Automated Vulnerabilitiy Detection Program

Author: Mr.Tanakorn Narinsuvan

Advisor: Asst.Prof.Dr. Suphakit Awiphan

This independent study presents an **Automated Vulnerability Detection Program**, which was developed to assist in the automated assessment and detection of security vulnerabilities in networks and websites. The program integrates three primary security tools: **Nmap, Nikto, and OWASP ZAP**.

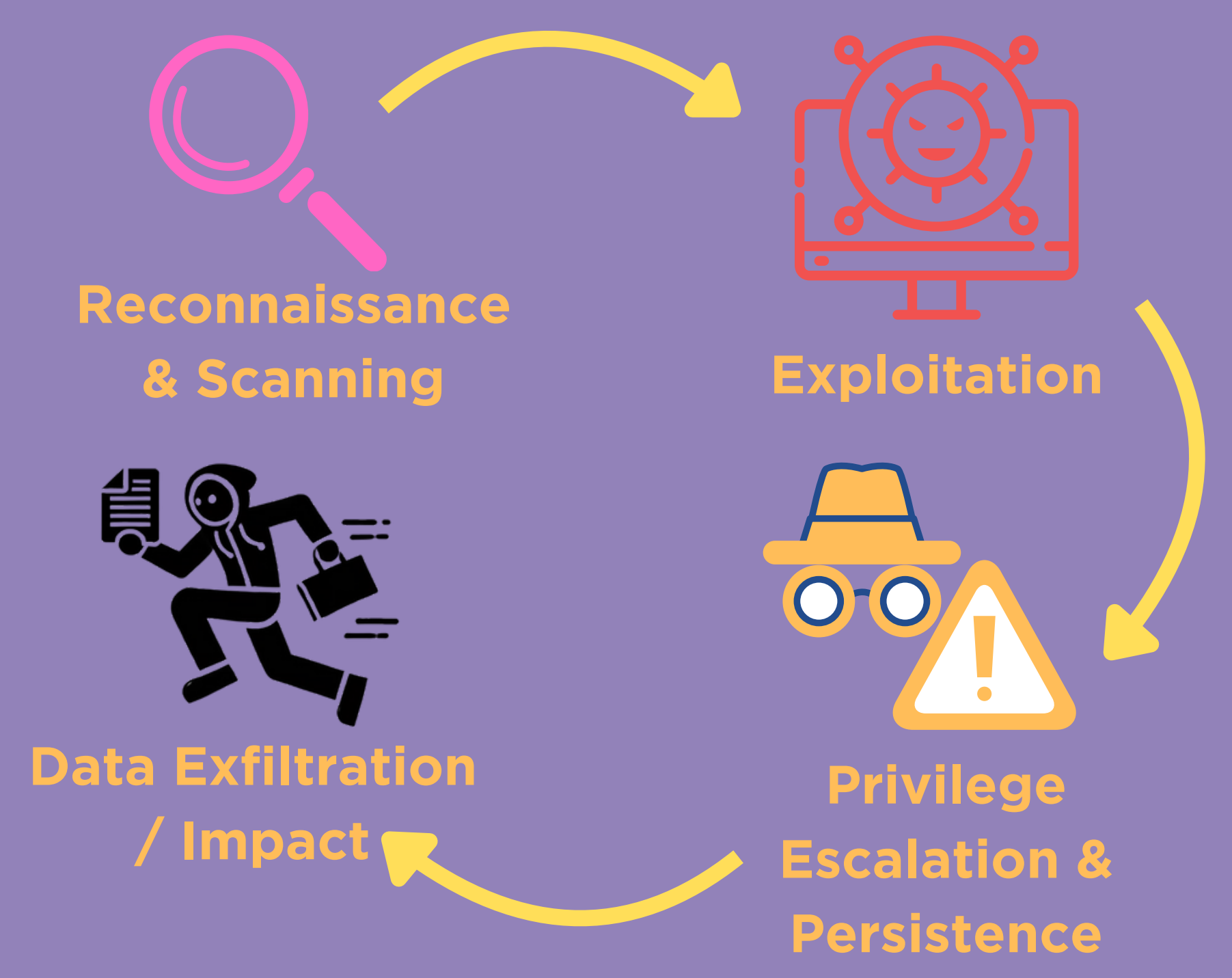
- Nmap scans open ports and services in a network
- Nikto identifies known vulnerabilities in web servers
- OWASP ZAP assesses web application security.

Examples of vulnerabilities that can be detected include **Opened ports, OS detection, SQL Injection, and Authentication Bypass**. The program is easy to use, requiring only the input of the desired vulnerability category, helping non-expert users better understand and mitigate cybersecurity risks.

INTRODUCTION

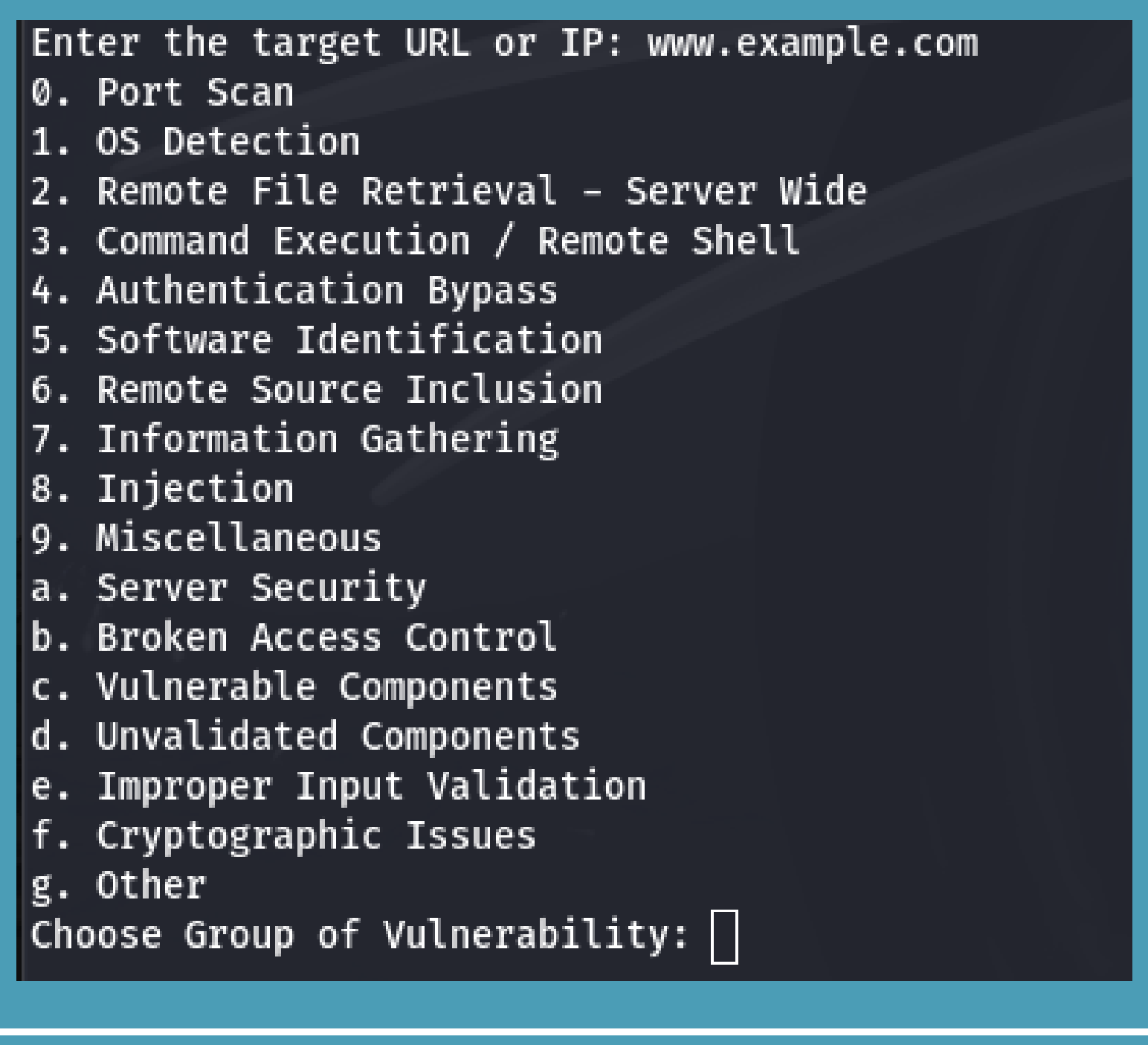
We developed the **Automated Vulnerability Detection Program** to improve efficiency and expedite the identification of security weakness.

This program streamlines system and network scanning by automating the detection of potential vulnerabilities, eliminating the need for manual inspection. By enhancing accuracy and speed, this tool enables organizations to proactively mitigate cybersecurity risks and strengthen their security posture.


Reconnaissance & Scanning
Exploitation
Privilege Escalation & Persistence
Data Exfiltration / Impact

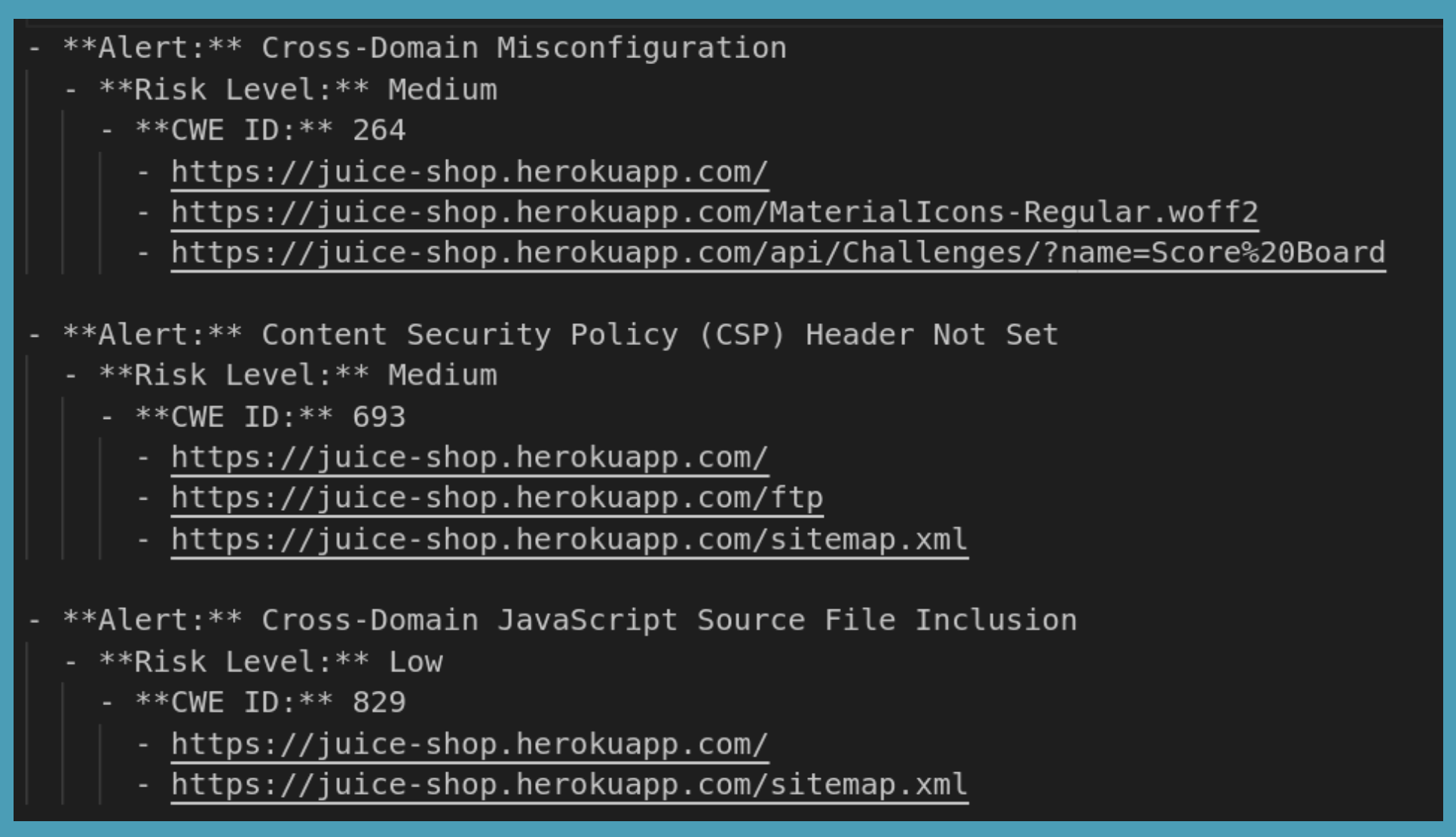
DESIGN

Example of an interface screen that allows the user to enter the vulnerability they want to check



```
Enter the target URL or IP: www.example.com
0. Port Scan
1. OS Detection
2. Remote File Retrieval - Server Wide
3. Command Execution / Remote Shell
4. Authentication Bypass
5. Software Identification
6. Remote Source Inclusion
7. Information Gathering
8. Injection
9. Miscellaneous
a. Server Security
b. Broken Access Control
c. Vulnerable Components
d. Unvalidated Components
e. Improper Input Validation
f. Cryptographic Issues
g. Other
Choose Group of Vulnerability: ▯
```

Example of vulnerability detection results

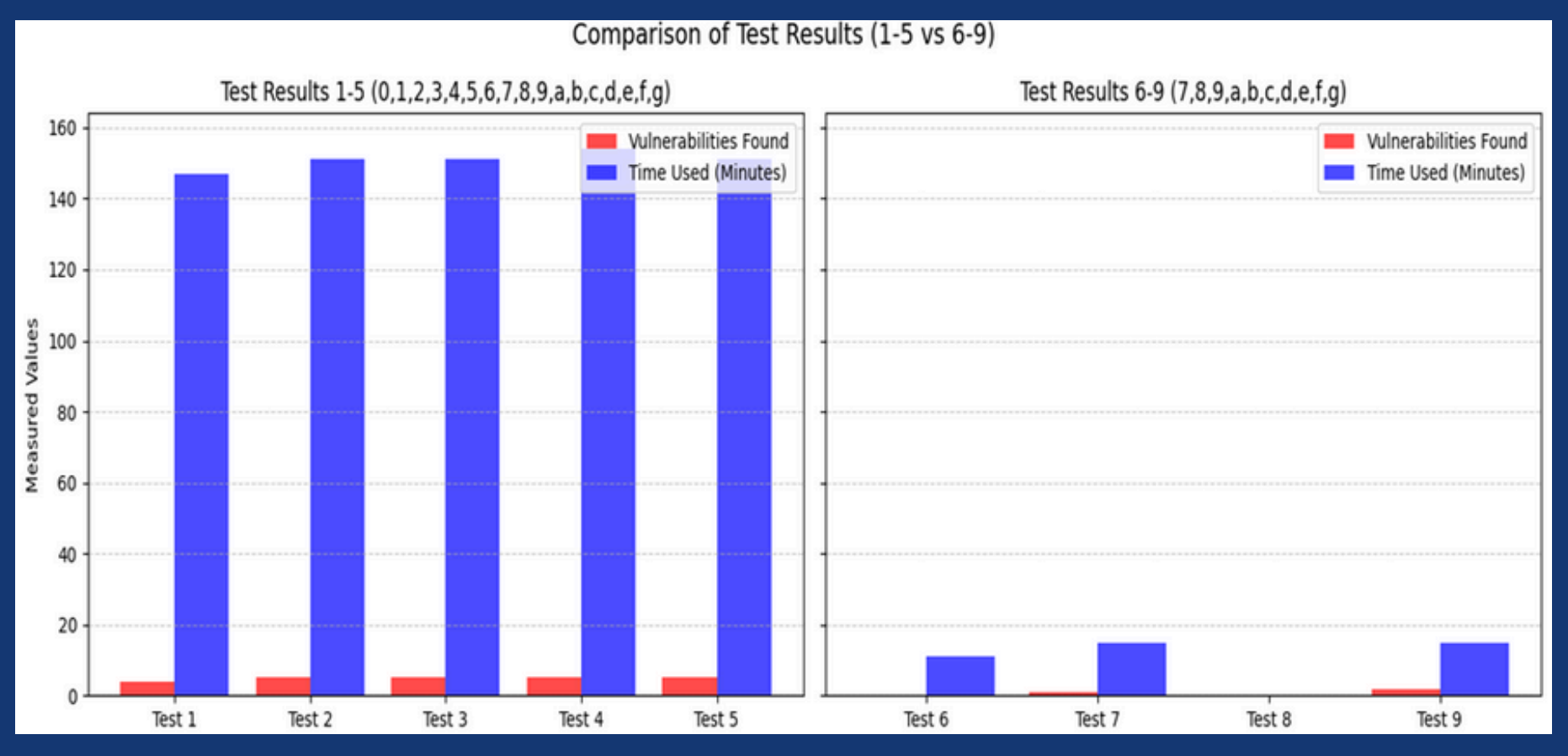


```
- **Alert:** Cross-Domain Misconfiguration
  - **Risk Level:** Medium
    - **CWE ID:** 264
      - https://juice-shop.herokuapp.com/
      - https://juice-shop.herokuapp.com/MaterialIcons-Regular.woff2
      - https://juice-shop.herokuapp.com/api/Challenges/?name=Score%20Board

- **Alert:** Content Security Policy (CSP) Header Not Set
  - **Risk Level:** Medium
    - **CWE ID:** 693
      - https://juice-shop.herokuapp.com/
      - https://juice-shop.herokuapp.com/ftp
      - https://juice-shop.herokuapp.com/sitemap.xml

- **Alert:** Cross-Domain JavaScript Source File Inclusion
  - **Risk Level:** Low
    - **CWE ID:** 829
      - https://juice-shop.herokuapp.com/
      - https://juice-shop.herokuapp.com/sitemap.xml
```

- **Summarizing potentially hazardous URLs**
- **Assigning a risk level**
- **Assigning a CWE standard**

RESULT

Overall, the program functions as expected within the predefined environment. However, some features are not fully operational, and certain functions have been partially removed by the developer.



Experimental results show a significant time difference between two test scenarios

Scenario one assessing Vulnerabilities 0-g Tests 1-5

- Use more time
- Use more resource
- Detect more vulnerabilities

Scenario one assessing Vulnerabilities 7-g Tests 6-9

- Use less time
- Use less resource
- Detect less vulnerabilities

TECHNOLOGY

OWASP ZAP
Nmap
Kali Linux
Nikto
VMware WorkSatation

CONCLUSION

The program functions as expected within the predefined environment, but some features remain incomplete or have been partially removed. Future improvements should enhance efficiency, add features, and optimize performance by refining or replacing certain tools.

REFERENCE

- A. Jakobsson and I. Häggström, Study of the techniques used by OWASP ZAP for analysis of vulnerabilities in web applications, Master's thesis, Linköping University, 2022. [Online]. Available: https://www.diva-portal.org/smash/get/diva2:1675227/FULLTEXT01.pdf.
- https://cwe.mitre.org/
- https://www.zaproxy.org/docs/alerts/

204499 Independent Study, Academic Year 2024

Department of Computer Science, Faculty of Science, Chiang Mai University