# Cybersecurity Asset Management: Completing the process

**Activity 2: Vulnerability-Based Asset Prioritization**

**Overview**

**Title:** Risk-Based Asset Classification  
**Scenario:** Management has asked you to prioritize which assets need immediate security attention based on actual vulnerabilities.

**Files**

**docker-compose.vulnerable.yml**

yaml

version: '3'

services:

*# Deliberately vulnerable web application*

dvwa:

image: vulnerables/web-dvwa

container\_name: dvwa-app

ports:

- "8085:80"

networks:

- scan-network

*# Outdated WordPress for vulnerability scanning*

old-wordpress:

image: wordpress:4.6 *# Old version with known vulnerabilities*

container\_name: outdated-blog

ports:

- "8086:80"

environment:

WORDPRESS\_DB\_HOST: db

WORDPRESS\_DB\_USER: exampleuser

WORDPRESS\_DB\_PASSWORD: examplepass

WORDPRESS\_DB\_NAME: exampledb

networks:

- scan-network

*# Database for WordPress*

db:

image: mysql:5.5 *# Old MySQL version*

container\_name: old-mysql

environment:

MYSQL\_ROOT\_PASSWORD: rootpass

MYSQL\_DATABASE: exampledb

MYSQL\_USER: exampleuser

MYSQL\_PASSWORD: examplepass

networks:

- scan-network

*# No external ports - internal only*

*# Modern secure application for contrast*

secure-app:

image: nginx:latest

container\_name: secure-webapp

ports:

- "8087:80"

networks:

- scan-network

networks:

scan-network:

driver: bridge

**Tasks**

**Part 1: Environment Setup**

bash

*# Start the vulnerable environment*

sudo docker-compose -f docker-compose.vulnerable.yml up -d

*# Discover the network and assets*

sudo docker network inspect kali-container-lab\_scan-network

**Part 2: Comprehensive Vulnerability Scanning**

bash

*# Deep vulnerability scanning with nikto*

sudo nikto -h http://localhost:8085 -Format csv -o dvwa-vuln-scan.csv

sudo nikto -h http://localhost:8086 -Format csv -o wordpress-vuln-scan.csv

sudo nikto -h http://localhost:8087 -Format csv -o secureapp-vuln-scan.csv

*# Service version detection for CVEs*

sudo nmap -sV --script vuln 172.20.0.0/24

**Part 3: CVSS Scoring & Prioritization**

Research CVSS scores for found vulnerabilities and create:

**Deliverables:**

1. vulnerability-priority-list.csv with columns:
   * Asset Name, Vulnerability, CVSS Score, Exploit Available, Patch Available, Priority (Critical/High/Medium/Low)
2. remediation-plan.txt with:
   * Immediate actions (next 24 hours)
   * Short-term fixes (1 week)
   * Long-term solutions (1 month)

**Activity 3: Asset Management Policy Creation**

**Overview**

**Title:** Policy Development & Compliance  
**Scenario:** The CISO wants you to draft the company's first formal Asset Management Policy.

**Tasks**

**Part 1: Research & Analysis**

Research these frameworks and identify 5 asset management requirements from each:

* NIST Cybersecurity Framework
* ISO 27001
* PCI-DSS

**Part 2: Policy Development**

Create asset-management-policy.md with these sections:

1. **Purpose & Scope**
2. **Asset Classification Scheme**
3. **Inventory Requirements**
4. **Ownership & Responsibilities**
5. **Lifecycle Management**
6. **Security Controls by Classification**
7. **Compliance & Auditing**
8. **Policy Review Schedule**

**Part 3: Implementation Guide**

Create implementation-checklist.txt with:

* 30-day immediate actions
* 90-day short-term goals
* 1-year long-term objectives

**Part 4: Compliance Assessment**

Using your Activity 1 results, grade the company's current compliance with your policy (A-F scale).

**Deliverables**

1. Complete asset management policy document
2. Implementation roadmap
3. Current state compliance assessment
4. Gap analysis presentation (5 slides)

**Activity 4: Asset Monitoring & Change Detection**

**Overview**

**Title:** Continuous Monitoring & Alerting  
**Scenario:** Implement monitoring to detect unauthorized assets and changes.

**Files**

**docker-compose.monitoring.yml**

yaml

version: '3'

services:

*# Base infrastructure*

web-server:

image: nginx:alpine

container\_name: prod-webserver

ports:

- "8090:80"

networks:

- prod-network

database:

image: postgres:13

container\_name: prod-database

networks:

- prod-network

*# Monitoring container*

monitoring:

image: alpine:latest

container\_name: asset-monitor

networks:

- prod-network

command: tail -f /dev/null *# Keep container running*

networks:

prod-network:

driver: bridge

**Tasks**

**Part 1: Baseline Establishment**

bash

*# Create initial asset baseline*

sudo nmap -sS -O 172.21.0.0/24 > baseline-scan.txt

sudo docker ps --format "table {{.Names}}\t{{.Image}}\t{{.Ports}}" > container-baseline.txt

**Part 2: Unauthorized Asset Injection**

bash

*# Instructor/TA will add this container during the lab*

sudo docker run -d --name rogue-server --network kali-container-lab\_prod-network nginx:alpine

**Part 3: Change Detection Scripting**

Create asset-monitor.sh that:

1. Scans the network and compares to baseline
2. Checks for new containers
3. Alerts on unexpected open ports
4. Generates a diff report

**Part 4: Alert System Design**

Design a simple alert system in alert-design.txt:

* What constitutes an "alertable" event?
* Who gets notified for different alert types?
* What's the response procedure?

**Deliverables**

1. Working change detection script
2. Alert system design document
3. Incident report for the rogue server
4. Monitoring policy recommendations

**Activity 5: Business Impact Analysis**

**Overview**

**Title:** Asset Valuation & Business Impact  
**Scenario:** Calculate the actual business value of assets to justify security spending.

**Tasks**

**Part 1: Asset Valuation**

For each asset from Activity 1, calculate:

1. **Replacement Cost:** Hardware/software replacement
2. **Data Value:** Cost to recreate lost data
3. **Downtime Cost:** Revenue loss per hour of outage
4. **Reputation Impact:** Estimated brand damage cost

**Part 2: Business Impact Analysis**

Create business-impact-worksheet.csv with columns:

* Asset, Function, Max Tolerable Downtime, Recovery Time Objective, Recovery Point Objective, Single Loss Expectancy, Annualized Loss Expectancy

**Part 3: Budget Justification**

Using your calculations, create a security-budget-proposal.docx that:

* Justifies $50,000 for security controls
* Shows ROI for each proposed control
* Prioritizes based on risk reduction per dollar

**Part 4: Executive Presentation**

Create a 5-slide presentation for management explaining:

1. Our most valuable assets
2. Biggest risks
3. Recommended investments
4. Expected risk reduction
5. Compliance benefits

**Deliverables**

1. Complete asset valuation spreadsheet
2. Business impact analysis
3. Budget proposal document
4. Executive presentation

**Activity 6: Incident Response Integration**

**Overview**

**Title:** Asset Management in Incident Response  
**Scenario:** A breach has occurred! Use your asset management data to respond effectively.

**Scenario Brief**

"Alert! We've detected malicious activity from IP 172.19.0.3. The attacker may have compromised multiple systems."

**Tasks**

**Part 1: Rapid Asset Identification**

bash

*# Emergency asset discovery*

sudo nmap -T5 -A 172.19.0.0/24 > incident-scan.txt

*# Container forensic analysis*

sudo docker ps -a > container-forensic.txt

sudo docker network inspect kali-container-lab\_corp-network > network-forensic.txt

**Part 2: Impact Assessment**

Using your asset inventory, determine:

1. Which assets are definitely compromised?
2. Which assets are potentially at risk?
3. What business processes are affected?
4. What data might be exposed?

**Part 3: Containment Strategy**

Create containment-plan.txt with:

1. Immediate isolation procedures
2. Communication plan
3. Evidence preservation steps
4. Business continuity actions

**Part 4: Lessons Learned**

Write incident-after-action-report.md covering:

* How asset management helped/delayed response
* Gaps in our asset knowledge
* Improvements needed

**Deliverables**

1. Incident scan results and analysis
2. Containment plan
3. Communication timeline
4. After-action report with improvement recommendation