Asset Discovery Activity: Corporate Network Discovery - Complete Activity

**Scenario:** You are a cybersecurity intern tasked with performing the first-ever asset inventory for a small company.

**Part 0: Environment Setup**

**Step 1: Verify Docker Installation**

bash

*# Check if Docker is installed and running*

sudo docker --version

sudo systemctl status docker

*# If Docker is not installed, run:*

sudo apt update

sudo apt install docker.io

sudo systemctl start docker

sudo systemctl enable docker

*# Add your user to docker group (optional, for convenience)*

sudo usermod -aG docker $USER

*# Log out and back in for group changes to take effect*

**Step 2: Create Lab Directory and Files**

bash

*# Create lab directory*

mkdir kali-container-lab

cd kali-container-lab

**Step 3: Create Docker Compose File**

Create a file called docker-compose.yml with the following content:

yaml

version: '3'

services:

*# Web server 1 - Company website*

web-server:

image: nginx:alpine

container\_name: company-website

ports:

- "8080:80"

networks:

- corp-network

*# Web server 2 - Internal portal*

internal-portal:

image: httpd:alpine

container\_name: employee-portal

ports:

- "8081:80"

networks:

- corp-network

*# Database server (no external ports)*

database:

image: postgres:13

container\_name: company-db

environment:

POSTGRES\_PASSWORD: "test123"

POSTGRES\_DB: "company\_data"

networks:

- corp-network

*# No ports exposed externally - internal only!*

*# Legacy system*

legacy-app:

image: nginx:alpine

container\_name: legacy-system

ports:

- "8082:80"

networks:

- corp-network

networks:

corp-network:

driver: bridge

**Step 4: Start the Corporate Network**

bash

*# Download and start all containers*

sudo docker-compose up -d

*# Verify all containers are running*

sudo docker ps

*# Expected output: 4 containers running*

**Part 1: Network Discovery (25 minutes)**

**Step 1: Discover the Docker Network**

bash

*# List all Docker networks*

sudo docker network ls

*# Inspect our specific corporate network*

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

**Task:** Record the subnet range and container IP addresses.

**Step 2: Scan for Active Hosts**

bash

*# Use the actual subnet from the inspect command (e.g., 172.19.0.0/16)*

sudo nmap -sn 172.19.0.0/24

**Task:** How many active hosts did you find? List their IP addresses.

**Step 3: Port Scanning**

bash

*# Scan internal container IPs*

sudo nmap -sS -T4 172.19.0.2 172.19.0.3 172.19.0.4 172.19.0.5

*# Scan external access points (localhost mapped ports)*

sudo nmap -sS -p 8080,8081,8082,5432 localhost

**Document Your Findings:** Create a table of discovered hosts and open ports.

**Part 2: Service Identification (25 minutes)**

**Step 4: Service Detection**

bash

*# Identify services and versions*

sudo nmap -sV -sC 172.19.0.2 172.19.0.3 172.19.0.4 172.19.0.5

**Step 5: Web Asset Discovery**

bash

*# Discover hidden web content on each service*

sudo gobuster dir -u http://localhost:8080/ -w /usr/share/wordlists/dirb/common.txt -o scan-8080.txt

sudo gobuster dir -u http://localhost:8081/ -w /usr/share/wordlists/dirb/common.txt -o scan-8081.txt

sudo gobuster dir -u http://localhost:8082/ -w /usr/share/wordlists/dirb/common.txt -o scan-8082.txt

*# View results*

cat scan-8080.txt

cat scan-8081.txt

cat scan-8082.txt

**Step 6: Vulnerability Assessment**

bash

*# Quick security scans*

sudo nikto -h http://localhost:8080 -o nikto-8080.txt

sudo nikto -h http://localhost:8081 -o nikto-8081.txt

sudo nikto -h http://localhost:8082 -o nikto-8082.txt

*# View results*

cat nikto-8080.txt

**Document Your Findings:** Record service versions and any discovered vulnerabilities or information disclosures.

**Part 3: Asset Documentation**

**Step 7: Complete the Asset Inventory**

Create a file called asset-inventory.csv with the following columns:

| Asset Name | Type | Internal IP | External Port | Software | Owner | Classification | Risk Level | Notes |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |

**Classification Guide:**

* **Public:** Information for general public
* **Internal:** Company internal use only
* **Confidential:** Sensitive business information
* **Restricted:** Highly sensitive (customer data, passwords)

**Risk Level Guide:**

* **Critical:** Business would fail if compromised
* **High:** Significant financial/operational impact
* **Medium:** Moderate disruption
* **Low:** Minimal impact

**Step 8: Create Network Diagram**

Create a simple diagram showing:

* Internet connection
* External services (ports 8080, 8081, 8082)
* Internal services (database)
* Network boundaries

**Step 9: Risk Assessment Report**

Create a file called risk-assessment.txt with:

1. Highest risk asset and why
2. Most important security finding
3. Top 3 recommendations for management

**Expected Results**

**Discovered Assets:**

1. **company-website** - Web Server (nginx) - External facing - Port 8080
2. **employee-portal** - Web Server (Apache) - External facing - Port 8081
3. **company-db** - Database (PostgreSQL) - Internal only - No external ports ✅
4. **legacy-system** - Web Server (nginx) - External facing - Port 8082

**Key Security Findings:**

* ✅ Database is properly isolated (security best practice)
* ⚠️ Employee portal exposed externally (should be internal)
* ⚠️ Legacy system may be unnecessary (increases attack surface)
* ⚠️ Default configurations reveal server information

**Deliverables**

Submit the following by end of lab:

* docker-compose.yml file
* asset-inventory.csv - Complete asset spreadsheet
* Network diagram (can be hand-drawn and photographed)
* risk-assessment.txt - Summary report
* Screenshots of key scan results