\_**
   * **Location**: **\_\_\_\_**
   * **Exploitation Potential**: **\_\_\_\_**
   * **Expected Impact**: **\_\_\_\_**

### 2.3 Network Traffic Patterns

**Objective**: Understand normal vs. suspicious network behavior

#### Commands to Execute:

# Monitor network behavior during testing  
python src/analyzer/network\_cli.py --capture-traffic --duration 300 --educational --output reports/pentest\_network.json --format json

#### Network Analysis Results:

**Normal Traffic Patterns Observed**:

**Suspicious Patterns to Watch For**:

### Knowledge Check 2:

**Q1**: How do SAST and DAST findings complement each other in penetration testing? **A1**: **\_\_\_\_**\_\_\_\_\*\*

**Q2**: What factors determine the exploitability of a vulnerability? **A2**: **\_\_\_\_**\_\_\_\_\*\*

**Q3**: How should vulnerabilities be prioritized for exploitation attempts? **A3**: **\_\_\_\_**\_\_\_\_\*\*

## ⚔️ Phase 3: Controlled Exploitation (90 minutes)

### 3.1 SQL Injection Exploitation

**Objective**: Safely demonstrate SQL injection impact

#### Pre-Exploitation Checklist:

* ☐ Target confirmed to be in sandbox environment
* ☐ Exploitation method reviewed with instructor
* ☐ Documentation template ready
* ☐ Safety procedures understood

#### Exploitation Attempts:

# Test basic SQL injection  
curl -X POST "http://localhost:5000/login" \  
 -d "username=admin' OR '1'='1&password=anything" \  
 -H "Content-Type: application/x-www-form-urlencoded"

**Results Documentation**:

* **Success/Failure**: **\_\_\_\_**
* **Data Accessed**: **\_\_\_\_**
* **Error Messages**: **\_\_\_\_**
* **Impact Assessment**: **\_\_\_\_**

### 3.2 Cross-Site Scripting (XSS) Testing

**Objective**: Demonstrate XSS vulnerabilities and impact

#### Exploitation Attempts:

# Test for reflected XSS  
curl "http://localhost:5000/search?q=<script>alert('XSS')</script>"  
  
# Test for stored XSS  
curl -X POST "http://localhost:5000/comment" \  
 -d "comment=<img src=x onerror=alert('Stored XSS')>" \  
 -H "Content-Type: application/x-www-form-urlencoded"

**Results Documentation**:

* **XSS Type Found**: **\_\_\_\_**
* **Payload Used**: **\_\_\_\_**
* **Response Received**: **\_\_\_\_**
* **Potential Impact**: **\_\_\_\_**

### 3.3 Configuration Exploitation

**Objective**: Exploit misconfigurations and weak settings

#### Exploitation Attempts:

# Test debug mode exposure  
curl "http://localhost:5000/debug" -v  
  
# Test weak authentication  
curl -X POST "http://localhost:8080/login" \  
 -d "username=admin&password=admin" \  
 -H "Content-Type: application/x-www-form-urlencoded"

**Results Documentation**:

* **Configuration Issues Found**: **\_\_\_\_**
* **Information Disclosed**: **\_\_\_\_**
* **Authentication Bypassed**: **\_\_\_\_**
* **Access Gained**: **\_\_\_\_**

### Exploitation Summary Table:

| Vulnerability | Exploitation Success | Impact Level | Evidence Gathered |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

### Knowledge Check 3:

**Q1**: What is the difference between a proof-of-concept and a weaponized exploit? **A1**: **\_\_\_\_**\_\_\_\_\*\*

**Q2**: How do you ensure exploitation activities remain within ethical boundaries? **A2**: **\_\_\_\_**\_\_\_\_\*\*

**Q3**: What should you do if you accidentally access unintended data during testing? **A3**: **\_\_\_\_**\_\_\_\_\*\*

## 🔍 Phase 4: Post-Exploitation Analysis (45 minutes)

### 4.1 Persistence and Access Maintenance

**Objective**: Understand how attackers maintain long-term access

#### Simulation Commands:

# Monitor persistent connections  
python samples/backdoor-apps/backdoor\_app.py &  
python src/analyzer/network\_cli.py --monitor-connections --duration 120 --educational

**Persistence Mechanisms Observed**:

1. **Method**: **\_\_\_\_**
   * **Detection Difficulty**: **\_\_\_\_**
   * **Persistence Duration**: **\_\_\_\_**
2. **Method**: **\_\_\_\_**
   * **Detection Difficulty**: **\_\_\_\_**
   * **Persistence Duration**: **\_\_\_\_**

### 4.2 Data Exfiltration Patterns

**Objective**: Recognize data theft techniques and patterns

#### Simulation Commands:

# Monitor exfiltration patterns  
python samples/network-scenarios/backdoor\_simulation.py 90 &  
python src/analyzer/network\_cli.py --capture-traffic --duration 100 --educational

**Exfiltration Analysis**:

* **Data Types Targeted**: **\_\_\_\_**
* **Exfiltration Methods**: **\_\_\_\_**
* **Traffic Patterns**: **\_\_\_\_**
* **Detection Indicators**: **\_\_\_\_**

### 4.3 Impact Assessment

**Objective**: Evaluate the full business impact of successful exploitation

#### Impact Analysis Framework:

**Confidentiality Impact**:

* **Data Accessed**: **\_\_\_\_**
* **Sensitivity Level**: **\_\_\_\_**
* **Affected Parties**: **\_\_\_\_**

**Integrity Impact**:

* **Data Modified**: **\_\_\_\_**
* **System Changes**: **\_\_\_\_**
* **Trust Implications**: **\_\_\_\_**

**Availability Impact**:

* **Services Affected**: **\_\_\_\_**
* **Downtime Potential**: **\_\_\_\_**
* **Recovery Complexity**: **\_\_\_\_**

### Knowledge Check 4:

**Q1**: What is the difference between impact and exploitability in risk assessment? **A1**: **\_\_\_\_**\_\_\_\_\*\*

**Q2**: How do you assess the business impact of a technical vulnerability? **A2**: **\_\_\_\_**\_\_\_\_\*\*

**Q3**: What factors determine how long an attacker can maintain access? **A3**: **\_\_\_\_**\_\_\_\_\*\*

## 📊 Phase 5: Professional Reporting (60 minutes)

### 5.1 Executive Summary Draft

**Objective**: Communicate risk to business stakeholders

#### Executive Summary Template:

**Assessment Overview**:

* **Target Environment**: **\_\_\_\_**
* **Testing Duration**: **\_\_\_\_**
* **Methodology Used**: **\_\_\_\_**

**Key Findings**:

* **Critical Vulnerabilities**: **\_** (Number)
* **High-Risk Issues**: **\_** (Number)
* **Successfully Exploited**: **\_** (Number)

**Business Risk Level**: ❑ Critical ❑ High ❑ Medium ❑ Low

**Top 3 Recommendations**:

### 5.2 Technical Findings Report

**Objective**: Provide detailed technical information for remediation

#### Vulnerability Details Template:

**Vulnerability 1**:

* **Title**: **\_\_\_\_**
* **Severity**: ❑ Critical ❑ High ❑ Medium ❑ Low
* **CVSS Score**: **\_**
* **Location**: **\_\_\_\_**
* **Description**: **\_\_\_\_**
* **Exploitation Steps**: **\_\_\_\_**
* **Impact**: **\_\_\_\_**
* **Remediation**: **\_\_\_\_**
* **Evidence**: **\_\_\_\_**

**Vulnerability 2**:

* **Title**: **\_\_\_\_**
* **Severity**: ❑ Critical ❑ High ❑ Medium ❑ Low
* **CVSS Score**: **\_**
* **Location**: **\_\_\_\_**
* **Description**: **\_\_\_\_**
* **Exploitation Steps**: **\_\_\_\_**
* **Impact**: **\_\_\_\_**
* **Remediation**: **\_\_\_\_**
* **Evidence**: **\_\_\_\_**

### 5.3 Risk Prioritization Matrix

| Vulnerability | Exploitability | Impact | Risk Score | Priority |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

### 5.4 Remediation Roadmap

**Objective**: Provide actionable steps for security improvement

#### Immediate Actions (0-30 days):

#### Short-term Improvements (1-3 months):

#### Long-term Strategy (6-12 months):

### Knowledge Check 5:

**Q1**: How should technical findings be communicated differently to executives vs. developers? **A1**: **\_\_\_\_**\_\_\_\_\*\*

**Q2**: What factors should influence vulnerability remediation prioritization? **A2**: **\_\_\_\_**\_\_\_\_\*\*

**Q3**: How do you balance technical accuracy with business communication needs? **A3**: **\_\_\_\_**\_\_\_\_\*\*

## 🎓 Self-Assessment and Reflection

### Technical Skills Self-Evaluation

Rate your performance in each area (1=Needs Improvement, 5=Excellent):

* **Reconnaissance**: 1 - 2 - 3 - 4 - 5
* **Vulnerability Assessment**: 1 - 2 - 3 - 4 - 5
* **Exploitation**: 1 - 2 - 3 - 4 - 5
* **Post-Exploitation Analysis**: 1 - 2 - 3 - 4 - 5
* **Report Writing**: 1 - 2 - 3 - 4 - 5

### Ethical Understanding Self-Check

**Q1**: Do I fully understand the legal and ethical boundaries of penetration testing? ❑ Yes, completely ❑ Mostly ❑ Somewhat ❑ Need more training

**Q2**: Am I confident I can apply these skills ethically in real-world scenarios? ❑ Yes, completely ❑ Mostly ❑ Somewhat ❑ Need more training

**Q3**: Do I understand the professional responsibilities of security practitioners? ❑ Yes, completely ❑ Mostly ❑ Somewhat ❑ Need more training

### Reflection Questions

**1. What was the most challenging aspect of this penetration testing exercise?**

**2. How has this exercise changed your understanding of cybersecurity?**

**3. What ethical dilemmas did you encounter during the exercise?**

**4. How would you explain the value of penetration testing to a business owner?**

**5. What additional skills or knowledge do you need to develop for a cybersecurity career?**

## 🔄 Integration Review

### Connection to Previous Exercises

**How did SAST findings inform your penetration testing approach?**

**How did DAST results guide your exploitation attempts?**

**How did Network Analysis help you understand attack patterns?**

**How did Sandbox Analysis inform your post-exploitation assessment?**

### Career and Next Steps

**Interest in Cybersecurity Career**: ❑ Very interested ❑ Somewhat interested ❑ Neutral ❑ Not interested

**Areas of Cybersecurity Most Interesting**: ❑ Penetration Testing ❑ Security Analysis ❑ Incident Response ❑ Security Architecture ❑ Compliance ❑ Forensics ❑ Risk Management ❑ Security Awareness

**Next Learning Goals**:

**Instructor Use Only**

**Student Performance Summary**:

* **Technical Competency**: \_\_\_/40 points
* **Professional Skills**: \_\_\_/30 points
* **Ethical Understanding**: \_\_\_/30 points
* **Total Score**: \_\_\_/100 points

**Additional Comments**: