User Stories for CSCD 350 Spring 2024

ScanMasterX

V 1.0

Team #10: UwUltimate Stardust Crusaders

Submitted By

Lewis Thomas	Lthomas32@ewu.edu
William Reese	Wreese1@ewu.edu
Eric Leachman	Eleachman04@ewu.edu
Dennis Vinnikov	dvinnikov@ewu.edu
Alexa Darrington	adarrington@ewu.edu

Instructor: Sanmeet Kaur

GSA: Harley Davis, Dominic MacIsaac

Lab Section: Rm 219 – Section 2

Date: 04/27/2024

GitHub Repository: https://github.com/Sanmeet-EWU/github-teams-project-bid-uwultimate-stardust-crusaders/tree/main

Section 1 – User Stories

U1 – GUI to interact with the menu options for our ScanMasterX application.

As a customer, I would like a user interface to interact with the program easily and choose options.

Assumptions and Details:

- We will have a GUI with menus and comprehensive prompts and output.
- We will use Tkinter and subnetting to achieve this.

Acceptance Criteria:

Given a GUI with a menu of options

And a user deciding what to test

When they choose an option

Then our GUI program will display desired information.

U2 – Component to display network topology.

As a customer, I would like a component to display my network's topology.

Assumptions and Details:

- We will scan using NMAP and visualize and display the output for the user.
- This will be a foundational component within our program.

Acceptance Criteria:

Given some network information

And the NMAP framework

When our tool maps out the topology

Then it will be displayed in a visualized format.

U3 - Component to scan multiple subnets.

As a business owner, I want to be able to scan multiple subnets with this tool to cover all my bases.

Assumptions and Details:

- We will gather information from the user to scan all possible subnets.
- NIST, MITRE and Metasploit will be used.

Acceptance Criteria:

Given a businesses' subnets

And our programs use of frameworks

When they want to see vulnerable ports

Then our program will provided that output

U4 – Ability to scan a client for protocol security risks.

As a home network enthusiast, I want to ensure I am using secure protocols on my systems, so that I can have peace of mind that my home network is secure.

Assumptions and Details:

- The protocols are commonly known protocols that have been tested.

Acceptance Criteria:

Given this program, When I scan my systems, Then the scan should give information on the protocols I am running on my systems.

U5 – Visualization of network vulnerabilities.

As a grandma who knows nothing about the internet, I want a visual product that can give me feedback on how vulnerable my home network is, so that I can be more educated and cautious about my internet habits.

Assumptions and Details:

- The software is pre-installed.
- The software is intuitive enough for the technologically non-savvy to use.

Acceptance Criteria:

Given this program

When I run the program

Then some information is construed in a clear decisive manner so that lay persons can make decisions and be educated.

U6 – Implement a component to run commonly used exploits.

As a developer, I need to identify my vulnerable machines so that I can prevent remote access.

Assumptions and Details:

- I will only be able to exploit known exploits.
- It will either pass or fail.

Acceptance Criteria:

Given a virtual machine

And virtual machine is on the same network

When I test the machine

Then I should either be given access or fail.

U7 – Ability to identify vulnerabilities on a server.

As a business owner, I would like to be able to easily scan vulnerabilities on my server.

Assumptions and Details:

- The user would like to get a detailed report on how to better secure their servers.

Acceptance Criteria:

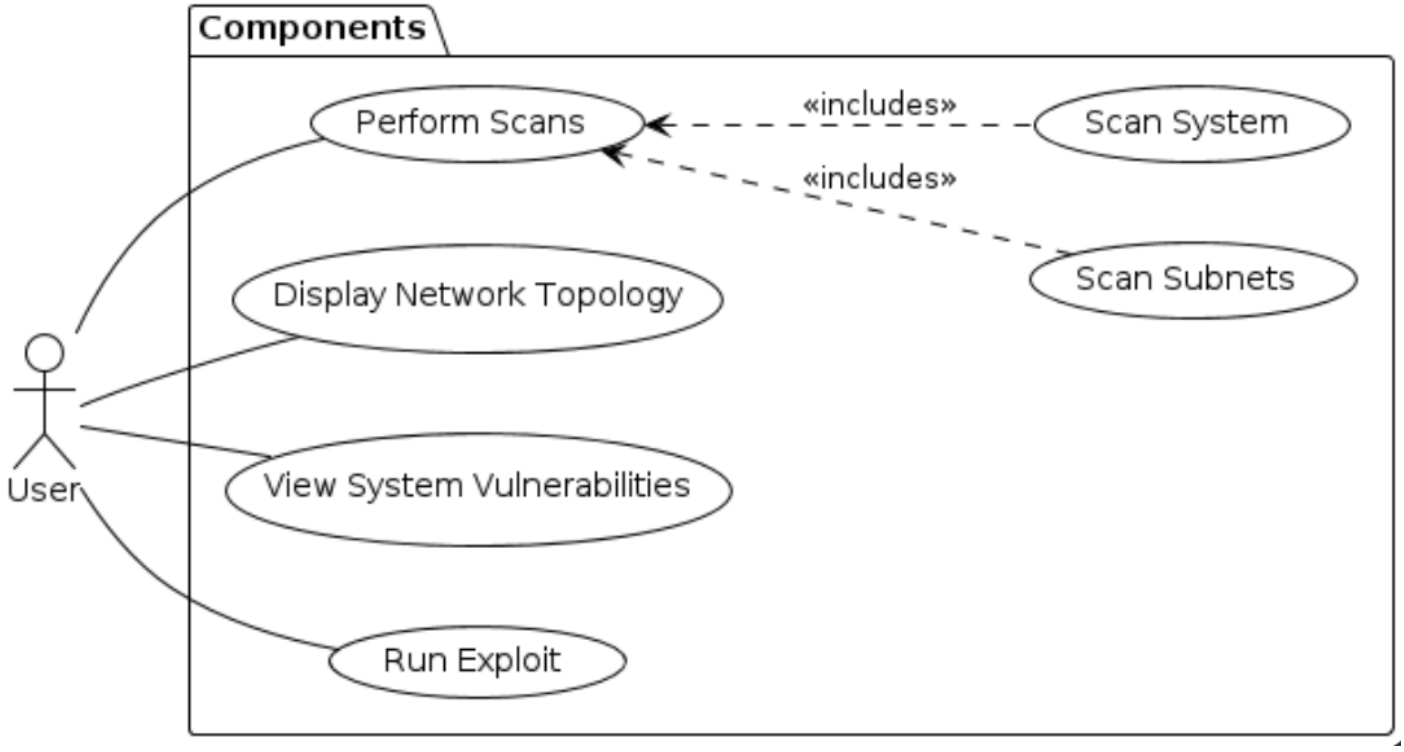
Given a server owner

And they wish to secure their servers

When they run a scan on their server

Then they will get a detailed report of potential vulnerabilities.

Section 2 – Use Case Diagram



Section 3 – Requirements & Specifications

Section 3.1 – Server Vulnerability Identifier Component

[Req 1, U5, U7 – Vulnerability Identifier] There must be a feature to identify vulnerabilities on a server.

- **[Spec 1.1]** The publicly available Metasploit API will be used to launch exploits and consequentially identify vulnerabilities.
- **[Spec 1.2]** The user may choose to run the vulnerability scanner in terse mode, meaning that the whole test will end once a single vulnerable port is found.

Section 3.2 – Network Security Component Requirements

[Req 2, U3 – Port Evaluation] There must be a component that will evaluate the security of the ports on a given server.

- **[Spec 2.1]** The NMAP API will be used to verify whether an open port has the potential to be exploited.

[Req 3, U3, U5, U4 – Port Selection] The user must be able to select which device(s), which ports, or which group of ports, that they wish to test and scan.

- **[Spec 3.1]** A list of the most used ports will be provided to the user as a selectable option for the scan.
- **[Spec 3.2]** The user will have the option to scan their local device.
- **[Spec 3.3]** The user will have the option to scan their local network.
- **[Spec 3.4]** The user will have the option to scan the provided IP address.

[Req 4, U3 – Subnet Scanner] The port scanning component must have the option to scan subnets.

Section 4 – Glossary

Exploit: a segment of code or a program that maliciously takes advantage of vulnerabilities or security flaws in software or hardware.

GUI: graphical user interface, a way for user to interact with the app by manipulating graphical elements such as icons, buttons, sliders and menus.

Network Security: the protection of the underlying networking infrastructure from unauthorized access, misuse, or theft.

Network Topology: the physical and logical arrangement of nodes and connections in a network.

NMAP: Network Mapper, a free and open-source tool used for vulnerability checking, port scanning, and network mapping.

Server: a computer or computer program which manages access to a centralized resource or service in a network.

Subnet: a part of a larger network such as the internet.

Virtual Machine: a computer system created using software on one physical computer in order to emulate the functionality of another separate physical computer.