**Dissertation**

**Comparative study on open-source VAPT tools**

**Integrated Master in Computer Application**

(Cloud Technology and Information Security)

**Under**

**Assam Down Town University**

****

**Submitted by**

**Rituparna Seal**

ADTU/2018-23/ICA/011

Int. MCA (Cloud Technology and Information Security)

Semester: 9th

Computer Science & Engineering

**Under the Guidance of**

**Mr. Deepjyoti Roy**

Assistant Professor, Computer Science & Engineering

Faculty of Engineering & Technology

Assam DownTown University,

Guwahati-26, Assam

**Session**: September 2022- January 2023

***Certificate of Approval***

This project entitled “Comparative Study of open-source VAPT tools” is hereby approved as a credible engineering study carried out and presented in a satisfactory manner to narrate its acceptance as a prerequisite to the degree for which it is being submitted.

Examiner Dept of Computer Science and Engineering

***Certificate from Guide***

This is to certify that Rituparna Seal with registration number ADTU/2018-23/ICA/011 has completed Mini Project by 28th December 2022, entitled “Comparative Study of Open-source VAPT tools” at Assam Downtown University, and she has prepared this report under my guidance.

Date: Mr. Deepjyoti Roy

Assistant Professor

Dept of Computer Science and Engineering

Assam Downtown University

***Certificate from External Examiner***

This is to certify that Rituparna Seal, a student of Int. MCA (CTIS) 9th Semester has successfully completed the project entitled “Comparative study of open-source VAPT tools” during the session of September-December 2022.

It is to further certify that this project is the individual work of the candidate.

………………………………………

External Examiner

***Declaration***

I Rituparna Seal (ADTU/2018-23/ICA/011) hereby declare that the project entitled “Comparative study of open-source VAPT tools” submitted as Mini Project for 9th Semester, Int. MCA (CTIS) in Computer Science and Engineering to Assam Downtown University, Guwahati is an authentic record of my genuine work completed under the guidance of Dr. Banani Das and Mr. Deepjyoti Roy, Assistant Professor of Dept of Computer Science and Engineering, Assam Downtown University.

The information is true to the best of my knowledge.

…………………………………..

**Student’s Signature**

***Acknowledgment***

It is not possible to prepare a project report without the assistance and encouragement of other people. This one is certainly no exception.

At the very outset of this report, I would like to extend my sincere and heartfelt obligation towards all the personages who have helped me in the endeavor. Without their active guidance, help and cooperation, and encouragement, I would not have made headway in the project. I am ineffably indebted to my teachers, Mr. Deepjyoti Roy and Dr. Banani Das for their valuable guidance and support in the completion of this project.

I extend my gratitude to Assam Downtown University for giving me this opportunity.

I also acknowledge with a deep sense of reverence, my gratitude towards my parents and my friends who always help me morally.

Thanking you.

Rituparna Seal

***Abstract***

Web applications have become an easy target for hackers and attackers to steal sensitive information and also misuse any loophole. It has become the need of the hour to secure the web applications which has become a hub for hackers. In order to ensure the security of web applications one must perform testing and scanning on web applications regularly which is called Vulnerability Assessment and Penetration testing. Vulnerability assessment and penetration testing were performed in many stages using many tools such as Burp Suit, Sqlmap, Wpscan, Joomscan, and Nikto. This research was performed to compare the tools that are present as open-source in Kali Linux. There were six vulnerabilities found which were then analyzed and compared with respect to the tools. The open-source tools were compared and the result has been found.

***List of Figures and tables***

Figure 1 Use Case Diagram ………………………………………………24

Figure 2 Sequence Diagram………………………………………………25

Figure 3 Class Diagram………………………………………………….. 26

Figure 4 Project Lifecycle…………………………………………………27

Table 1 Parameters of Performance Analysis……………………………28

Figure 5 Using Burp Suite…………………………………………………30

Figure 6-8 Using Sqlmap…………………………………………….......31-32

Figure 9-15 Using Wpscan………………………………………………..33-36

Figure 16 Using Joomscan…………………………………………………37

Figure 17 Using Nikto………………………………………………………38

Table 2 Hardware Specification…………………………………………39

***List of Abbreviations***

|  |
| --- |
| VAPT - Vulnerability Assessment and Penetration Testing |
| OS – Operating System |
| Pen tester - Penetration tester |
| Reconnaissance - Information gathering phase of Ethical Hacking |
| OWASP – Open Web Application Security Project |
| Nmap – Network Mapper |
| TCP – Transmission Control Protocol |
| UDP – User Datagram Protocol |
| Nslookup – Name Server Lookup |
| DNS – Domain Name System |
| CLI – Command Line Interface |
| SYN - Synchronize |

**CONTENTS**

1. Introduction

1.1. Overview of the project………………………………………………………11

1.2. Motivation…………………………………………………………………….12

1.3. Scope & Objective……………………………………………………………12

1.4. Existing System………………………………………………………………12

1.5. Problem Definition…………………………………………………………...15

1.6. Proposed System……………………………………………………………..16

2. Theoretical Background………………………………………………………………..17

3. Literature Review or Related Work…………………………………………………18-19

4. Methodology

4.1 Tools used…………………………………………………………………20-21

4.2 Installation of the tools…………………………………………………….22-23

4.3. Use Case Diagram……………………………………………………………24

4.4. Sequence Diagram…………………………………………………………...25

4.5. Class Diagram………………………………………………………………..26

4.6. Schema Diagram……………………………………………………………..27

4.7 Parameters of performance analysis…………………………………………..28

5. Project Implementation

5.1. Description of the Software Used……………………………………………29

5.2. Snapshots(Module wise)………………………………………………….30-38

6. Result Analysis

6.1. Experimental setup…………………………………………………………..39

6.2. Performance comparison with existing system/algorithms…………….........40

7. Conclusion & Future Scope…………………………………………………………....41

*References……………………………………………………………………………………….*42-43

1. **INTRODUCTION**
   1. **OVERVIEW OF THE PROJECT**

The project was intended for a comparative study of VAPT tools of web applications, that fall under the field of Cyber Security. The project work was carried out using certain tools that are available as open source to find loopholes and vulnerable ports. We looked into the vulnerable ports and paths which could become an easy targets for attackers or hackers, and also looked into the types of attacks and risk factors involved.

Vulnerability assessment is the process of discovering and classifying security vulnerabilities in any environment. It measures possible weaknesses and also offers appropriate solutions to rectify the weaknesses. Furthermore, penetration testing involves exploiting those vulnerabilities to find out if a perceived vulnerability is legitimate. Hence, together, Vulnerability Assessment and Penetration Testing can help us find vulnerabilities in a network, system infrastructure, or web server, which could avoid and prevent various huge security threats in any given environment.

In this study, a comparison of five distinct open-source VAPT tools was carried out with respect to the vulnerabilities it detects best with the usage of two vulnerable web applications that are specifically designed for pen testers to test their skills and discover new areas in hacking. The fixes and risk ranges for the vulnerabilities were also discussed along with the steps that are performed to get to the conclusion.

* 1. **MOTIVATION**

With the use of the internet and technology, the tendency to take everything online and perform most of our daily activities, the security threat for both ends, that is the service providers and customers are increasing rapidly. Attackers target loopholes that are easy to hack, which results in a data breach, data theft, cross-site scripting, cross-origin resource sharing, and many more attacks.

Maintaining a website for the security of both parties has become mandatory. A less secure website is a risk to the organization and its clients. All Companies or Organizations that have their presence on the internet, need to have VAPT once in a while, not just because of the risk of sensitive data getting ex[posed, but also because most of the financial transactions nowadays are done through a business’ web application, which also brings the financial factors at risk.

If the risk factors are not discovered and discussed beforehand, this could lead to a real crisis for both the service provider and customers who rely on those services. In a world of the Internet and Technology where the risk of attacks and the attacks has been increasing rapidly, it is also equivalently crucial to determine what type of VAPT tool best suits the requirements, so that even small businesses and organizations, or seed businesses, who cannot afford to pay heavy amounts to the security professionals, because of less budget can carry out VAPT for their Web application with their technical team and fix the loopholes by themselves with the help of few simple steps.

* 1. **SCOPE & OBJECTIVE**

The primary objective of the project was to learn about any five open-source VAPT tools present for Web application scanning and testing through Kali Linux. After the study of the tools, we moved forward to a comparison of the tools with respect to the vulnerabilities and platforms in which they work best. The goal was also to determine which tools are best for what type of web application as there are various ways in today’s world through which a web application can be hosted. It can be made from scratch using many programming languages or it can also be made using WordPress.

The objective was also to learn about the types of vulnerabilities discovered and how those can be mitigated if they persist in a Web Application. Furthermore, the attacks that could happen and the risk level of those attacks were also concluded.

The range of the project was to make people aware of some open source VAPT tools better and also find what type of tools is best for determining a certain type of vulnerability. This was because there are many organizations and businesses that have their presence on the Internet, however, their web applications are extremely insecure and this could question the safety of the users visiting and using their websites. In order to ensure that the users' sensitive information stays secure, it is necessary to know what options they have when it comes to scanning, testing, and securing their web applications.

* 1. **EXISTING SYSTEM**

There are numerous open-source VAPT tools available for web application scanning. Conversely, there are no tools specified that target certain kinds of vulnerabilities. All the tools that are available are claimed to be the best for detecting all kinds of vulnerabilities. If we talk about the OWASP top 10, there are tools that are said to cover each one of them and not miss any. But, it is better to know which tool is best for detecting what kind of vulnerability.

In the existing system, any one of the various VAPT tools is trusted blindly and manual scan is not given priority. Scanning a web application manually makes sure that there were no vulnerabilities that were missed out while an automatic scan was running. This is the reason one must know what tools work best in detecting a particular vulnerability. If we look into the current scenario of the security of web applications, we will realize that most of the attacks on web applications happen either because they were not prior scanned or because the manual scan was not taken into consideration by many security professionals.

This is the reason why manual scanning of web applications must be considered and for that, it is necessary to know what options are available as open-source, free-to-use security testing tools.

* 1. **PROBLEM DEFINITION**

In today’s world, most tasks are performed through web applications. Websites have made human work much easier, as it connects documents locally and remotely.Web applications facilitate users to perform several tasks. Furthermore, after the outbreak of COVID-19 the usage of web applications for payment and transfer of valuable goods has increased rapidly. People find it much easier and simple to go cashless. Nonetheless, with the augment of usage of web applications in numerous sectors, such as healthcare, education, business, shopping, etc. the safety and security of web applications have become a concern.

With the raise in the usage of the internet for every other use, cyber-attacks have also been increasing rapidly. In addition, cyber-attacks are not only the concern of private organizations but also distinct government organizations worldwide that have become a target of many cyber attacks in recent times.

Moreover, there are many VAPT tools available, and identifying which one best suits the needs, is a tough task. Hence, the need to understand which tools work best for what kind of vulnerability is highly necessary. If manual scanning is not performed, there remains some percent of chances to overlook any vulnerability while automatic scanning seems to be less potential for an attack, but can become a spot for the attackers or hackers to target, even if small.

* 1. **PROPOSED SYSTEM**

The system that has been proposed after proper implementation and research was the use of a manual scanning technique for VAPT of web applications even after the automatic scan was carried out earlier.

The research was also done on the five tools that are Burp Suite, SQL map, Wpscan, Joomscan, and Nikto to find out which tools detect what type of vulnerabilities and which one misses some. After carrying out all the steps this was found that Nikto discovers more vulnerabilities than any other tool and can also be trusted with finding vulnerabilities that are not detected by automatic scanning or any other manual scanning tool.

1. **THEORETICAL BACKGROUND**

Vulnerability is any weakness in any environment that can be exploited by hackers or attackers to achieve their unlawful goals. Vulnerability assessment is the process of discovering and classifying security vulnerabilities in any environment. It measures possible weak paths and also offers appropriate solutions to rectify the weaknesses. Furthermore, penetration testing involves exploiting those vulnerabilities to find out if a perceived vulnerability is legitimate. Hence, together, Vulnerability Assessment and Penetration Testing can help us find vulnerabilities in a network, system infrastructure, or web server, which could avoid and prevent various huge security threats in any given environment. The security and safety of any infrastructure primarily depend on VAPT and the actions taken to mitigate the vulnerabilities found in the VAPT report.

In this paper, we have used some open-source VAPT tools to find vulnerabilities in Web applications and also determine which tools work best for which vulnerability and in what interface, and in what type of Web application. Web applications are programs that are made available to users via a simple user-friendly interface, which is commonly stored in a remote server.

The first phase of VAPT is Reconnaissance which is the information-gathering phase carried out before VAPT. In reconnaissance as much information is gathered about the target as possible, this could be used during the testing and scanning part. Reconnaissance is performed to gather basic information such as IP address, open or closed ports, service provider, platform, etc. of any given infrastructure.

VAPT tools are primarily of two types. One is open source that can be used by everyone and the other is the paid versions that are needed to be purchased. In this project, the open-source VAPT tools are used to test the functionality and scan the web applications for any vulnerabilities.

1. **LITERATURE REVIEW OR RELATED WORK**

Several research studies have discussed the issue of evaluating the performance and effectiveness of web application assessment tools. Several researchers researched only commercial or open-source tools, while others combined open-source and commercial tools.

Ricardo Araújo et al. [1] presented a recent and detailed comparison of security scanners that are available for use by organizations with lower resources, such as small and medium-sized enterprises. In addition, they compared open-source tools and highlighted their performances according to the number of vulnerabilities detected and false positives.

Roy Shanto et al. [2] provided a survey that is based on taxonomy and a detailed overview of adversarial reconnaissance techniques. The taxonomy categorizes reconnaissance techniques based on the technical approach, including target footprinting, social engineering, network scanning, and local discovery.

Doupé et al. [3] assessed ten tools which are Acunetix, AppScan [[**18**](https://www.mdpi.com/2076-3417/12/8/4077#B18-applsci-12-04077)], BurpSuite [[**23**](https://www.mdpi.com/2076-3417/12/8/4077#B23-applsci-12-04077)], Arachni, Hailstorm, NTOSpider, Paros, N-Stalker, Webinspect, and W3af for web application assessment. These tools were selected to consider a variety of open-source and commercial scanners.

Lilan Hu et al. [4] presented an in-depth study of the existing security vulnerability detection technology, combined with the development process of machine learning security vulnerability detection technology, the requirements of the security vulnerability detection model are analyzed in detail, and a cross-site scripting security vulnerability detection model for web application is designed and implemented. Based on the existing network vulnerability detection technology and tools, the verification code identification function is added, which solves the problem that the data can be submitted to the server only by inputting the verification code.

Richard Amankwah et al. [5] provide a comparison work to determine the vulnerability detection capabilities of eight web vulnerability scanners (both open and commercial) using two vulnerable web applications: WebGoat and Damn vulnerable web application. The eight WVSs studied were: Acunetix; HP WebInspect; IBM AppScan; OWASP ZAP; Skipfish; Arachni; Vega; and Iron WASP. The performance was evaluated using multiple evaluation metrics: precision, recall, Youden index, OWASP web benchmark evaluation, and the web application security scanner evaluation criteria. The experimental results show that, while commercial scanners effectively detect security vulnerabilities, some open-source scanners (such as ZAP and Skipfish) can also be effective.

The evaluation focused mainly on the selected web application assessment capabilities against XSS (Cross-Site Scripting), SQL injection, code injection, and broken access controls. Exposed vulnerabilities were categorized to enable the assessment process for seventeen different vulnerabilities.

1. **METHODOLOGY**
   1. **TOOLS USED**
      1. **Nmap** – Nmap is used in the reconnaissance of web applications. It is an open-source tool designed for discovering networks and also for security auditing. We used to discover open and closed ports for gathering information which helped us in VAPT. It works with TCP and UDP. We discovered the options for scanning using the command “nmap –h”
      2. **Recon –ng** – Recon –ng is designed specifically for web reconnaissance and is written in Python. It is a passive reconnaissance framework. Using this tool we were able to find the current version of the application.
      3. **Nslookup** – We used Nslookup in Kali Linux command prompt to get information about the DNS server. It was used to find any specific details about the DNS record. For a detailed response, we used the command “nslookup –type=any” in the CLI.
      4. **P0f –** With the help of p0f we detected the OS of the target. It uses SYN packets to detect the OS, without sending any data.

* + 1. **Burp Suite** – Burp suite comprises a set of tools that are used for penetration testing of web applications. There are two versions of Burp Suite that exist. One is the Community edition and the other is the professional edition. We implemented and checked the functionality of the Community edition on testphp.vulnweb.com.
    2. **SQL map** – Sql map is an open-source VAPT tool in Kali Linux. It is used on the vulnerable web application testphp.vulnweb.com for this project. It is one of the most used open source VAPT tools for web application testing. It is easy to use and also provides accurate results.
    3. **Wpscan** – Wpscan is installed on Kali Linux to scan a web application that runs on WordPress. This tool only works for Web applications that are hosted on WordPress and hence we used torontostandard.com as our target which is a vulnerable web application hosted on WordPress. We have included this in our research because of the increasing number of web applications hosted on wordpress every day.
    4. **Joomscan** – Joomscan is a pre-installed tool on Kali Linux. It is easy to use and has a user-friendly interface which makes it very convenient to use and detect vulnerabilities in a web application.
    5. **Nikto** – Nikto was used to find vulnerabilities on testphp.vulnweb.com. It has various test cases which makes it a very great choice for VAPT of a web application.
  1. **INSTALLATION OF THE TOOLS**
     1. **Nmap** - Nmap is a built-in tool in Kali Linux and does not require any further installation of packages. The current version of nmap is 7.93.
     2. **Recon-ng** – To install Recon-ng on Kali Linux we used the simple command “*sudo apt install recon-ng*” on Kali Linux. The installation size was 272 KB and the current version is 5.1.2.
     3. **Nslookup** – Nslookup does not need to be installed, and can be used directly on Kali Linux command prompt.
     4. **P0f** - To install Recon-ng on Kali Linux we used the simple command “*sudo apt install p0f*” on Kali Linux. The installation size was 219 KB and the current version is 3.09b.
     5. **Burp Suite** - To install Recon-ng on Kali Linux we used the simple command “*sudo apt install burp suite*” on Kali Linux. The installation size was 223.45 MB and the current version is 2022.9.6.
     6. **SQL map –** SQL map does not need to be installed, and can be used directly on Kali Linux command prompt
     7. **Wpscan –** To install Wpscan we require to clone it from github with the help of a few simple commands-

*sudo apt-get install git*

*sudo apt-get install git ruby ruby-dev libcurl4-openssl-dev make*

*git clone https://github.com/wpscanteam/wpscan.git*

*cd wpscan*

*sudo gem install bundler && bundle install --without test*

* + 1. **Joomscan** - To install Joomscan on Kali Linux we used the simple command “*sudo apt install joomscan*” on Kali Linux. The installation size was 274 KB and the current version is 0.0.7.
    2. Nikto - To install Nikto on Kali Linux we used the simple command “*sudo apt install nikto*” on Kali Linux. The installation size was 2.38 MB and the current version is 2.1.6.
  1. **USE CASE DIAGRAM**

Penetration Tester

Fig: 1

* 1. **SEQUENCE DIAGRAM**

Third Phase

Second Phase

First Phase

Penetration Tester

Reconnaissance

Comparison

Literature Study

Recommendation

Discovery

Planning

Stop

Fig: 2

* 1. **CLASS DIAGRAM**

Vulnerability Asessment

Reconnaissance

Exploiting Vulnerabilities

Discovering Vulnerabilities

Penetration Testing

Recommendation or mitigation

Fig: 3

* 1. **PROJECT LIFECYCLE**

Fig: 4

* 1. **PARAMETERS OF PERFORMANCE ANALYSIS**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Severity | | High | High | Low | Low | High | High |  | |
| Tools | Vulnerabilities  detected | SQL Injection | XSS | Robots.txt | CORS | CSRF | XST | Total Vulnerabilities | Web app |
| Burp Suite | No | No | No | No | Yes | No | 1 | Testphp.vulnweb.com |
| Sqlmap | Yes | Yes | No | No | No | No | 2 | Testphp.vulnweb.com |
| Wpscan | No | No | No | Yes | No | No | 1 | Torontostandard.com |
| Joomscan | Yes | No | Yes | No | No | No | 2 | Testphp.vulnweb.com |
| Nikto | No | Yes | Yes | No | No | Yes | 3 | Testphp.vulnweb.com |

**Table: 1**

1. **PROJECT IMPLEMENTATION**
   1. **DESCRIPTION OF SOFTWARE USED**

The software that is primarily used to perform the task is Kali Linux as an Operating System and VMWare Workstation Pro as Virtual Machine.

* + 1. **Reconnaissance**

Reconnaissance is where information is gathered about the particular web application that is to be tested. In this project information is gathered using the following tools:

1. Nmap
2. Recon-ng
3. Nslookup
4. P0f
   * 1. **VAPT**

Vulnerability analysis on the web applications is done using some open-source Kali Linux tools. The tools that are used for VAPT are:

1. Burp Suite
2. SQL map
3. Wpscan
4. Joomscan
5. Nikto
   * 1. **Web applications**

We have performed VAPT on two web applications that are designed for the purpose of hacking and testing. The reason why we have performed VAPT on two web applications and not just one will be discussed later in this paper.

1. Testphp.vulnweb.com
2. Torontostandard.com
   1. **SNAPSHOTS**
      1. **Using Burp Suite**

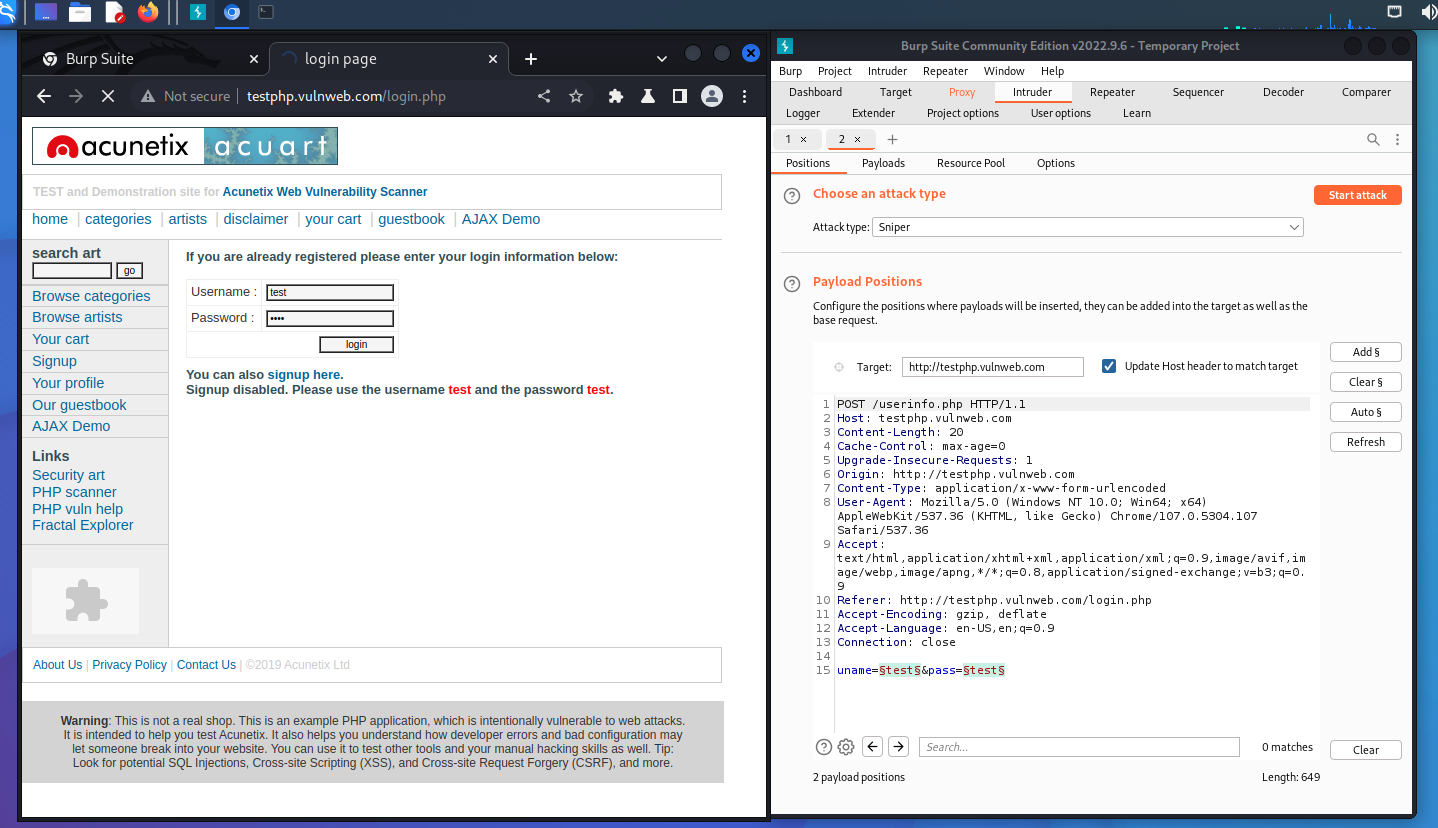
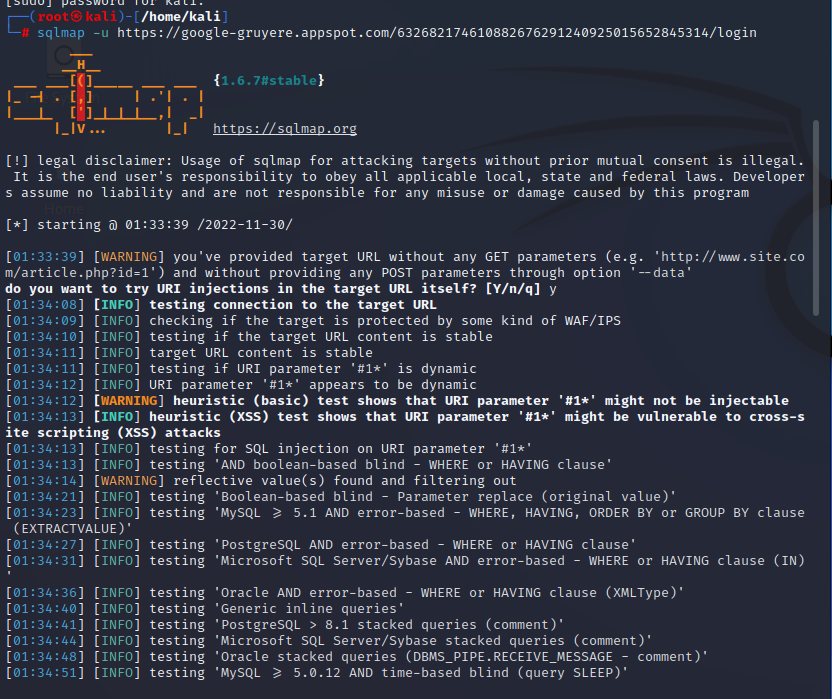
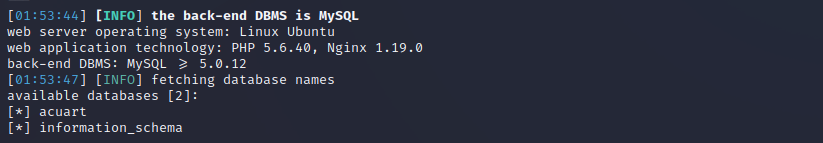


Fig: 5

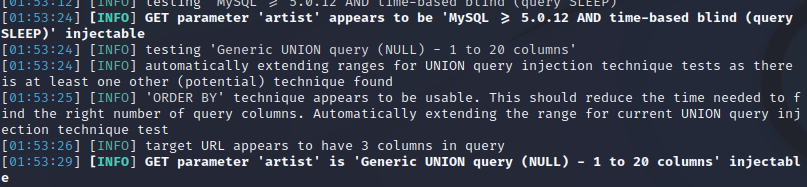
* + 1. **Using Sql map**

****

**Fig: 6**

****

**Fig: 7**

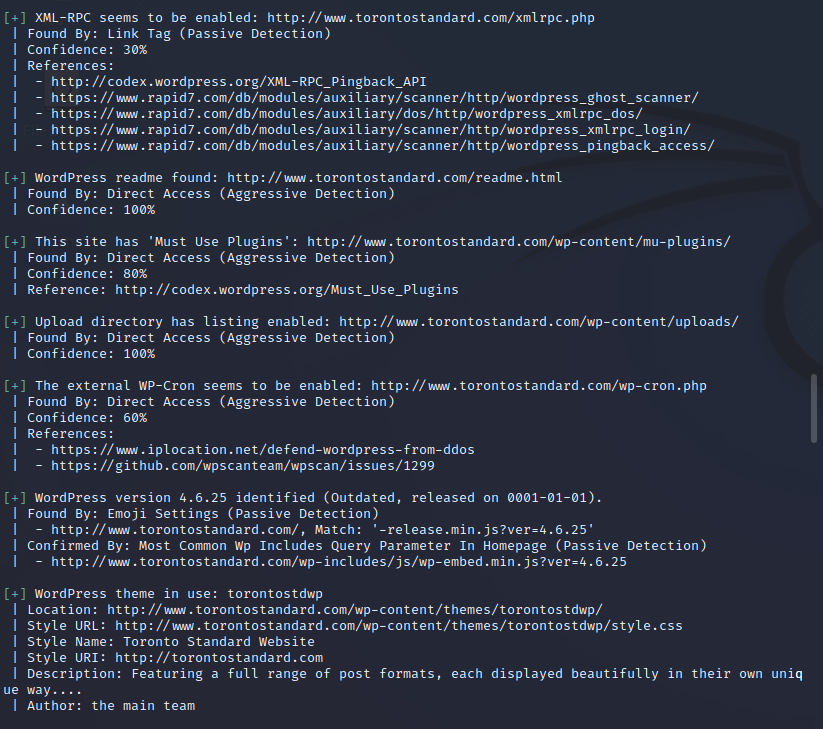
****

**Fig: 8**

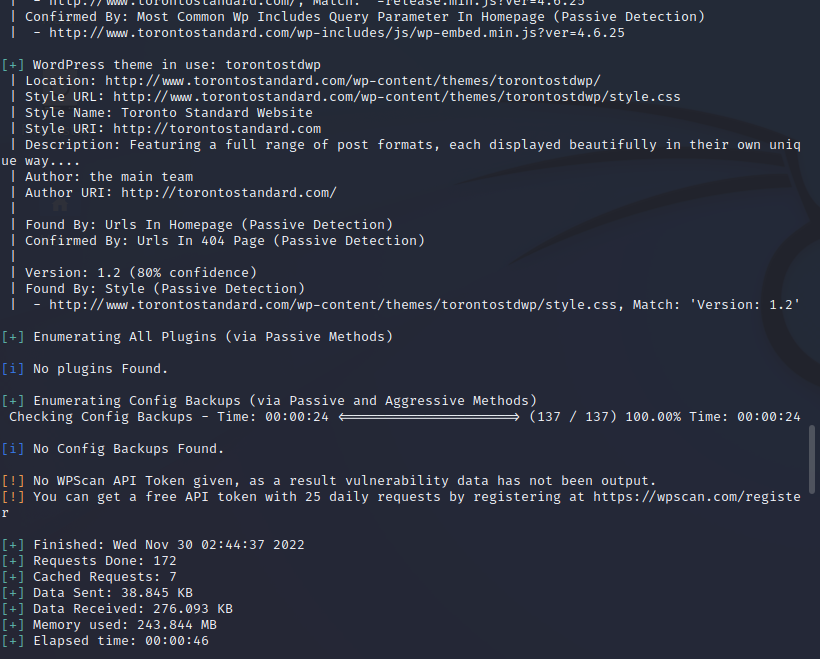
* + 1. **Using Wpscan**



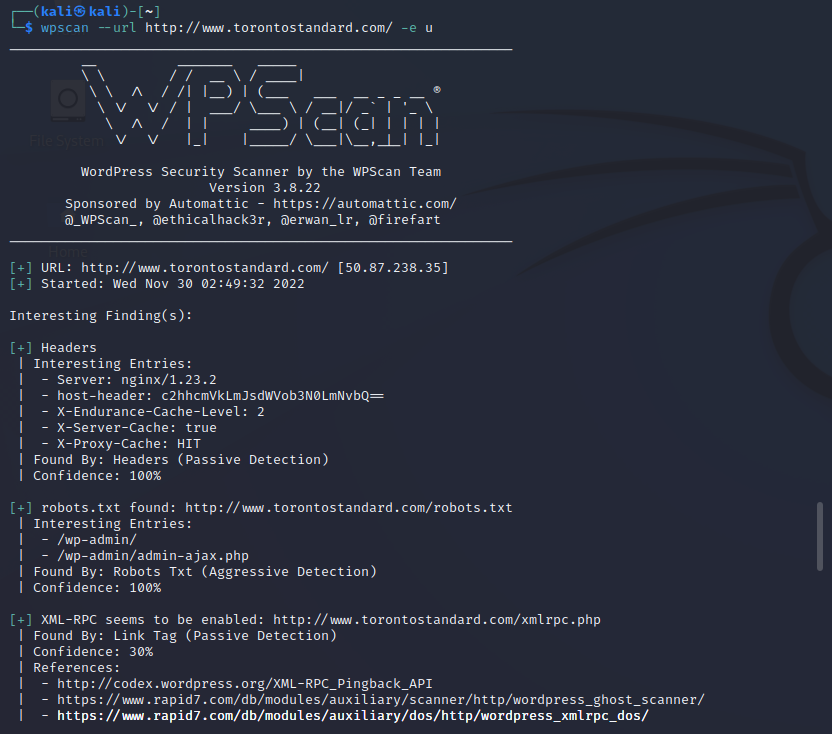
**Fig: 9**



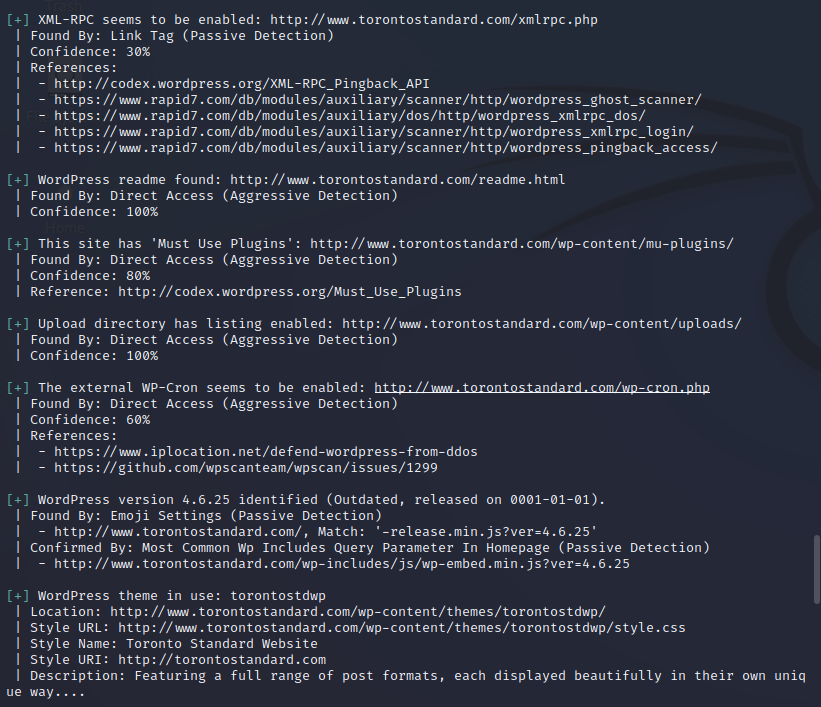
**Fig: 10**



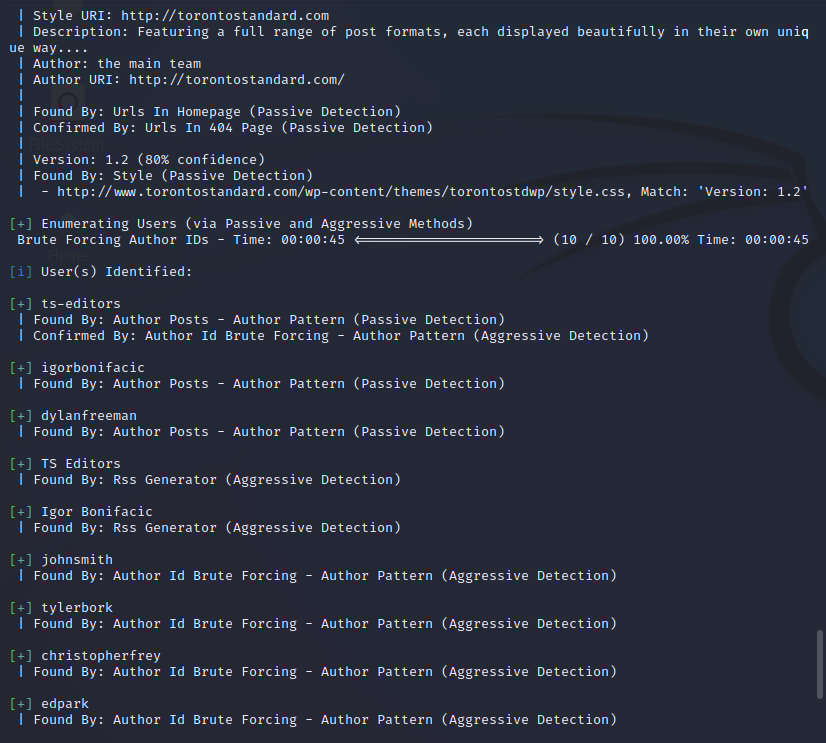
**Fig: 11**



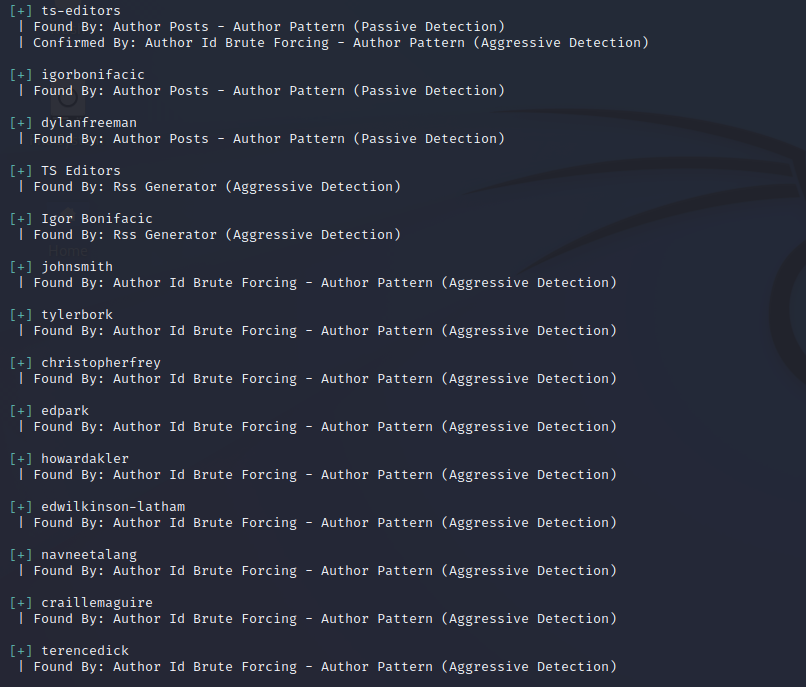
**Fig: 12**



**Fig: 13**

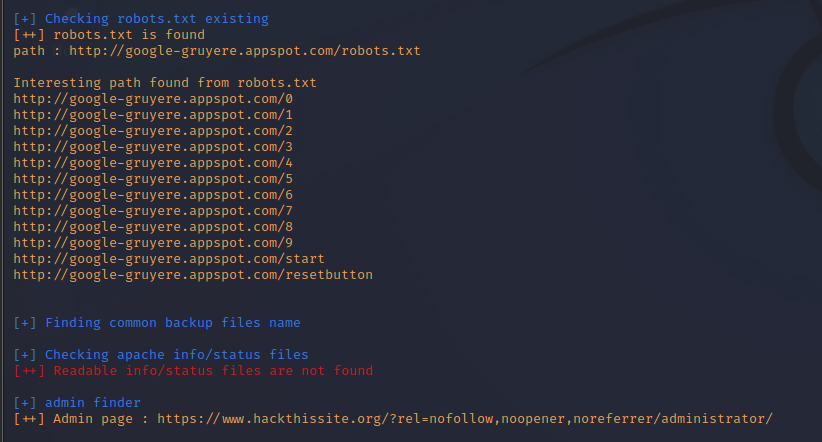


**Fig: 14**



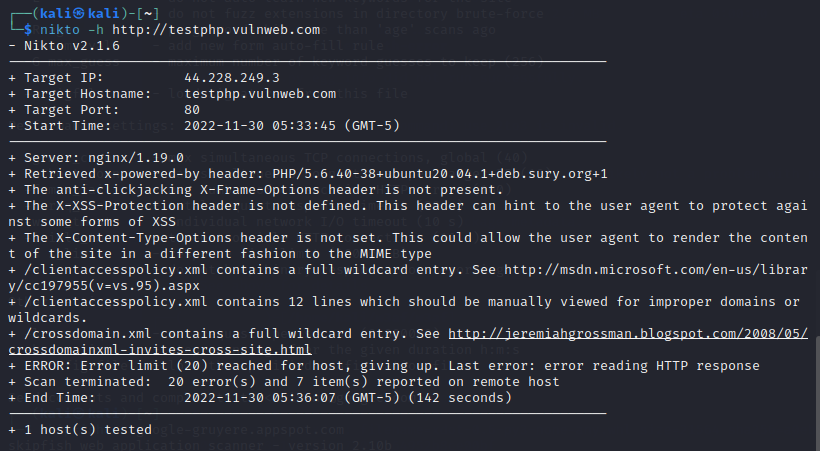
**Fig: 15**

* + 1. **Using Joomscan**

****

**Fig: 16**

* + 1. **Using Nikto**

****

**Fig: 17**

1. **RESULT ANALYSIS**
   1. **EXPERIMENTAL SETUP**
      1. **Hardware Specification**

The detailed hardware specification used to carry out the project is listed below

|  |  |
| --- | --- |
| Processor : | Intel(R) Core(TM) i5-8250U CPU @ 1.60GHz 1.80 GHz |
| Memory: | 8 GB |
| System: | 64-bit |
| Operating System | Kali Linux |

Table 2

The VAPT is done in a Virtual environment using VM Ware Workstation Pro with Kali Linux as Virtual Operating System. All the functionalities are performed inside Kali Linux Operating System.

* + 1. **Software Specification**

The software that is primarily used to perform the task is Kali Linux as an Operating System and VMWare Workstation Pro as Virtual Machine.

* 1. **PERFORMANCE COMPARISON WITH EXISTING SYSTEM**

The existing system of VAPT for web applications does not have any specific tool to find a specific vulnerability. Moreover with the use of the auto scan feature of various tools some minor vulnerabilities are overlooked which is not in the case of manual scanning using VAPT tools.

Hence, knowing which tool works best for what type of environment or platform and on what vulnerabilities it becomes easy to perform a VAPT.

1. **CONCLUSION AND FUTURE SCOPE**

Through this research work, this was found that not all open-source tools detect every vulnerability that exists in a web application, and therefore it is necessary to know which tool works best for what type of vulnerability and with what type of web application. We have found run tests on such open-source tools and have listed the vulnerabilities detected with each one of them

In our future research, we aim in comparing more such open-source tools used for VAPT to discover more about them.

***References***

1. Araújo, Ricardo, et al. “A Performance Assessment of Free-To-Use Vulnerability Scanners - Revisited.” *ICT Systems Security and Privacy Protection*, 2021, pp. 53–65, https://doi.org/10.1007/978-3-030-78120-0\_4.

<https://link.springer.com/chapter/10.1007/978-3-030-78120-0_4>

1. “Survey and Taxonomy of Adversarial Reconnaissance Techniques | ACM Computing Surveys.” *ACM Computing Surveys*, 2022, dl.acm.org/doi/abs/10.1145/3538704.

<https://dl.acm.org/doi/abs/10.1145/3538704>

1. Doupé, Adam, et al. “Why Johnny Can’t Pentest: An Analysis of Black-Box Web Vulnerability Scanners.” *Detection of Intrusions and Malware, and Vulnerability Assessment*, 2010, pp. 111–31, https://doi.org/10.1007/978-3-642-14215-4\_7.

<https://link.springer.com/chapter/10.1007/978-3-642-14215-4_7>

1. Hu, Lilan, et al. “Web Application Vulnerability Detection Method Based on Machine Learning.” *Journal of Physics: Conference Series*, vol. 1827, no. 1, Mar. 2021, p. 012061, https://doi.org/10.1088/1742-6596/1827/1/012061.

<https://iopscience.iop.org/article/10.1088/1742-6596/1827/1/012061/meta>

1. Amankwah, Richard, et al. “An Empirical Comparison of Commercial and Open‐Source Web Vulnerability Scanners.” *Software: Practice and Experience*, vol. 50, no. 9, July 2020, pp. 1842–57, https://doi.org/10.1002/spe.2870.

<https://onlinelibrary.wiley.com/doi/abs/10.1002/spe.2870>

1. “View of Vulnerability Assessment and Penetration Testing on the Xyz Website Using Nist 800-115 Standard.” *Syntaxliterate.co.id*, 2022, [www.jurnal.syntaxliterate.co.id/index.php/syntax-literate/article/view/5800/3111](http://www.jurnal.syntaxliterate.co.id/index.php/syntax-literate/article/view/5800/3111).
2. SACHIN UMRAO, et al. “VULNERABILITY ASSESSMENT and PENETRATION TESTING.” *Interscience Research Network*, 2016, [www.interscience.in/ijcct/vol7/iss3/11/](http://www.interscience.in/ijcct/vol7/iss3/11/).
3. Goel, Jai Narayan, and B. M. Mehtre. “Vulnerability Assessment & Penetration Testing as a Cyber Defence Technology.” *Procedia Computer Science*, vol. 57, 2015, pp. 710–15, <https://doi.org/10.1016/j.procs.2015.07.458>.
4. Penetration testing A Hands-On In t r o d u c t i o n to Hacking by Georgia Weidman
5. S. Nagpure and S. Kurkure, "Vulnerability Assessment and Penetration Testing of Web Application," 2017 International Conference on Computing, Communication, Control and Automation (ICCUBEA), 2017, pp. 1-6, doi: 10.1109/ICCUBEA.2017.8463920.
6. “What Is Cross-Site Scripting and How Can You Fix It?” *Acunetix*, 21 July 2022, [www.acunetix.com/websitesecurity/cross-site-scripting/](http://www.acunetix.com/websitesecurity/cross-site-scripting/).
7. “Cross-Site Request Forgery Prevention - OWASP Cheat Sheet Series.” *Owasp.org*, 2020, cheatsheetseries.owasp.org/cheatsheets/Cross-Site\_Request\_Forgery\_Prevention\_Cheat\_Sheet.html.
8. “Cross Site Tracing | OWASP Foundation.” *Owasp.org*, 2012, owasp.org/www-community/attacks/Cross\_Site\_Tracing.