PORTIFY – SCAN IT OUT

A

Project Report

submitted in partial fulfillment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE

Specialization in

Cyber Security and Forensics

By:

Name	Roll No
Manik Garg	R134216076
Kritika Sharma	R134216072
Manila Chawla	R134216077
Vikalp	R134216150

Under the guidance of

Dr. Susheela Dahiya

Assistant Professor (S.G.)

Department of Computer Applications

UPES

UNIVERSITY WITH A PURPOSE

Department of Systemics

School of Computer Science

UNIVERSITY OF PETROLEUM AND ENERGY STUDIES

Bidholi, Via Prem Nagar, Dehradun, Uttarakhand

2019-2020

UNIVERSITY WITH A PURPOSE

CANDIDATES DECLARATION

I/We hereby certify that the project work entitled **Portify – Scan It Out** in partial fulfilment of the requirements for the award of the Degree of Bachelor of Technology in Computer Science And Engineering with Specialization in Cyber Security and Forensics and submitted to the Department of Systemics at School of Computer Science, University of Petroleum And Energy Studies, Dehradun, is an authentic record of my/ our work carried out during a period from **January, 2020** to **April, 2020** under the supervision of **Dr. Susheela Dahiya, Assistant Professor (S.G.), Department of Computer Application**.

The matter presented in this project has not been submitted by me/ us for the award of any other degree of this or any other University.

Manik Garg
R134216076

Kritika Sharma
R134216072

Manila Chawla
R134216077

Vikalp
R134216150

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

(Date: 18 April 2020)

(Dr. Susheela Dahiya)
Project Guide

Dr. Neelu Jyoti Ahuja
Head
Department of Systemics
School of Computer Science
University of Petroleum and Energy Studies
Dehradun - 248 001 (Uttarakhand)

ACKNOWLEDGEMENT

We wish to express our deep gratitude to our guide **Dr. Susheela Dahiya**, for all advice, encouragement and constant support she has given us throughout our project work. This work would not have been possible without her support and valuable suggestions.

We sincerely thank to our Head of the Department, **Dr. Neelu Jyoti Ahuja**, for her great support in doing our **Portify – Scan it out** at **SoCS**.

We are also grateful to **Dr. Manish Prateek, Dean SOCS**, UPES, for giving us the necessary facilities to carry out our project work successfully.

We would like to thank all our friends for their help and constructive criticism during our project work. Finally, we have no words to express our sincere gratitude to our parents who have shown us this world and for every support they have given us.

Name:	**Manik Garg**	**Kritika Sharma**	**Manila Chawla**	**Vikalp**
Roll No.	**R134216076**	**R134216072**	**R134216077**	**R134216150**

ABSTRACT

The idea is to create a tool for finding vulnerable ports on a target IP. The tool will list down the status of all the ports on the target URL or IP address, basis the port range entered by the user. Furthermore, the tool will provide the number of vulnerabilities associated with each of the open ports with the list of vulnerable ports along with the associated vulnerabilities and mitigation methods. The project will consist of independent modules of each framework and one cumulative frontend system. Incremental model will be used to develop the tool as this model provides flexibility to the developer for adding further specifications in form of independent modules and allows programmer to present the executable tool in a shorter duration of time.

Keywords: Security, Port Scanning, Data, Risk, Threat, Vulnerability.

Table of Contents

LIST OF TABLES

List of Tables

LIST OF FIGURES

List of Figures

1. Introduction

Security is one of the most important aspect in current world scenario. Data being the most important and cloistered part of our lives, needs to be protected and for that purpose we need to implement various security controls at all places which are involved in any sort of data related operations. Security controls refer to the mechanisms used to restrict the access to the data or assets to maintain confidentiality, integrity and availability thus maintaining the CIA Triad.

Information security risks can be defined as consequence of uncertainty on information security objectives. A control is a measure implemented to prevent from or to reduce the impact of security risks. A control can decrease the risk by reducing the possibility of an event, the impact or both. Information security risk management is very important for business, government, and for individuals in order to protect their information. To manage the risks, organizations need to assess the security risks to their valuable assets and plan for mitigating control actions to address these risks.

In computer networking, a port is a communication endpoint. At the software level, within an operating system, a port is a logical concept that detects a definite process or a category of network service. Ports can be known for each protocol which is a combination by 16-bit unsigned numbers, commonly known as the port number. An open port is an attack surface which can be vulnerable to remotely executable attacks. An important belief in security is reducing your attack surface and ensure that servers have the least number of unprotected services. The best way to identify an open port is to check if socket connection can be made or not.

Socket: A socket is one endpoint of a two-way communication link between two programs running on the network. A socket is bound to a port number so that the TCP layer can identify the application that data is destined to be sent to.

Different Ports have different services associated with them. Each of the service has its own sets of vulnerabilities, thus its own sets of risks. Each service maybe associated to one or more ports but vice versa is not true. For some services, there are different ports available depending on the type of security like for HTTP/HTTPS & FTP/SFTP.

Threat Identification: An assessment identifying the actual threats to critical assets and infrastructure in order to prioritize security measures;

Vulnerability Assessment: An assessment addressing vulnerability of a port facility by identifying weaknesses that may pose as a likely target; and

Risk Assessment: Identification and evaluation of the potential impacts or consequences that could occur as a result of the threats and vulnerabilities.

Following is a table of common port numbers:

Table 1: Common services with the protocols

Protocol	TCP/UDP	Port Number
File Transfer Protocol (FTP)	TCP	20/21
Secure Shell (SSH)	TCP	22
Telnet	TCP	23
Simple Mail Transfer Protocol (SMTP)	TCP	25
Domain Name System (DNS)	TCP/UDP	53
Dynamic Host Configuration Protocol (DHCP)	UDP	67/68
Trivial File Transfer Protocol (TFTP)	UDP	69
Hypertext Transfer Protocol (HTTP)	TCP	80
Post Office Protocol (POP) version 3	TCP	110
Network Time Protocol (NTP)	UDP	123
NetBIOS	TCP/UDP	137/138/139
Internet Message Access Protocol (IMAP)	TCP	143
Simple Network Management Protocol (SNMP)	TCP/UDP	161/162
Border Gateway Protocol (BGP)	TCP	179

2. Literature Review

[1] Software Engineering by K.K. Aggarwal & Yogesh Singh Third Edition published by New Age International Publishers 2008: We have used this resource to study about various life cycle models and take the best suitable model for our project, which helps in producing quality software that is maintainable, delivered on time within budget and also satisfies it's requirements.

[2] Owasp.org. 2020. OWASP Top Ten: We referred the OWASP top ten vulnerabilities to learn about the necessity of conducting port scanning in order to identify vulnerabilities which can be exploited due to the open ports identified.

[3] Pentest-Tools.com. 2020. Online Port Scanner with Nmap - Discover Open TCP Ports | Pentest-Tools.Com: We visited an online port scanner in order to identify how generally the port scanners function and understand the minimum inputs that must be required from the user's end to conduct a successful scan.

[4] Wilkins, S., 2020. TCP/IP Ports and Protocols | TCP/IP Ports and Protocols | Pearson IT Certification: This article looks at the common protocols, provides a basic description of their function and lists the port numbers that they are commonly associated with. This helped us identify the most common port numbers and brainstorm our way to identify what vulnerabilities could be present.

3. Problem Statement

An open port is an attack surface which could be vulnerable to a remotely exploitable vulnerability. Even though the port scanning can present the status (open or closed) of the ports, it's still difficult to prioritize the resolution of the associated vulnerability with each of the open ports.

4. Objective

To create a vulnerability mapping tool that identifies vulnerable ports on a URL or target IP provided by the user and presents the results based on the port scanner along with the mitigation methods associated with each vulnerability.

5. Design Methodology

For the structured development of the project, we are using the Incremental SDLC Model. Following are the phases that marks the completion of tasks framed for that phase:

System Feasibility: This phase helped us in analyzing the feasibility of the project.

Software Plans & Requirements: During this phase we analyzed all the requirements and planned the development cycle.

Product Design: We developed zero Level DFD in this phase to know how the data would interact with the tool and the modules. The main modules identified for the project are:

- DNS Resolution
- Port Scanning
- Vulnerability Mapping
- Mitigation Methods

This will mark the end of requirement verification

Detailed Design: Here we worked on use case diagram and sequence diagram to define how sub-modules would be implemented as the part of main module.

Implementation: The implementation phase was when we coded all the modules.

Integration: This phase aimed at making the product out of the modules i.e. we integrated the modules.

Testing: This was our system testing phase. We tested all the merged modules. This acted as level one testing for our product.

Operations & Maintenance: This is our last phase where we worked towards achieving feedback and further improvements were done accordingly.

In order to identify whether a port is open or not, the best way is to try and make a socket connection. We have imported the *java.net* package in the Java platform, which provides a class, Socket, that implements one side of a two-way connection. Once the necessary details are entered by the user, i.e., target URL or IP address and port range, scan on the port range starts.

The status of the port, whether the port is ‘Open’ or ‘Closed’, needs to be stored. We have used HashMap to store the status. *HashMap* provides the basic implementation of the Map interface of Java. It stores the data in (Key, Value) pairs. To access a value, one must know its key. Here, the port number is the key and status (open / closed) is the value.

In case the port status is identified as open, the vulnerabilities associated with that port are displayed and if the user wishes to see the mitigation techniques of the displayed vulnerabilities.

5.1. Product Design

Level – 0 Data flow Diagram:

A level 0 data flow diagram (DFD), also known as a context diagram, shows a data system as a whole to understand the flow of the data and accentuates the way it interacts with external objects.

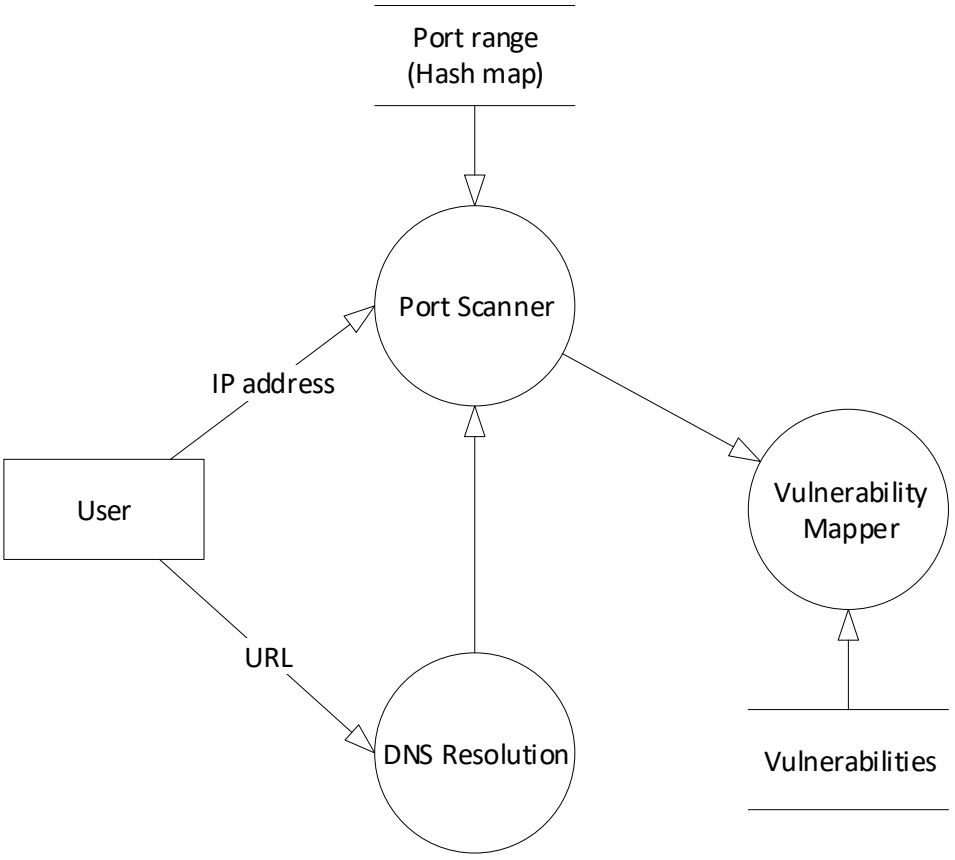

Figure 1: Level - 0 Data Flow Diagram

5.2. Detailed Design

a. Use Case Diagram

A use case diagram is a depiction of a user's communication with the system that shows the association between the user and the different use cases in which the user is involved.

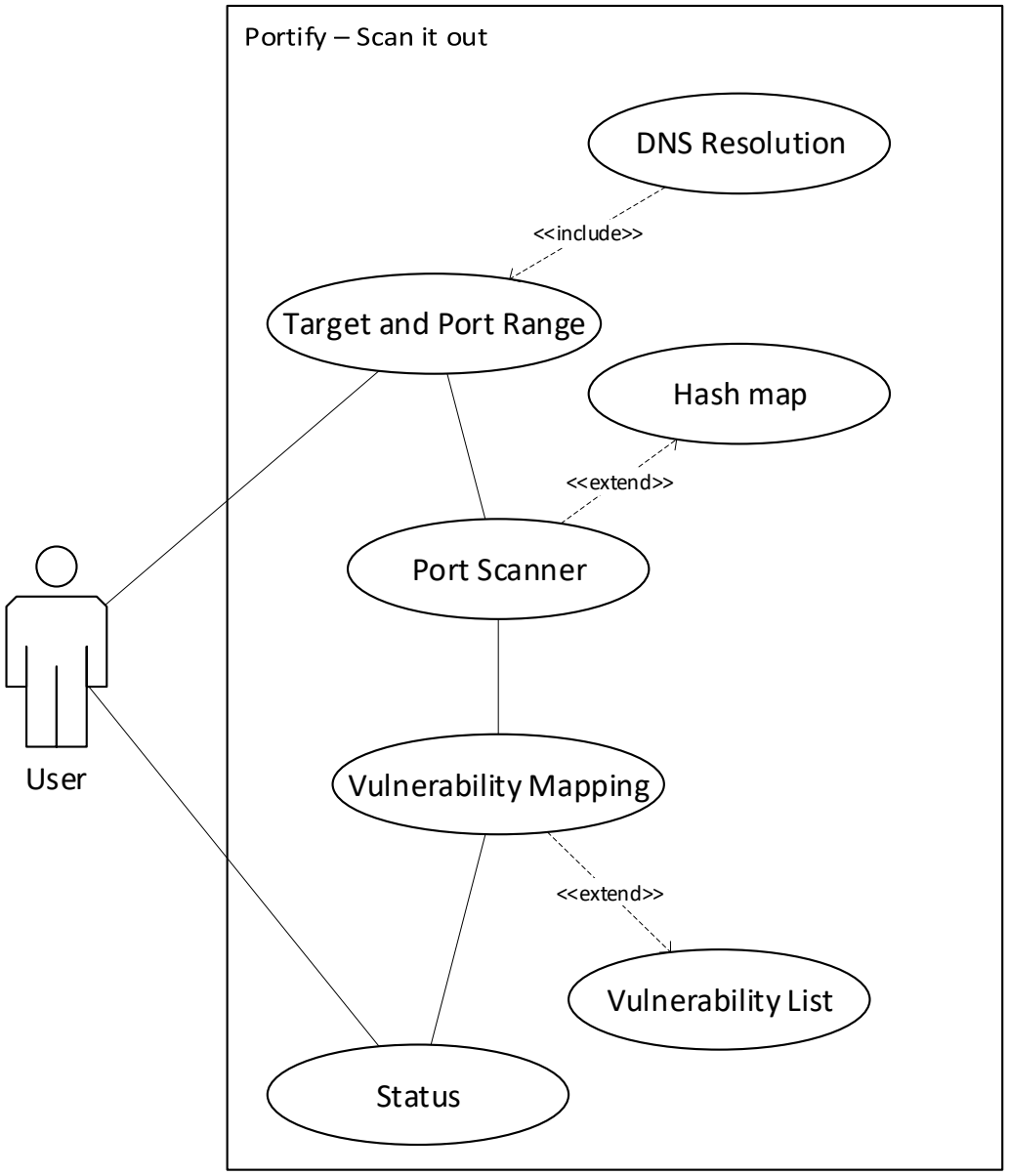

Figure 2: Use Case Diagram

b. Sequence Diagram

A sequence diagram shows object communications represented in time arrangement. It portrays the objects and classes involved in the situation and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario.

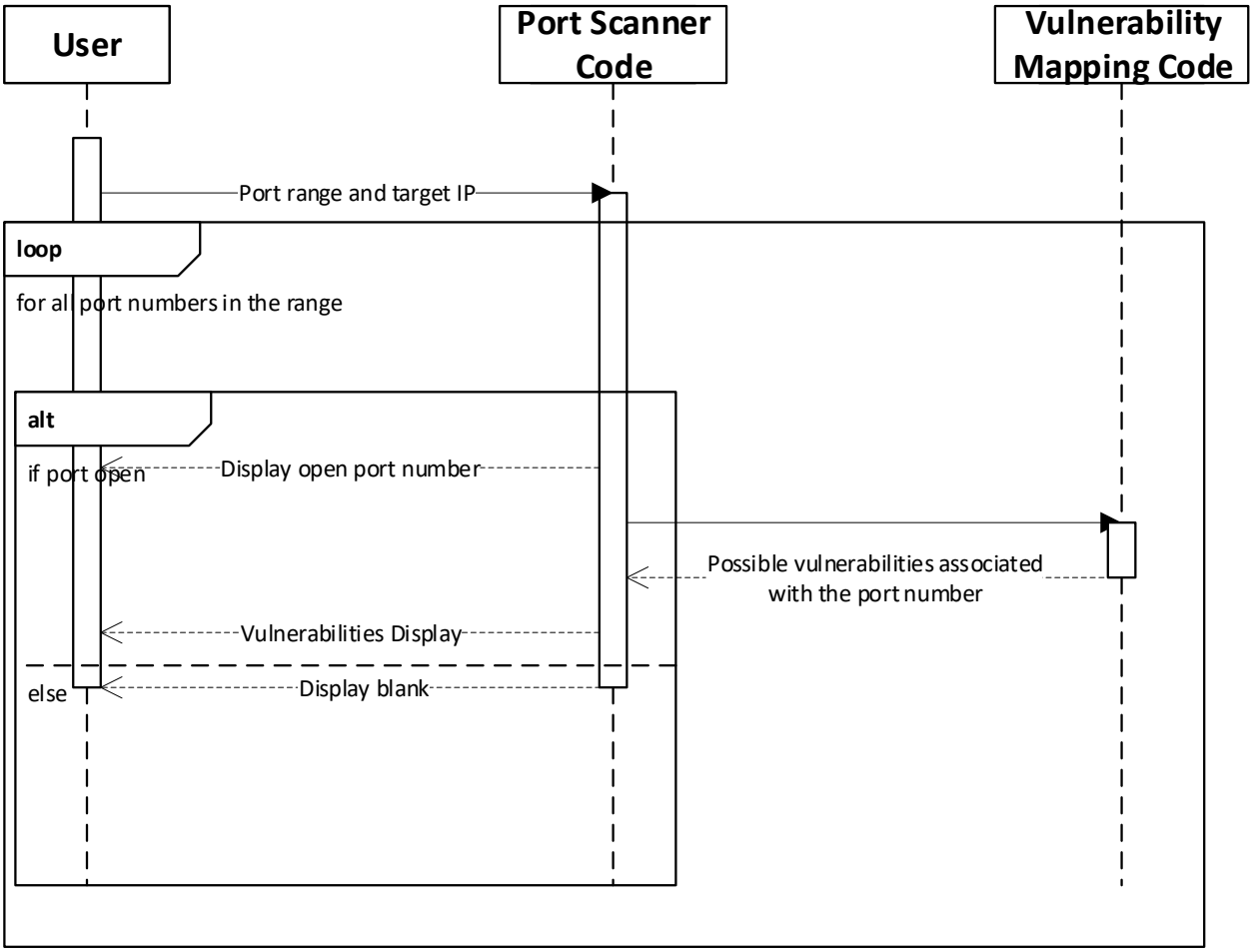

Figure 3: Sequence Diagram

6. Implementation

The port range and URL or target IP are given as input on the main interface. Once the user fills the details, the scanning begins along with a progress loader. Afterwards the complete status list is provided to the user along with the vulnerability status and risk rating associated with it.

The port scanner is created by making a socket connection to the target IP for each port and store the status as open and closed based on the connection success in a HashMap.

Steps: -

1. Create a module for main menu interface.
2. Create independent modules for port scanning and vulnerability listing.
3. Create a module for mapping open ports to associated vulnerabilities
4. Club the Modules using Bottom Up approach.

6.1. Pseudocode

1. Import following java packages:
    i. import java.util.*;
    ii. For Socket class: import java.net.*;
    iii. For HashMap class: import java.util.HashMap;

2. User can enter the following details:
    i. Target URL or IP address
    ii. Port Range

3. Implement HashMap for storing the port number and its status:
   HashMap<Integer, String> port = new HashMap<>

4. Create a socket connection:
   Socket soc = new Socket(ip,i);

5. Store the status in the HaspMap:
    i. In case the port is open: port.put(i,"Open");
    ii. In case the port is closed: port.put(i,"Closed");

6. Display the vulnerabilities associated with open ports

7. Display the mitigation techniques of the associated vulnerabilities.

6.2. Output Screen

```
WELCOME TO PORT BASED VULNERABILITY SCANNER


Enter the starting Port Address:
80
Enter the ending Port Address:
81
Enter the target ip address as w.x.y.z :
google.com
```

Figure 4: Output Screen 1 - Enter details

```
Starting Port Scan on host: google.com
----------100%

Current Port Status for target google.com is

{80=Open, 81=Closed}

Vulnerability Status:

Port 80:
Service Name: HTTP
Vulerabilities:
Do you want to display the mitigation guidelines for Common Vulnerabilities (Y/N)?
y
```

Figure 5: Output Screen 2 - Port scan result

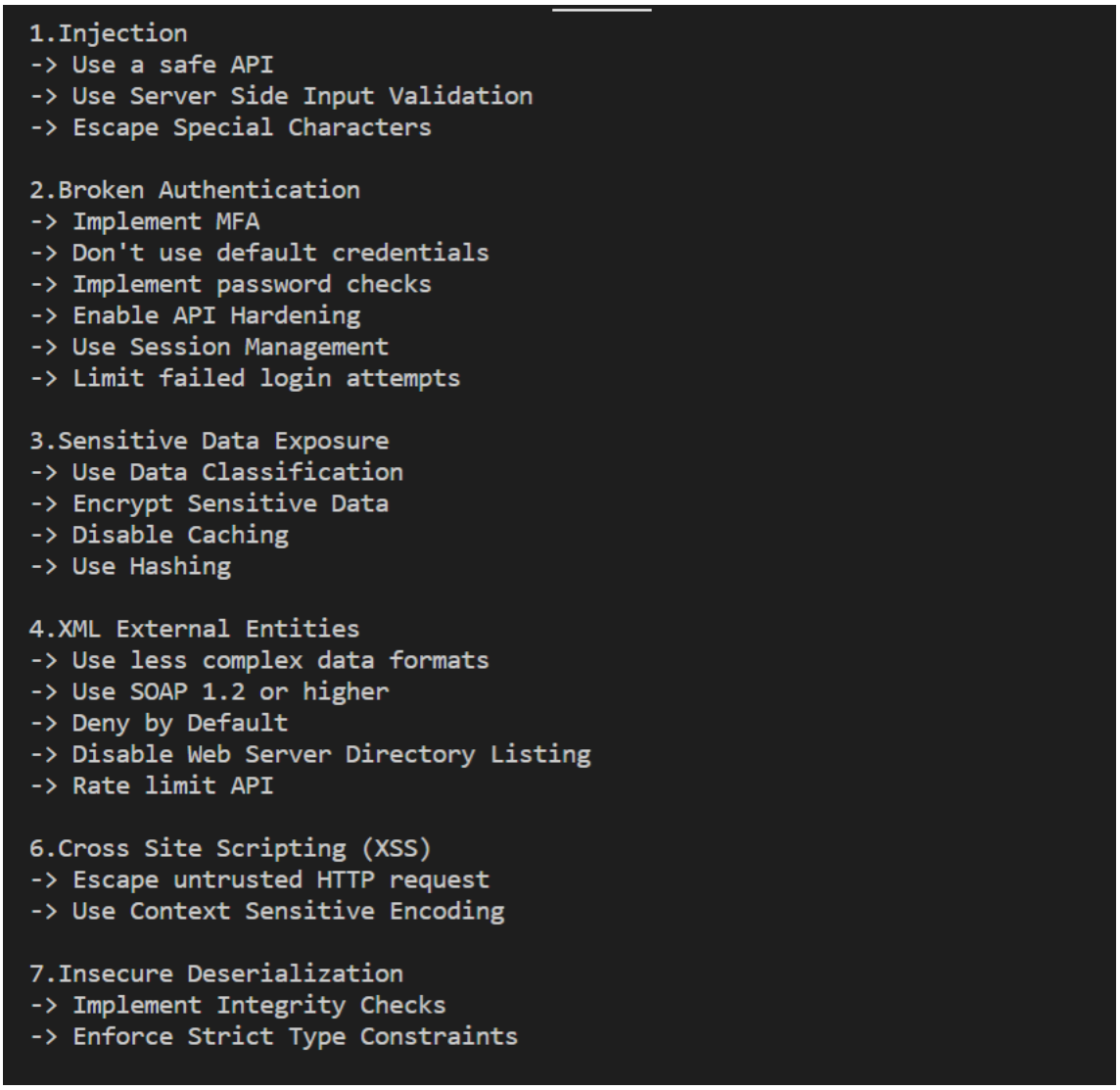

Figure 6: Output Screen 3 - Port scan result (cont..)

6.3. Result Analysis

As per the output screenshots attached above, the results of Portify were aligned with the initial project idea. Detailed analysis is offered below:

- **Port Range:** As the user enters the port range, it gives user the flexibility to focus on a range of port address and gain knowledge about the status of specific port addresses.
- **DNS resolution:** IP address is not always a convenient option, but URLs are more user friendly. Portify was able to successfully conduct DNS resolution of URLs entered by the user.
- **Port Scanner:** The results showed either open or closed. But in case any port is unreachable and inadequate information is received by the scanner, then it marked that port ‘closed’ as well. In order to give more detailed information to the user, Other statuses should also be incorporated, like filtered or unfiltered.
- **Vulnerability Mapping:** In order to provide a precise and user-friendly result, Portify was able to present critical vulnerabilities, but less critical vulnerabilities should also be displayed.

7. Conclusion and Future Scope

The conclusion is that information security is a growing industry and has a vast scope. Safeguarding the sensitive information of users from the people with wrong intent is the biggest challenge faced by the industries today. It is very crucial for the organizations to protect their information systems, network devices, check for open ports and should implement strong security controls to safeguard the sensitive data of users and employees.

We were successfully able to achieve our objective of making a vulnerability mapping tool which identifies vulnerable ports on a URL or target IP provided by the user basis the port scan along with the mitigation techniques associated with each vulnerability.

Following tasks were accomplished, basis the methodology used:

- DNS resolution,
- Port Scanning on a URL/IP address,
- Identification of vulnerabilities associated to open ports, and
- Mitigation techniques

In order to enhance Portify's functionality and accuracy, synchronization to the CVSS database could be done to showcase the vulnerabilities and their corresponding score in real time. This approach will provide the user with detailed information to make an informed decision about implementing the mitigation techniques.

References

[1] Aggarwal, K. and Singh, Y., 2008. Software Engineering. New Delhi: New Age International.

[2] Owasp.org. 2020. OWASP Top Ten. [online] Available at: <https://owasp.org/www-project-top-ten/> [Accessed 17 April 2020].

[3] Pentest-Tools.com. 2020. Online Port Scanner with Nmap - Discover Open TCP Ports | Pentest-Tools.Com. [online] Available at: <https://pentest-tools.com/network-vulnerability-scanning/tcp-port-scanner-online-nmap> [Accessed 17 April 2020].

[4] Wilkins, S., 2020. TCP/IP Ports and Protocols | TCP/IP Ports and Protocols | Pearson IT Certification. [online] Pearsonitcertification.com. Available at: <https://www.pearsonitcertification.com/articles/article.aspx?p=1868080> [Accessed 17 April 2020].

A APPENDIX I PROJECT CODE

```
import java.util.*;
import java.net.*;
import java.util.HashMap;
import java.util.Map;

public class portify
{
    public static void main(String args[])
    {
        System.out.println("\nWELCOME TO PORT BASED VULNERABILITY SCANNER");
        int start=0,end=0;
        Scanner sc= new Scanner(System.in);

        System.out.println("\n\nEnter the starting Port Address:");
            //Taking port range
        start = sc.nextInt();

        System.out.println("Enter the ending Port Address:");
        end = sc.nextInt();

        System.out.println("Enter the target ip address as w.x.y.z :");
       //Taking target

        String ip;
        ip = sc.next();

        /*InetAddress address = InetAddress.getByName("www.google.com");

        System.out.println(address.getHostAddress());*/

        HashMap<Integer, String> port = new HashMap<> ();
       //Hashmap for storing port number and its status

        //System.out.println("Debug == "+(end-start));

        System.out.println("\nStarting Port Scan on host: "+ ip );

        for(int i=start,j=0;i<=end;i++,j++)
                //Loop for all ports in the range
        {
            try {
                Socket soc = new Socket(ip,i);
        //establishing socket connection to check if port is open

                System.out.print("\b\b\b----"+((100/(end-
start))*j)+"%");      //for loader

                // System.out.println(i+" Open");
```

```
                port.put(i,"Open");
        //adding port status in map
                soc.close();

            } catch (Exception e) {

                System.out.print("");
                port.put(i,"Closed");
            }
            System.out.print("\b\b\b----"+((100/(end-
start)*j))+"%");          //for loader
            //System.out.println(i+" Closed");
        }


        System.out.println("\n\nCurrent Port Status for target "+ip+" is\n")
;
        System.out.println(port);
        //printing port status for all ports

        System.out.println("\nVulnerability Status: ");
        if(port.get(21)=="Open")
        {
            System.out.println("\nPort 21:\nService Name: FTP\nVulerabilitie
s: Broken Access Control, Sensitive Data Exposure");
        }
        if(port.get(22)=="Open")
        {
            System.out.println("\nPort 22:\nService Name: SSH\nVulerabilitie
s: Sensitive Data Exposure");
        }
        if(port.get(23)=="Open")
        {
            System.out.println("\nPort 23:\nService Name: Telnet\nVulerabili
ties: Broken Authentication, Sensitive Data Exposure");
        }
        if(port.get(25)=="Open")
        {
            System.out.println("\nPort 25:\nService Name: SMTP\nVulerabiliti
es: Broken Access Control");
        }
        if(port.get(53)=="Open")
        {
            System.out.println("\nPort 53:\nService Name: DNS\nVulerabilitie
s: Broken Authentication");
        }
        if(port.get(80)=="Open")
        {
            System.out.println("\nPort 80:\nService Name: HTTP\nVulerabiliti
es: Sensitive Data Exposure, Broken Access Control, Injection, XSS");
        }
        if(port.get(110)=="Open")
```

```
        {
            System.out.println("\nPort 110:\nService Name: POP3\nVulerabilit
ies: Security Misconfigurations, Sensitive Data Exposure");
        }
        if(port.get(135)=="Open")
        {
            System.out.println("\nPort 135:\nService Name: RPC\nVulerabiliti
es: Sensitive Data Exposure");
        }
        if(port.get(137)=="Open")
        {
            System.out.println("\nPort 137:\nService Name: NetBIOS\nVulerabi
lities: Broken Access Control, Sensitive Data Exposure");
        }
        if(port.get(443)=="Open")
        {
            System.out.println("\nPort 443:\nService Name: HTTPS\nVulerabili
ties: Broken Access control");
        }
        if(port.get(1433)=="Open")
        {
            System.out.println("\nPort 1433:\nService Name: SQL\nVulerabilit
ies: Injection, XSS");
        }


        if(port.get(21)!="Open"&&port.get(22)!="Open"&&port.get(23)!="Open"&
&port.get(25)!="Open"&&port.get(53)!="Open"&&port.get(80)!="Open"&&port.get(
110)!="Open"&&port.get(135)!="Open"&&port.get(443)!="Open"&&port.get(1433)!=
"Open"&&port.get(137)!="Open")
        {
            System.out.println("\nNo Vulnerabilities found on host "+ip);
        }
        else
        {
            String choice;

            System.out.println("Do you want to display the mitigation guidel
ines for Common Vulnerabilities (Y/N)?");
            choice = sc.next();

            if(choice.equals("Y")||choice.equals("y"))
            {
                System.out.println("\n1.Injection\n-> Use a safe API\n-
> Use Server Side Input Validation\n-> Escape Special Characters");
                System.out.println("\n2.Broken Authentication\n-
> Implement MFA\n-> Don't use default credentials\n-
> Implement password checks\n-> Enable API Hardening\n-
> Use Session Management\n-> Limit failed login attempts");
                System.out.println("\n3.Sensitive Data Exposure\n-
> Use Data Classification\n-> Encrypt Sensitive Data\n-> Disable Caching\n-
> Use Hashing");
```

```
                System.out.println("\n4.XML External Entities\n-
> Use less complex data formats\n-> Use SOAP 1.2 or higher\n-
> Disbale XEE processing\n-> Use server side Input Validation");
                System.out.println("\n5.Broken Access Control\n-
> Deny by Default\n-> Disable Web Server Directory Listing\n-
> Rate limit API");
                System.out.println("\n6.Cross Site Scripting (XSS)\n-
> Escape untrusted HTTP request\n-> Use Context Sensitive Encoding");
                System.out.println("\n7.Insecure Deserialization\n-
> Implement Integrity Checks\n-> Enforce Strict Type Constraints");
            }
            else
            {
                System.out.println("\nThank You for using our tool :)");
            }


        }
        System.out.println("\n");
    }
}
```