A Project Report-I

On

Cybersquad

Submitted in partial fulfilment of the requirement of
University of Mumbai for the Degree of

Bachelor of Engineering

In

CSE IOT and Cyber Security including Blockchain

Submitted By

Om Sanjay Kadam
Niraj Sunil Kadam
Ashish Santosh Pawar
Siddhant Sarjerao Dutal

Supervisor

Dr Madhu Nashipudimath

Department of CSE IOT and Cyber Security including Blockchain
Smt. Indira Gandhi College of Engineering, Ghansoli – 400701
UNIVERSITY OF MUMBAI

Academic Year 2024 – 25

Department of CSE IOT and Cyber Security including Blockchain
SMT. INDIRA GANDHI COLLEGE OF ENGINEERING
GHANSOLI – 400701

CERTIFICATE

This is to certify that the requirements for the Major Project-I entitled '**CYBERSQUAD**' have been successfully completed by the following students:

Name	**Reg No.**
Om Sanjay Kadam	2021ci30f
Niraj Sunil Kadam	2021ci29f
Ashish Santosh Pawar	2021ci49f
Siddhant Sarjerao Dutal	2021ci14f

in partial fulfilment of Bachelor of Technology of Mumbai University in the Department of CSE IOT and Cyber Security including Blockchain, Smt. Indira Gandhi College Of Engineering, Ghansoli – 400701 during the Academic Year 2024 – 2025 .

Supervisor
(Dr Madhu Nashipudimath)

Head of Department
(Dr. Madhu Nashipudimath)

Principal
(Dr. Sunil Chavan)

Department of CSE IOT and Cyber Security including Blockchain
SMT. INDIRA GANDHI COLLEGE OF ENGINEERING
GHANSOLI – 400701

PROJECT APPROVAL FOR B.E

This project entitled **“CYBERSQUAD”** by Om Sanjay Kadam,Niraj Sunil Kadam , Ashish Santosh Pawar Siddhant Sarjerao Dutal Name is approved for the degree of **CSE IOT and Cyber Security including Blockchain.**

Examiners:

1. ________________

2. ________________

Supervisors:

1. ________________

Date:

Place:

Department of CSE IOT and Cyber Security including Blockchain

SMT. INDIRA GANDHI COLLEGE OF ENGINEERING

GHANSOLI – 400701

DECLARATION

We declare that this written submission for the B.E project entitled **“CYBERSQUAD”** represents our ideas in our own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any ideas / data / fact / source in our submission. We understand that any violation of the above will cause disciplinary action by the institute and also evoke penal action from the sources which have not been properly cited or from whom prior permission have not been taken when needed.

Project Group Members:

Om Sanjay Kadam

Niraj Sunil Kadam

Ashish Santosh Pawar

Siddhant Sarjerao Dutal

Date:

Place

ACKNOWLEDGEMENTS

We extend our deepest appreciation to our esteemed guide and Head of Department , **Dr. Madhu Nashipudimath,** for her exceptional guidance, unwavering support, and constant encouragement. Her expertise has been instrumental in the development of our designed algorithms and has provided us with practical insights into the application of various theoretical frameworks. Her mentorship has been a cornerstone of our project's success.

Furthermore, we would like to acknowledge the coordinator **Prof Sarita Bopalkar** and faculty members for their valuable guidance and encouragement. Their dedication to our academic success has been a source of inspiration and has significantly contributed to the completion of our project.

We would like to express our profound gratitude to the Principal **Dr Sunil Chavan** for providing us with the invaluable opportunity to integrate the learning from this graduate course into our project work. The institution's support and encouragement have been pivotal in our academic journey.

Lastly, we would like to express our heartfelt thanks to the lab staff for their technical support and assistance, which were crucial in the practical execution of our project. Their expertise and readiness to help have been greatly appreciated.

Om Sanjay Kadam

Niraj Sunil Kadam

Ashish Santosh Pawar

Siddhant Sarjerao Dutal

Table of Contents

Abstract

Cybersquad is a Python-based cybersecurity tool designed to perform comprehensive vulnerability scans on systems by providing the IP address. This paper presents Cybersquad's design, implementation, and application, detailing its two scanning modes: a basic scan for summarizing open ports and directories, and an aggressive scan for identifying vulnerabilities, their severity, CVE IDs, and available exploits.

The basic scan quickly provides an overview of a system's exposure, while the aggressive scan offers in-depth vulnerability analysis. Cybersquad's development addresses the need for a user-friendly yet powerful tool for security professionals, penetration testers, and system administrators. Its architecture, functionality, and practical applications are discussed, along with a case study demonstrating its effectiveness in identifying known vulnerabilities.

This research highlights Cybersquad's capability to streamline vulnerability assessments, making advanced security analysis accessible and actionable. By automating vulnerability identification and reporting, Cybersquad helps users prioritize and mitigate security risks effectively. The paper concludes with an evaluation of the tool's performance, its limitations, and potential future enhancements to enhance its utility in cybersecurity.

Figures Table

Introduction

1.1 Fundamentals

In the modern digital landscape, cybersecurity has become a paramount concern for organizations and individuals alike. The rapid evolution of cyber threats, including malware, ransomware, and sophisticated attacks, necessitates the use of advanced tools to protect sensitive information and infrastructure [1]. Traditional cybersecurity tools often require extensive configuration, deep technical knowledge, and significant time investment, making them less accessible to a broader audience [2]. Recognizing this gap, we developed Cybersquad, a Python-based cybersecurity tool designed to simplify and streamline the process of performing comprehensive vulnerability scans [3].

1.2 Objectives

The primary objectives of this research paper are to:

- Introduce Cybersquad: Provide a detailed overview of the tool, including its purpose, design, and key features [4].
- Describe Implementation: Explain the technical implementation of Cybersquad, focusing on its two scanning modes: basic and aggressive [5].
- Demonstrate Effectiveness: Showcase the tool's capabilities through practical applications and case studies, highlighting its effectiveness in identifying vulnerabilities [6].
- Provide Recommendations: Offer actionable insights and recommendations based on scan results to help users improve their system security [7].

1.3 Scope

This paper encompasses the following aspects of Cybersquad:

- Design and Architecture: A comprehensive description of Cybersquad's architecture, including the modules for network scanning, vulnerability detection, and reporting [8].
- Scanning Modes: Detailed explanation of the basic and aggressive scanning modes, outlining their respective functionalities and use cases [9].
- Practical Applications: Real-world applications of Cybersquad in various cybersecurity scenarios, demonstrating its utility and effectiveness [10].
- Case Study: An in-depth case study showcasing Cybersquad's performance in a controlled environment with known vulnerabilities [11].
- Analysis and Evaluation: Critical analysis of the tool's performance, including its strengths, limitations, and areas for improvement [12].
- Future Enhancements: Discussion of potential future developments and enhancements to further improve Cybersquad's capabilities and effectiveness [13].

Literature Survey

2.1 Introduction

The increasing complexity of cyber threats necessitates effective vulnerability scanning tools that can help identify and mitigate security risks. This literature survey focuses on Vulnerability Scanning Tools which are instrumental in basic network and directory scanning, respectively. Additionally, we explore the limitations of these tools and how our tool, Cybersquad, addresses these shortcomings by integrating more advanced features through custom scripts and utilizing vulnerability data from the National Vulnerability Database (NVD) [1][2][6]

2.2 Literature Review

➔ Nmap (Network Mapper):

Overview: Nmap is a powerful open-source tool for network exploration and security auditing. It is primarily used for discovering hosts and services on a computer network by sending packets and analyzing the responses [1].

➔ DirB (Directory Buster):

Overview: DirB is a simple command-line tool used for web directory brute-forcing. It helps identify hidden directories and files on web servers by making multiple requests based on a wordlist [7].

➔ Custom Scripts Using NVD Data:

Overview: To enhance vulnerability detection beyond the capabilities of Nmap and DirB, custom scripts have been developed leveraging data from the National Vulnerability Database (NVD). These scripts perform deeper scans and analyze vulnerabilities associated with identified services and applications [6].

2.3 Summary of Literature Review

The literature review encompasses several papers that explore various aspects of vulnerability scanning and the limitations of existing tools. The review begins with "Port Scanning Techniques Tools and Detection" by Coyle [1], which provides a comprehensive overview of port scanning but is criticized for lacking detail and subtlety. The 2018 publication "Vulnerability Scanning. CompTIA® PenTest+" by Abbas Moallem et al. [2] discusses the challenges of current vulnerability scanning tools, such as information overload and the inability to provide actionable solutions.

Suliman Alazmi's 2022 paper, "Effectiveness of Web Application Vulnerability Scanners" [3], highlights the significant variation in detection rates among different scanners, indicating a lack of consistency in their effectiveness. J. Fonseca et al.'s 2017 paper, "Testing and Comparing Web Vulnerability Scanning Tools for SQL Injection and XSS Attacks" [4], reveals that these tools often produce unreliable results for detecting common web vulnerabilities.

Finally, the 2024 paper "Survey on detecting and preventing web application broken access control attacks" by Ahmed Anas, Salwa Elgamal, and B. Youssef [5] provides a broad survey of the current research on detecting access control vulnerabilities but notes limitations in the effectiveness of these approaches.

Overall, the literature review underscores the need for a more comprehensive and effective vulnerability scanning tool that addresses the limitations identified in existing tools. Cybersquad aims to fill this gap by offering a tool with both basic and aggressive scanning modes, extensive port coverage, and detailed vulnerability reporting, including CVE IDs and exploit suggestions.

Paper Name	Description	Publish Year	Limitation	Author(s)
Port Scanning Techniques Tools and Detection	This review aims to consolidate varied information regarding Port Scanning, and examine the tools, techniques, and detection algorithms used.	2024	Lack of Detail, losing subtlety in the process.	**Coyle, S.**
Vulnerability Scanning. CompTIA® PenTest+	The correction reports generated from the tools typically cause important info overload whereas failing to produce unjust solutions	2018	Current vulnerability scanning tools can cause info overload and produce unjust solutions	Abbas Moallem et al.
Effectiveness of Web Application Vulnerability Scanners	Web applications have been a significant target for successful security breaches in the last few years.	2022	Highly dissimilar detection rates between evaluations.	Suliman Alazmi D. Leon
Testing and Comparing Web Vulnerability Scanning Tools for SQL Injection and XSS Attacks	Automatic web vulnerability scanners can help to locate these vulnerabilities and are popular tools among developers of web applications.	2017	Results unreliable for detecting SQL injection and XSS attacks in web applications.	J. Fonseca et al.
Survey on detecting and preventing web application broken access control attacks	This paper gives a broad survey of the current research progress on approaches used to detect access control vulnerabilities exploitations and attacks in web application components.	2024	Limitations in detecting access control vulnerabilities, exploitations, and attacks,	Ahmed Anas, Salwa Elgamal, B. Youssef

Design

The design of Cybersquad involved several key steps to ensure that the tool would effectively address the limitations identified in the literature review. These steps included:

3.1.1 Existing System Architecture

The existing system architecture for cybersecurity vulnerability scanning typically involves using separate, standalone tools for different aspects of security assessments. For example, Nmap might be used for scanning open ports, while tools like DirB are used for identifying publicly accessible directories. Vulnerability databases, such as the National Vulnerability Database (NVD) or Vulners, are accessed manually or through separate scripts to identify vulnerabilities and associated details.

This approach is often fragmented, requiring security professionals to manually integrate results from various tools and databases.

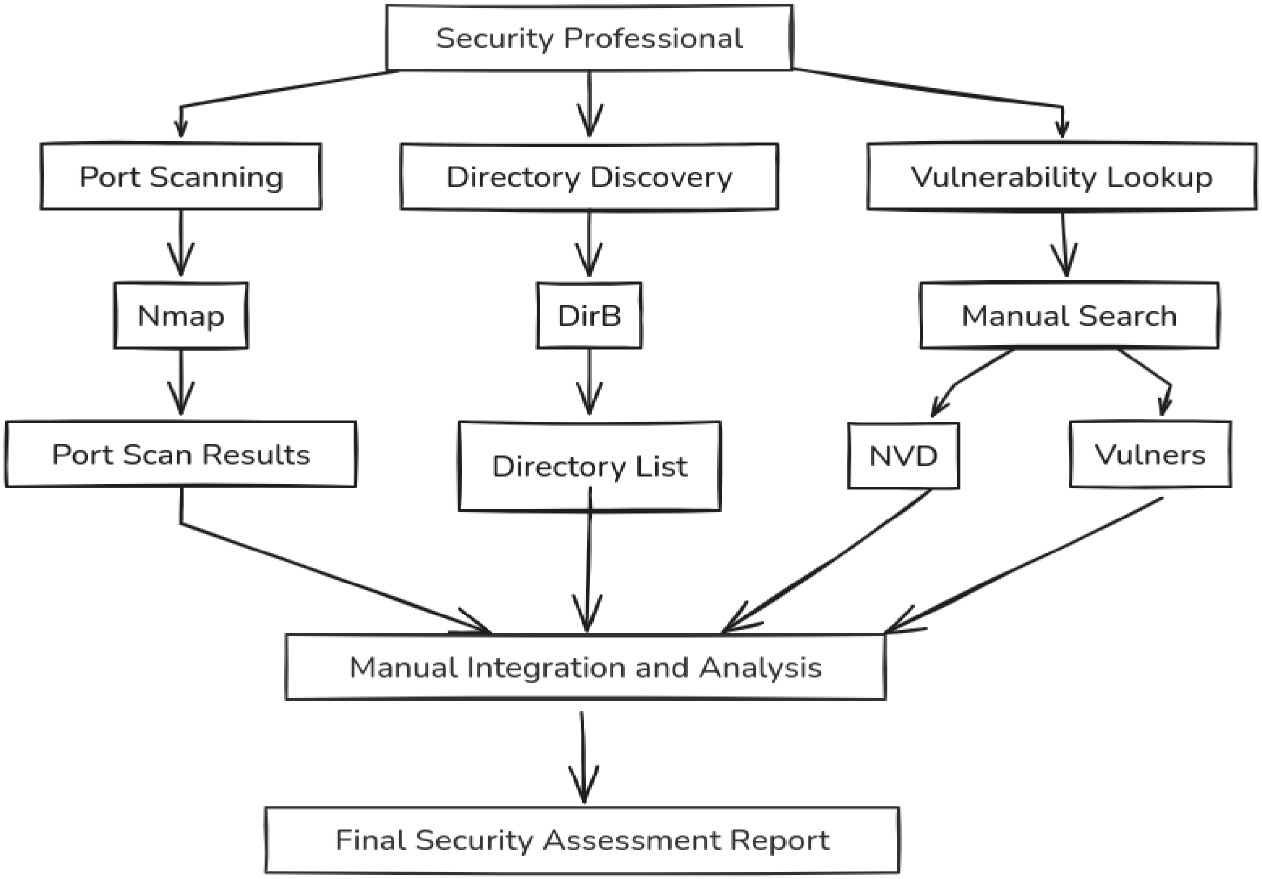


Fig No : 1 Existing System Architecture

3.1.2 Proposed System Architecture

Cybersquad addresses the limitations of the existing system by integrating multiple functionalities into a single, cohesive tool. The proposed system architecture is designed to streamline the vulnerability assessment process, making it more efficient and user-friendly [1].

The architecture of Cybersquad consists of three main modules: the Scanner Module, the Analysis Module, and the Reporting Module.

1. Scanner Module

The Scanner Module is responsible for managing the scanning process. It interfaces with Nmap for identifying open ports and with DirB for discovering publicly accessible directories and files. Additionally, it integrates with the Vulners API to retrieve detailed information about vulnerabilities, including severity levels, CVE IDs, and available exploits. This integration allows for comprehensive scans that cover both basic exposure and in-depth vulnerability analysis [2].

2. Analysis Module

The Analysis Module processes the scan results, identifying and categorizing vulnerabilities based on their severity. This module ensures that the data is accurate and actionable, providing users with a clear understanding of the potential risks. By automating the analysis process, Cybersquad reduces the likelihood of human error and ensures consistent results across different scans [3].

3. Reporting Module

The Reporting Module generates detailed reports that summarize the findings of the scans. These reports include an overview of the system's exposure, detailed vulnerability analysis, and recommended mitigation steps. The reports are designed to be easy to understand, even for users who may not have a deep technical background. This module enhances the usability of Cybersquad, making it accessible to a broader range of users, including system administrators and less experienced security professionals [4].

Overall, the proposed system architecture of Cybersquad offers a unified interface that integrates multiple tools and databases into a single platform. This integration not only simplifies the vulnerability assessment process but also enhances the accuracy and consistency of the results. By automating key tasks and providing comprehensive reports, Cybersquad helps users prioritize and mitigate security risks more effectively, addressing the key limitations of the existing system architecture [5].

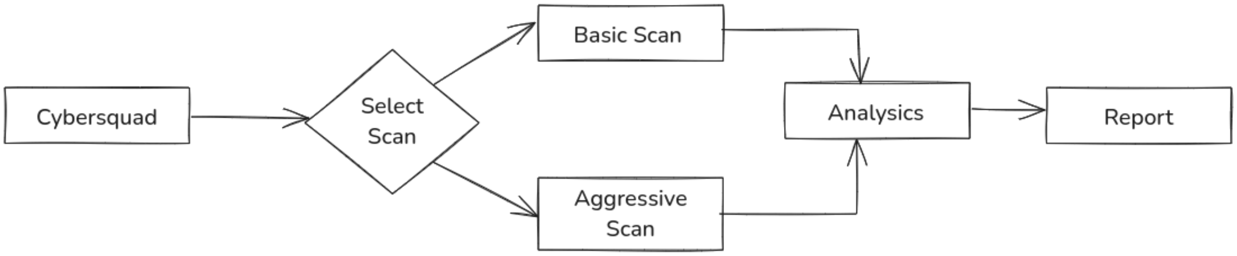


Fig No : 2 Proposed System Architecture

3.2 Methodology

3.2.1 Existing Methodology

The traditional methodology for conducting cybersecurity vulnerability assessments typically involves several disjointed steps, often requiring manual intervention and significant expertise. Here's a breakdown of the common approach:

- Tool Selection and Configuration

Security professionals choose appropriate tools for different aspects of the assessment. For instance, Nmap is used for network scanning to identify open ports, while DirB or similar tools are employed to find accessible directories and files. Vulnerability databases such as the National Vulnerability Database (NVD) or Vulners are consulted separately for detailed information on known vulnerabilities [1].

- Manual Scanning

Each selected tool is run individually. Nmap scans provide a list of open ports, while DirB scans reveal accessible directories. These scans need to be configured manually, and results are collected separately [2].

- Data Aggregation

Results from the various tools are manually aggregated. Security professionals must compile the data from Nmap, DirB, and any other tools used, integrating this information into a single report [3].

- Vulnerability Research

Once the basic data is collected, the next step involves researching identified vulnerabilities. This typically means manually querying vulnerability databases or using scripts to pull relevant CVE (Common Vulnerabilities and Exposures) information [4].

- Analysis and Reporting

The final step involves analyzing the collected data to determine the severity of vulnerabilities and potential impacts. This analysis is then documented in a report, which includes findings, severity levels, CVE IDs, and recommendations for mitigation. This process is often manual and requires considerable expertise to interpret the results accurately and comprehensively [5].

This existing methodology is labor-intensive, prone to errors, and inconsistent due to its manual nature and reliance on multiple, separate tools. It also requires a high level of expertise to ensure that all relevant vulnerabilities are identified and appropriately prioritized [6].

3.2.2 Proposed Methodology

Cybersquad's methodology aims to streamline and automate the vulnerability assessment process, providing a comprehensive, accurate, and user-friendly approach [1]. Here's how it works:

➔ Unified Tool Integration

Cybersquad integrates Nmap, DirB, and the Vulners API into a single platform. This integration eliminates the need for multiple standalone tools, reducing complexity and potential errors associated with manual configuration and data collection [2].

➔ Automated Scanning

- **Basic Scan:** Initiated through Cybersquad's command-line interface, the basic scan uses Nmap to identify open ports and DirB to find accessible directories and files. This scan quickly provides an overview of a system's exposure [3].
- **Aggressive Scan:** This scan delves deeper by interfacing with the Vulners API. It not only identifies vulnerabilities but also provides detailed information, including severity, CVE IDs, and potential exploits [4].

➔ Data Aggregation and Analysis

- **Scanner Module:** Automatically aggregates data from the various scanning tools.
- **Analysis Module:** Processes this data, identifying and categorizing vulnerabilities based on severity. This automated analysis ensures consistency and accuracy, reducing the potential for human error [5].

➔ Automated Reporting

The Reporting Module generates comprehensive reports that summarize the scan findings. These reports include:

- An overview of open ports and directories.
- Detailed vulnerability analysis with severity levels, CVE IDs, and potential exploits.[6]

➔ User-Friendly Interface

Cybersquad’s command-line interface simplifies the process for users, allowing them to specify the target IP address and choose between basic and aggressive scan modes. This approach makes advanced vulnerability assessments accessible to users with varying levels of expertise [7].

➔ Continuous Improvement

Cybersquad’s development roadmap includes enhancements such as improved algorithms to reduce false positives, performance optimizations, the development of a graphical user interface (GUI), and expanded integration with other security tools and databases [8].

By automating key tasks and providing a unified platform for vulnerability assessment, Cybersquad’s methodology addresses the limitations of the existing approach. It reduces manual effort, improves accuracy, and ensures that comprehensive, actionable insights are readily available, enabling users to effectively prioritize and mitigate security risks [9].

3.3 Requirements for Implementations

3.3.1 Algorithms / Techniques

Cybersquad employs a variety of algorithms and techniques to conduct comprehensive vulnerability assessments. The key algorithms and techniques used are:

1. Nmap for Network Scanning: Nmap is a powerful network scanning tool used to identify open ports and services running on a target system. It employs various scanning techniques, such as TCP SYN scan, TCP Connect scan, and UDP scan, to determine the state of network ports.

2. DirB for Directory Enumeration: DirB is a web content scanner that brute-forces directories and files on web servers. It uses wordlists to identify accessible directories and files that may not be publicly listed.

3. Vulners API Integration: The Vulners API provides access to a comprehensive database of vulnerabilities, including details on CVE IDs, severity levels, and available exploits. Cybersquad interfaces with this API to retrieve detailed vulnerability information based on the scan results.

4. Data Aggregation and Analysis: The Analysis Module in Cybersquad processes the aggregated data from Nmap, DirB, and the Vulners API. It uses classification algorithms to categorize vulnerabilities based on their severity and potential impact.

5. Automated Reporting: The Reporting Module generates detailed reports by applying templating techniques to compile the scan results, analysis, and recommendations into a structured format.

3.3.2 Use Case Diagram / Activity Diagram

Use Case Diagram:

The use case diagram outlines the primary interactions between users and the Cybersquad system.

Actors: Security Professional, System Administrator

Use Cases:

- Initiate Basic Scan
- Initiate Aggressive Scan
- View Scan Results
- Generate Report

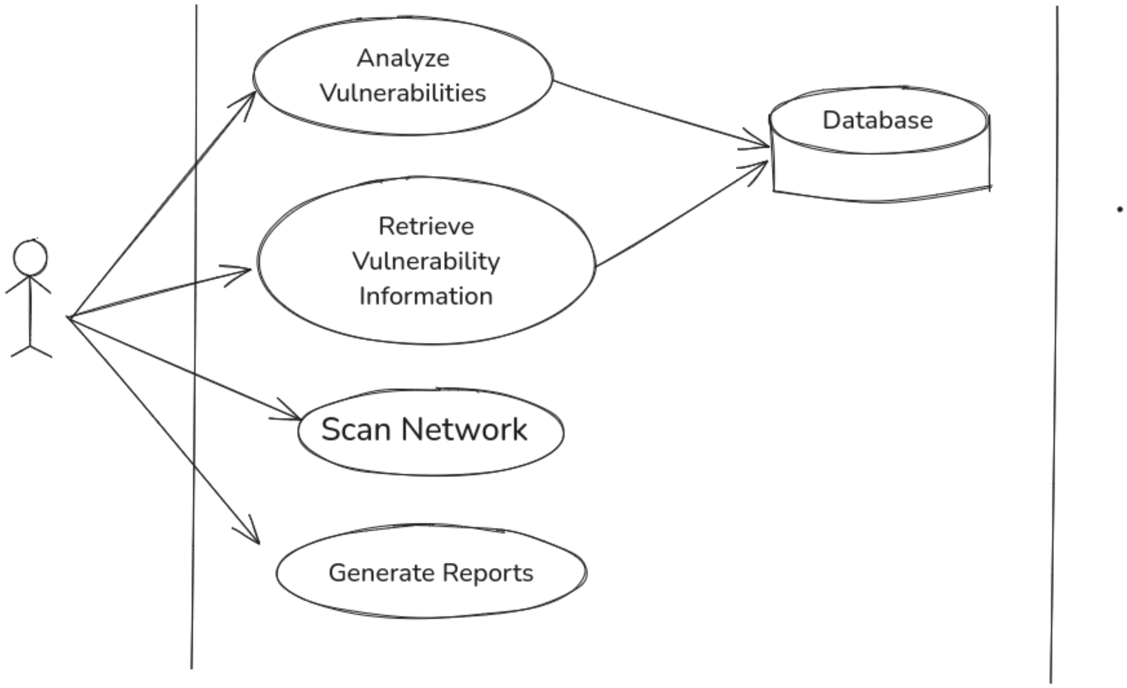


Fig No : 3 Use Case Diagram

Activity Diagram:

The activity diagram shows the flow of actions involved in performing a vulnerability scan using Cybersquad.

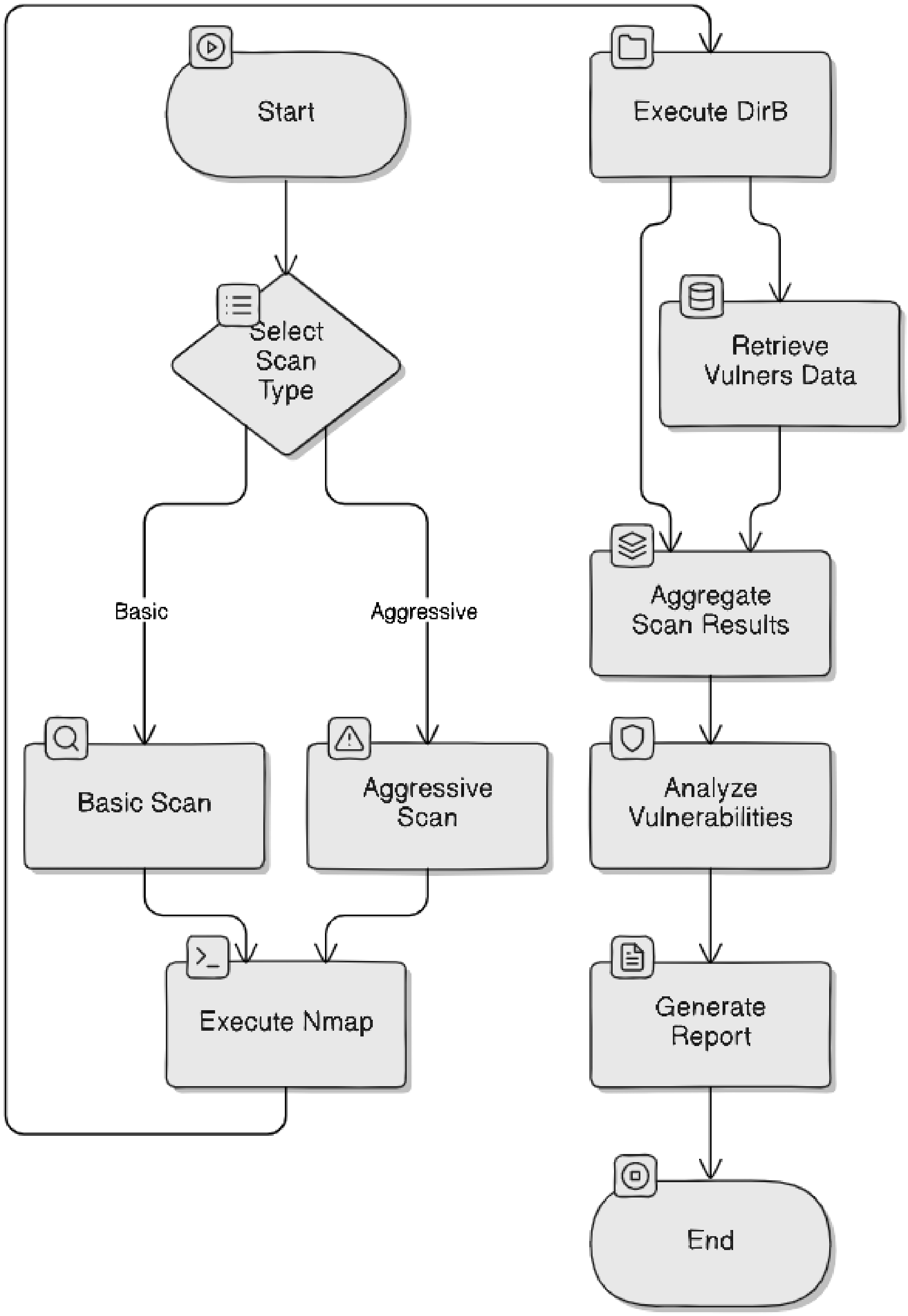


Fig No : 4 Activity Case Diagram

Sample Vulnerability Scanning Report

4.1.1 Introduction

Title: Vulnerability Scanning of the "Planet Earth" Machine Using Cybersquad

In this report, we detail the vulnerability scanning process conducted on the "Planet Earth" machine utilizing Cybersquad, a Python-based cybersecurity tool designed for comprehensive vulnerability assessments. The primary objective of this scan was to identify potential security vulnerabilities within the system, evaluate their severity, and propose remediation measures.

4.1.2 Objective

The key objectives of this vulnerability scanning exercise included:

- Identifying open ports and services running on the "Planet Earth" machine.
- Assessing the security posture of the machine by discovering known vulnerabilities.
- Providing detailed information on identified vulnerabilities, including their CVE (Common Vulnerabilities and Exposures) identifiers, severity ratings, and potential exploits.
- Offering recommendations for mitigating the identified vulnerabilities.

4.1.3 Methodology

➔ **Preparation:**

- Defined the scope of the scan by specifying the IP address of the "Planet Earth" machine.
- Ensured that all necessary permissions were obtained to conduct the scan in compliance with ethical hacking guidelines.

➔ **Scanning Modes:**

1. Basic Scan:

Conducted a basic scan to identify open ports and services.

Generated a summary report detailing the services detected and their respective versions.

2. Aggressive Scan:

Performed an aggressive scan to delve deeper into the security assessment.

Identified vulnerabilities associated with the detected services, providing information on severity levels and CVE IDs.

➔ **Data Analysis:**

Analyzed the results from both scans to determine the overall security posture of the "Planet Earth" machine.

Compiled findings into a comprehensive report format for easy interpretation.

4.1.4 Findings

The scans revealed several critical and high-severity vulnerabilities on the "Planet Earth" machine. Below are the key findings:

Extracted CPEs

- OpenBSD OpenSSH:
  - ➔ CPE: cpe:/a:openbsd:openssh:8.6:
  - ➔ Details: OpenSSH version 8.6 has been identified. It is essential to check for any known vulnerabilities associated with this version.

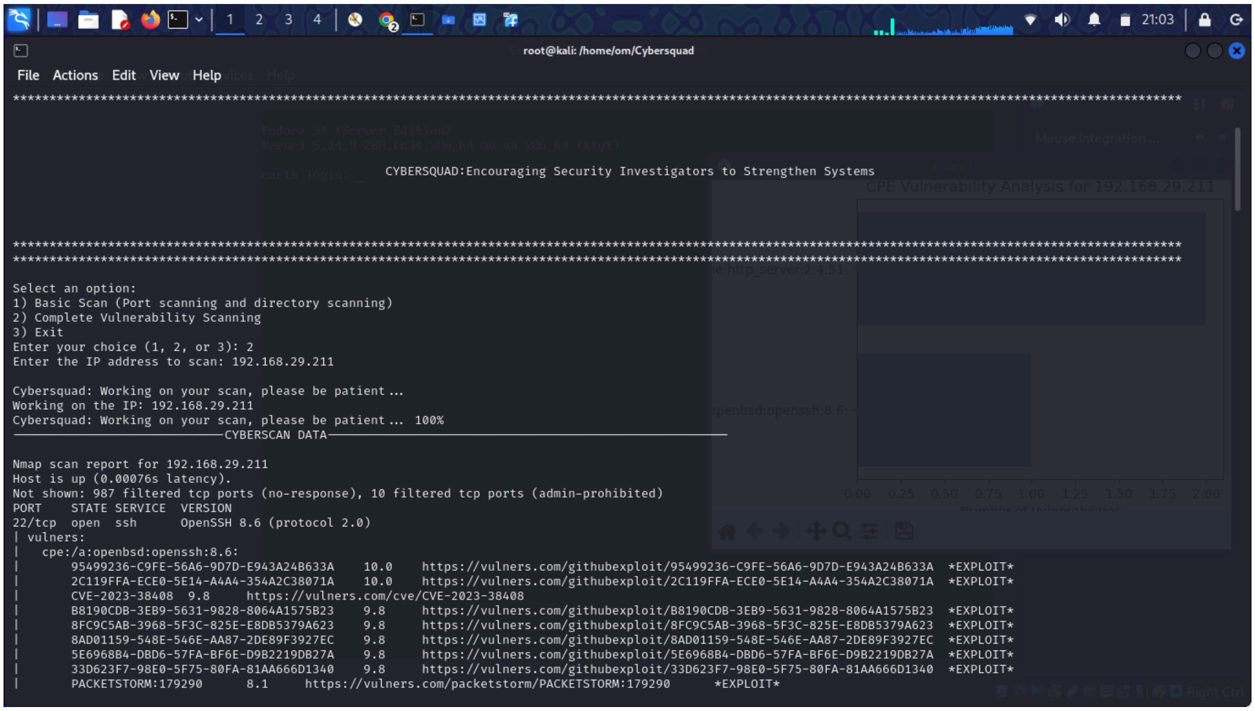
- Apache HTTP Server:
  - ➔ CPE: cpe:/a:apache:http_server:2.4.51:
  - ➔ Details: Apache HTTP Server version 2.4.51 is detected. This version has known vulnerabilities that should be assessed and remediated.

Vulnerability ID	**Severity**	**Description**
CVE-2023-38408	9.8	Critical vulnerability allowing remote code execution
95499236-C9FE-56A6-9D7D-E943A24B633A	10.0	Potential remote code execution
2C119FFA-ECE0-5E14-A4A4-354A2C38071A	10.0	Vulnerability in authentication handling
B8190CDB-3EB9-5631-9828-8064A1575B23	9.8	Denial of service
8FC9C5AB-3968-5F3C-825E-E8DB5379A623	9.8	Code execution risk

Vulnerability Assessment Summary

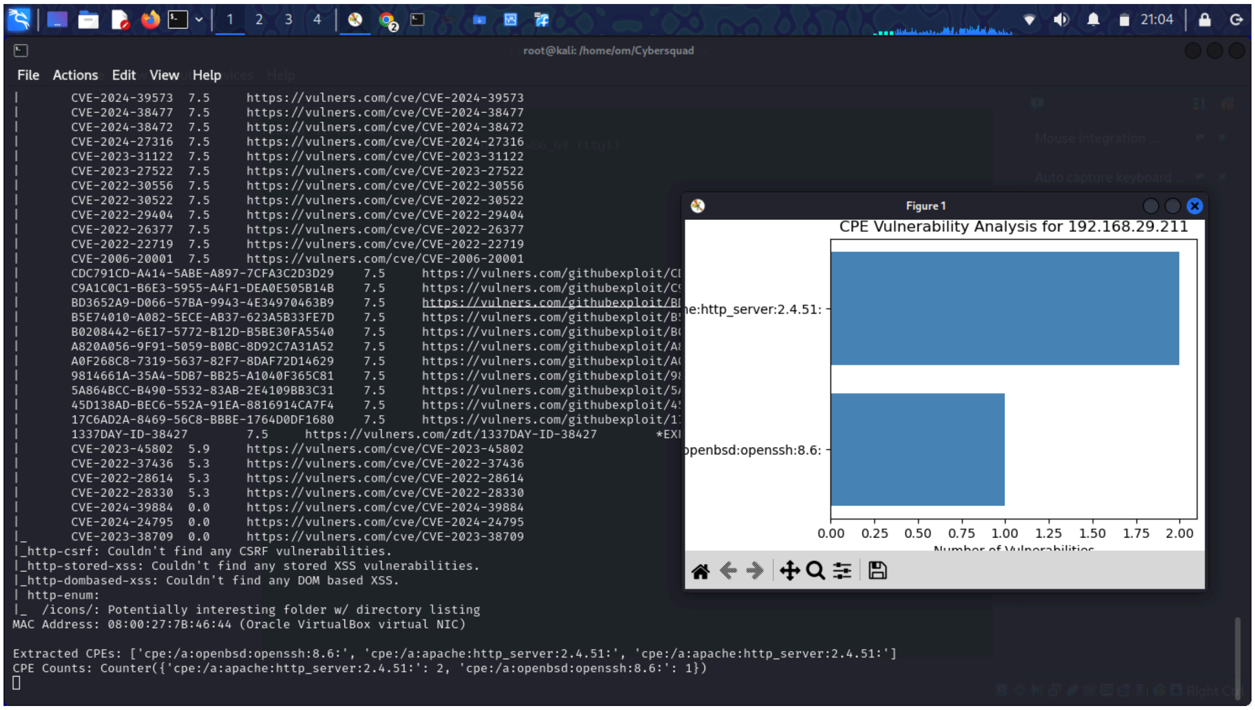
The vulnerabilities listed above indicate a critical security risk, particularly the presence of high-severity CVEs that could allow unauthorized access and potential remote code execution. The presence of multiple exploit links highlights the urgency to address these vulnerabilities promptly.

- Update OpenSSH to the latest stable version to address known vulnerabilities, including CVE-2023-38408
- Review and strengthen the SSH configuration settings to minimize attack vectors, such as disabling root login and using key-based authentication.



```
*****************************************************************************************************************************************************************

                                                CYBERSQUAD:Encouraging Security Investigators to Strengthen Systems


*****************************************************************************************************************************************************************
*****************************************************************************************************************************************************************

Select an option:
1) Basic Scan (Port scanning and directory scanning)
2) Complete Vulnerability Scanning
3) Exit
Enter your choice (1, 2, or 3): 2
Enter the IP address to scan: 192.168.29.211

Cybersquad: Working on your scan, please be patient ...
Working on the IP: 192.168.29.211
Cybersquad: Working on your scan, please be patient ...  100%
────────────────────────────CYBERSCAN DATA──────────────────────────────────────────────

Nmap scan report for 192.168.29.211
Host is up (0.00076s latency).
Not shown: 987 filtered tcp ports (no-response), 10 filtered tcp ports (admin-prohibited)
PORT    STATE SERVICE  VERSION
22/tcp  open  ssh      OpenSSH 8.6 (protocol 2.0)
| vulners:
|   cpe:/a:openbsd:openssh:8.6:
|       95499236-C9FE-56A6-9D7D-E943A24B633A    10.0    https://vulners.com/githubexploit/95499236-C9FE-56A6-9D7D-E943A24B633A  *EXPLOIT*
|       2C119FFA-ECE0-5E14-A4A4-354A2C38071A    10.0    https://vulners.com/githubexploit/2C119FFA-ECE0-5E14-A4A4-354A2C38071A  *EXPLOIT*
|       CVE-2023-38408  9.8     https://vulners.com/cve/CVE-2023-38408
|       B8190CDB-3EB9-5631-9828-8064A1575B23    9.8     https://vulners.com/githubexploit/B8190CDB-3EB9-5631-9828-8064A1575B23  *EXPLOIT*
|       8FC9C5AB-3968-5F3C-825E-E8DB5379A623    9.8     https://vulners.com/githubexploit/8FC9C5AB-3968-5F3C-825E-E8DB5379A623  *EXPLOIT*
|       8AD01159-548E-546E-AA87-2DE89F3927EC    9.8     https://vulners.com/githubexploit/8AD01159-548E-546E-AA87-2DE89F3927EC  *EXPLOIT*
|       5E6968B4-DBD6-57FA-BF6E-D9B2219DB27A    9.8     https://vulners.com/githubexploit/5E6968B4-DBD6-57FA-BF6E-D9B2219DB27A  *EXPLOIT*
|       33D623F7-98E0-5F75-80FA-81AA666D1340    9.8     https://vulners.com/githubexploit/33D623F7-98E0-5F75-80FA-81AA666D1340  *EXPLOIT*
|       PACKETSTORM:179290      8.1     https://vulners.com/packetstorm/PACKETSTORM:179290      *EXPLOIT*
```

Fig No : 5 Basic Scan Output



```
|       CVE-2024-39573  7.5     https://vulners.com/cve/CVE-2024-39573
|       CVE-2024-38477  7.5     https://vulners.com/cve/CVE-2024-38477
|       CVE-2024-38472  7.5     https://vulners.com/cve/CVE-2024-38472
|       CVE-2024-27316  7.5     https://vulners.com/cve/CVE-2024-27316
|       CVE-2023-31122  7.5     https://vulners.com/cve/CVE-2023-31122
|       CVE-2023-27522  7.5     https://vulners.com/cve/CVE-2023-27522
|       CVE-2022-30556  7.5     https://vulners.com/cve/CVE-2022-30556
|       CVE-2022-30522  7.5     https://vulners.com/cve/CVE-2022-30522
|       CVE-2022-29404  7.5     https://vulners.com/cve/CVE-2022-29404
|       CVE-2022-26377  7.5     https://vulners.com/cve/CVE-2022-26377
|       CVE-2022-22719  7.5     https://vulners.com/cve/CVE-2022-22719
|       CVE-2006-20001  7.5     https://vulners.com/cve/CVE-2006-20001
|       CDC791CD-A414-5ABE-A897-7CFA3C2D3D29    7.5     https://vulners.com/githubexploit/C
|       C9A1C0C1-B6E3-5955-A4F1-DEA0E505B14B    7.5     https://vulners.com/githubexploit/C
|       BD3652A9-D066-57BA-9943-4E34970463B9    7.5     https://vulners.com/githubexploit/B
|       B5E74010-A082-5ECE-AB37-623A5B33FE7D    7.5     https://vulners.com/githubexploit/B
|       B0208442-6E17-5772-B12D-B5BE30FA5540    7.5     https://vulners.com/githubexploit/B
|       A820A056-9F91-5059-B0BC-8D92C7A31A52    7.5     https://vulners.com/githubexploit/A
|       A0F268C8-7319-5637-82F7-8DAF72D14629    7.5     https://vulners.com/githubexploit/A
|       9814661A-35A4-5DB7-BB25-A1040F365C81    7.5     https://vulners.com/githubexploit/9
|       5A864BCC-B490-5532-83AB-2E4109BB3C31    7.5     https://vulners.com/githubexploit/5
|       45D138AD-BEC6-552A-91EA-8816914CA7F4    7.5     https://vulners.com/githubexploit/4
|       17C6AD2A-8469-56C8-BBBE-1764D0DF1680    7.5     https://vulners.com/githubexploit/1
|       1337DAY-ID-38427        7.5     https://vulners.com/zdt/1337DAY-ID-38427        *EX
|       CVE-2023-45802  5.9     https://vulners.com/cve/CVE-2023-45802
|       CVE-2022-37436  5.3     https://vulners.com/cve/CVE-2022-37436
|       CVE-2022-28614  5.3     https://vulners.com/cve/CVE-2022-28614
|       CVE-2022-28330  5.3     https://vulners.com/cve/CVE-2022-28330
|       CVE-2024-39884  0.0     https://vulners.com/cve/CVE-2024-39884
|       CVE-2024-24795  0.0     https://vulners.com/cve/CVE-2024-24795
|_      CVE-2023-38709  0.0     https://vulners.com/cve/CVE-2023-38709
|_http-csrf: Couldn't find any CSRF vulnerabilities.
|_http-stored-xss: Couldn't find any stored XSS vulnerabilities.
|_http-dombased-xss: Couldn't find any DOM based XSS.
| http-enum:
|_  /icons/: Potentially interesting folder w/ directory listing
MAC Address: 08:00:27:7B:46:44 (Oracle VirtualBox virtual NIC)

Extracted CPEs: ['cpe:/a:openbsd:openssh:8.6:', 'cpe:/a:apache:http_server:2.4.51:', 'cpe:/a:apache:http_server:2.4.51:']
CPE Counts: Counter({'cpe:/a:apache:http_server:2.4.51:': 2, 'cpe:/a:openbsd:openssh:8.6:': 1})
```

Fig No : 6 Aggressive Scan Output

Conclusion

The Cybersquad tool represents a significant advancement in vulnerability assessment within the cybersecurity landscape. Designed to cater to the needs of security professionals, penetration testers, and system administrators, Cybersquad offers a comprehensive solution for identifying and mitigating vulnerabilities in networked systems [1].Throughout the development and implementation of Cybersquad, key considerations were made to ensure its effectiveness, usability, and adaptability. The dual scanning modes—basic and aggressive—enable users to tailor their assessments according to the specific requirements of their environment. The basic scan provides a rapid overview of system exposure, while the aggressive scan delivers in-depth analysis, including vulnerability severity ratings and associated Common Vulnerabilities and Exposures (CVEs) [2].

While Cybersquad offers many advantages, it has also addressed several key drawbacks commonly found in existing vulnerability scanning tools:

- Limited Information: Traditional tools often provide superficial insights into vulnerabilities. Cybersquad enhances the depth of information available, offering detailed descriptions and severity ratings that assist users in understanding the implications of each vulnerability [3].
- Non-availability of Exploits: Many tools lack access to current exploit databases, limiting their effectiveness in real-world scenarios. Cybersquad integrates with external resources, providing users with links to relevant exploits and remediation steps, thereby facilitating faster and more effective mitigation [4].
- Cost: The financial barrier associated with many vulnerability scanning tools can deter organizations from conducting regular assessments. Cybersquad, being an open-source tool, reduces this barrier, making it accessible to a broader range of users [5].

REFERENCES

[1] Coyle, S. (2024). Port Scanning Techniques Tools and Detection. Preprints 2024, 2024030225. https://doi.org/10.20944/preprints202403.0225.v1

[2] Abbas Moallem et al. (2018). Vulnerability Scanning. CompTIA® PenTest+.

[3] Alazmi, S., & Leon, D. (2022). A Systematic Literature Review on the Characteristics and Effectiveness of Web Application Vulnerability Scanners. IEEE Access, PP, 1-1. https://doi.org/10.1109/ACCESS.2022.3161522

[4] Fonseca, J., Vieira, M., & Madeira, H. (2007). Testing and Comparing Web Vulnerability Scanning Tools for SQL Injection and XSS Attacks. 13th Pacific Rim International Symposium on Dependable Computing (PRDC 2007), 365-372. https://doi.org/10.1109/PRDC.2007.55.

[5] Anas, A., Elgamal, S., & Youssef, B. (2024). Survey on detecting and preventing web application broken access control attacks. International Journal of Electrical and Computer Engineering (IJECE). https://doi.org/10.11591/ijece.v14i1.pp772-781.

[6] National Institute of Standards and Technology. (2024). National Vulnerability Database (NVD). NIST

[7] OWASP. (2024). DirBuster - Directory Brute Forcing Tool. OWASP Foundation.

[8] Scandariato, R., Walden, J., Huygens, C., & Joosen, W. (2014). Predicting Vulnerable Software Components via Text Mining. IEEE Transactions on Software Engineering, 40(10), 993-1006. https://doi.org/10.1109/TSE.2014.2329899

[9] Chowdhury, M. M. U., & Zulkernine, M. (2010). Can Pre-release Testing Effectively Predict Field Failures? An Empirical Study of Open Source Projects. IEEE Transactions on Software Engineering, 36(4), 571-584. https://doi.org/10.1109/TSE.2009.89

[10] Bishop, M., & Bailey, M. (2008). A Critical Analysis of Vulnerability Taxonomies. IEEE Security & Privacy, 6(2), 80-83. https://doi.org/10.1109/MSP.2008.28

[11] Williams, L., & Williams, L. (2010). Internet Security: A Clear and Comprehensive Guide. Pearson Education.

[12] Howard, M., & LeBlanc, D. (2003). Writing Secure Code (2nd ed.). Microsoft Press.

[13] Arkin, B., Stender, S., & McGraw, G. (2010). Software Penetration Testing. IEEE Security & Privacy, 8(1), 84-87. https://doi.org/10.1109/MSP.2010.18

[14] McQueen, M. A., Boyer, W. F., Flynn, M. A., & Beitel, G. A. (2009). Time-to-compromise Model for Cybersecurity Risk Assessment. In Proceedings of the 2009 IEEE International Conference on Technologies for Homeland Security (pp. 125-130). https://doi.org/10.1109/THS.2009.5358555.