Week Four

Project

Jose Sandoval

National University

CYB333 Security Automation

Dr. Senica Woodruff

June 30, 2024

Abstract

This paper is going support the final project for CYB333 Security Automation. The project asked us to create a Python tool to automate security-related functions and provide a project plan, scope, demonstration of the tool and upload it to GitHub with the link submitted.

Week Four

Project Plan & Scope

**Objectives and Goal:**

* Build a vulnerability scanner tool to be used to identify weaknesses in the network.
* The vulnerability scanner will scan for all devices in the network and port to verify which are active and not. This will aid in verifying all devices within our network and ensure we have not been compromised. The scanner will allow for identifying weak networks and devices not allowed to be in the network.

**Security Tools and Libraries:**

* We will need to use the Python Standard Library and ‘Socket” module for network communications. We will also use ‘scapy’ and ‘nmap’ library for networks scanning as well.

**Development Tools:**

* The developmental tools used for this vulnerability scanner will be python language on PyCharm. We will also use GitHub to share the Vulnerability scanner.

**Plan Execution:**

* The vulnerability scanner will be a script file that will be run through the terminal line.

**Python Vulnerability Tool**

**impor****t** nmap  
  
  
def scan\_network():  
 nm = nmap.PortScanner()  
  
 # request for what network to scan  
 target\_network = input("Enter the network to scan (e.g., 192.168.1.0/24): ")  
  
 print(f"Scanning network {target\_network}...")  
 nm.scan(hosts=target\_network, arguments='-sn') # Perform host discovery  
  
 # List discovered hosts  
 print("\nHosts discovered:")  
 for host in nm.all\_hosts():  
 print(f"Host: {host} \tState: {nm[host].state()}")  
  
 # request to scan a specific port  
 if input("\nDo you want to scan specific ports? (y/n): ").lower() == 'y':  
 target\_hosts = list(nm.all\_hosts())  
 ports\_to\_scan = input("Enter specific ports to scan (e.g., 80,443,22): ")  
  
 # Convert ports to a list  
 ports\_list = [int(port.strip()) for port in ports\_to\_scan.split(',') if port.strip().isdigit()]  
  
 # Scan specific ports   
 for host in target\_hosts:  
 print(f"\nScanning ports on {host}...")  
 nm.scan(hosts=host, ports=','.join(map(str, ports\_list)), arguments='-sS')  
  
 # Display results for specific ports  
 for proto in nm[host].all\_protocols():  
 print(f"Protocol: {proto}")  
 ports = nm[host][proto].keys()  
 for port in ports:  
 state = nm[host][proto][port]['state']  
 print(f"Port: {port}\tState: {state}")  
  
 print("\nScan complete.")  
  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 scan\_network()

**Tool Demonstration**

**Prepare User Documentation:**

* Create user documentation that explains how to install, configure, and use the tool. Include examples, step-by-step guides, and troubleshooting tips.

**Demonstration:**

* Demonstrate the tool's functionality to potential users or stakeholders. Show how it automates security-related tasks and provide real-world examples.

**Deployment and Distribution:**

* Decide on how you'll distribute the tool. You might package it as an executable, distribute it through a package manager, or host it online. Ensure that the deployment aligns with the tool's architecture.

**Feedback and Improvement:**

* Collect user feedback and be open to improvements. Consider maintaining the tool and releasing updates to address new security challenges.

References

ITProTV. (n.d.). https://app.acilearning.com/course/version-control/overview-version-control