**LAB07 Detecting security vulnerability using AI agent**

**Class: M03 student ID: B2111933 Name: Truong Dang Truc Lam**

1. **Search one coding model and clone**

**A. AI agent code example for detecting security vulnerability of IP 127.0.0.1**

B. Simplified AI agent code example for detecting security vulnerability of IP 127.0.0.1

C. AI agent code example for detecting security vulnerability of url https://ctu.ac.vn

D. Simplified AI agent code example for detecting security vulnerability of url https://ctu.ac.vn

**2. Select environment/IDE => LAB06 slide**

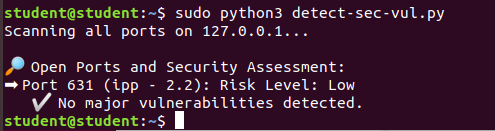
Python3 on Ubuntu

**3. Design System Architecture**

| Items | Resource | Other server |
| --- | --- | --- |
| VM type | Ubuntu |  |
| OS | Ubuntu |  |
| IP/URL | 127.0.0.1 |  |
| Attacking type | Network scanning, Vulnerability detection |  |
| Detecting type | Port scanning, Vulnerability scanning |  |
| Language,version | Python 3.x |  |
| Libraries | nmap (python-nmap), re (regular expressions) |  |
| Library Package | python-nmap |  |
| Algorithm |  |  |
| AI Tech. type | Network vulnerability detection, CVE matching |  |
| Database type |  |  |
| Software tool | Nmap (for scanning) |  |
| Python (for scripting) |  |
| Code Reference | The Python script shared above |  |
| Dataset reference |  |  |

**4. Execute the code and print out.** **Show used resources**

- Start the http server on Ubuntu VM

  
Result of the code (from ChatGPT)

**5. Screen-shot each step full screen Show used resources**

- Start the http server on Ubuntu VM

- Run the attack file and see the requests sent on Windows host

Note: From ChatGPT with love

import nmap

import re

def scan\_all\_ports(ip):

scanner = nmap.PortScanner()

print(f"Scanning all ports on {ip}...")

# Scan all ports with service detection and vulnerability scripts

scanner.scan(ip, "1-65535", arguments="-sV --script vuln")

open\_ports = []

for host in scanner.all\_hosts():

for port, port\_info in scanner[host].get("tcp", {}).items():

if port\_info.get("state") == "open":

open\_ports.append({

"port": port,

"service": port\_info.get("name", "unknown"),

"version": port\_info.get("version", "unknown"),

"vulnerabilities": port\_info.get("script", {})

})

return open\_ports

def analyze\_vulnerabilities(port\_data):

risk\_level = "Low"

vulnerabilities = port\_data.get("vulnerabilities", {})

if vulnerabilities:

for script, output in vulnerabilities.items():

if re.search(r"vulnerable|CVE-\d{4}-\d+", output, re.IGNORECASE):

risk\_level = "High"

break

elif "potential" in output.lower():

risk\_level = "Medium"

return risk\_level

if \_\_name\_\_ == "\_\_main\_\_":

target\_ip = "127.0.0.1"

results = scan\_all\_ports(target\_ip)

if results:

print("\n🔎 Open Ports and Security Assessment:")

for port\_info in results:

port = port\_info["port"]

service = port\_info["service"]

version = port\_info["version"]

risk = analyze\_vulnerabilities(port\_info)

print(f"➡ Port {port} ({service} - {version}): Risk Level: {risk}")

if risk == "High":

print(" ⚠️ Immediate action required: Check for security patches.")

elif risk == "Medium":

print(" 🟠 Review security settings and restrict access if necessary.")

else:

print(" ✅ No major vulnerabilities detected.")

else:

print("✅ No open ports detected.")

6. Explain the coding process (executed resource)

-Set up the http server on Ubuntu:

**1. Import Dependencies**

nmap: This library allows interaction with the Nmap tool, which is a popular network scanner used to discover open ports and check for vulnerabilities on hosts.

re: The regular expression library is used to match patterns (like CVE IDs or the word "vulnerable") in the vulnerability scan output.

**2. Function to Scan All Ports**

scan\_all\_ports(ip):

This function accepts an IP address (ip) and scans all ports (from 1 to 65535) on that IP.

* It uses nmap.PortScanner() to initiate a port scan.
* The scanner.scan(ip, "1-65535", arguments="-sV --script vuln") line scans all ports on the given IP.
  + -sV: Detects the version of services running on the open ports.
  + --script vuln: Runs Nmap’s vulnerability detection scripts to check for known vulnerabilities.
* It collects all open ports with details like:
  + Port number
  + Service running (e.g., HTTP, FTP)
  + Version of the service
  + Detected vulnerabilities associated with that service.

**3. Function to Analyze Vulnerabilities**

analyze\_vulnerabilities(port\_data):

* This function takes the vulnerability data for a particular port and analyzes it to determine a risk level.
* It looks for keywords like "vulnerable" or CVE (Common Vulnerabilities and Exposures) identifiers in the output.
  + If it finds a CVE or the word "vulnerable", it flags the risk level as High.
  + If it finds something indicating a potential issue (e.g., the word "potential"), it flags it as Medium.
* If no serious vulnerabilities are found, it assigns a Low risk level.

**4. Main Block**

The main block of the code runs when the script is executed directly. It does the following:

* Calls the scan\_all\_ports() function for 127.0.0.1 (localhost) to scan all its ports.
* If open ports are found, it loops through each port's details (service, version, vulnerabilities).
* For each open port, it calls analyze\_vulnerabilities() to determine the security risk level.
* Depending on the risk level, it prints:
  + High risk: Immediate action needed (like patching or disabling services).
  + Medium risk: Review security settings.
  + Low risk: No major vulnerabilities detected.
* If no open ports are found, it prints a message indicating that.

**Conclusion**

This script scans all ports on 127.0.0.1, checks the services running on those ports, detects vulnerabilities, and assesses the risk for each open port.

It then provides security recommendations based on the risk level found for each port.