International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING

ISSN:2147-6799 www.ijisae.org Original Research Paper

A Proposed Framework of VAPT Services in Web Application Deployed on Infrastructure as a Service (IaaS)

*Noraida Haji Ali[1], Nuur Ezaini Akmar Ismail[2], Masita Jalil[3], Farizah Yunus[4], Ahmad Dahari Jarno[5]

Submitted: 25/11/2023 **Revised**: 07/01/2024 **Accepted**: 18/01/2024

Abstract: Most companies in Malaysia require their employees to work from home due to the COVID-19 pandemic. This situation also increased the number of data generated from various sources, thus exposing them to different security risks. Even though the employees are encouraged to work from home because of the COVID-19 pandemic, they still need to communicate among themselves to do their work. However, working from home depends mainly on cloud computing (CC) applications that help employees accomplish their daily work efficiently. Injection attacks, such as SQL injection and Cross-Site Scripting (XSS), are critical security vulnerabilities that can lead to unauthorized access, data breaches, and potential service disruptions in web applications. With the increasing adoption of cloud computing, web applications deployed on cloud platforms like Amazon Web Services (AWS) are becoming more prevalent and vulnerable to such attacks. Therefore, it is crucial to develop practical Vulnerability Assessment and Penetration Testing (VAPT) techniques specifically tailored to identify and detect injection vulnerabilities in web applications deployed on AWS. However, existing VAPT methodologies often need more comprehensive coverage for injection vulnerabilities in cloud-based web applications, and they may not consider the unique characteristics and challenges associated with the AWS environment. This research addresses this gap by proposing an enhanced VAPT framework focusing specifically on injection attacks in web applications deployed on AWS.

Index Terms: *cloud computing, IaaS, Injection, SQL injection, Cross-Site Scripting (XSS), AWS and penetration testi*

1. Introduction

Penetration testing is a process or activity conducted by a Penetration tester without any evil motive to find security vulnerabilities and system flaws in a target environment before the weaknesses are exploited by the attackers [1,2]. This testing is crucial, especially to the financial institutions; and government sectors, since they store sensitive information, such as personal details, banking information, health data, etc, from their end user or customers. The purpose of conducting penetration testing is to discover vulnerabilities and any system flaws in the system environment, then mitigate the issues before they are exposed to the public, to ensure the system offered is secure, thus gain trust from the customer, to identify and prioritize security risks and to test security implementation in configuration working as expected [3].

Based on NIST SP800-145 [4]. A shared pool of reconfigurable computing resources (such as networks, servers, storage, apps, and services) that can be quickly provisioned and released with little management work or service provider interaction is what cloud computing is defined as: a model for ubiquitous, convenient, on-demand network access. Users can place all data and applications in the Cloud, while another party, the Cloud Service Provider (CSP), controls other processes

This research focused on developing a new framework and discovered the best method to conduct Vulnerability Assessment and Penetration Testing (VAPT) for the injection attacks in the web application deployed in the cloud computing service model Infrastructure as a Service (IaaS). There are three (3) types of cloud service models: Software as a Service (SaaS), Platform as a Service (PaaS), and Infrastructure as a Service (IaaS). For this research, we will use the IaaS platform provided by Amazon Web Services (AWS) since AWS offers virtualized computing resources, such as virtual machines, storage, and networking capabilities, enabling us to build and manage our Infrastructure within the AWS environment. We will configure the simulation of the injection attacks, such as SQL Injection and Cross-Site Scripting (XSS), using vulnerable web applications deployed in the AWS.

Throughout this paper, we will discuss the background of this research and provide brief information about the VAPT, which includes the VAPT approaches such as black-box testing, white-box testing, gray-box testing, dynamic testing, and static analysis, as well as the type of injection attacks in Section 2. Section 3 will elaborate on the research process, including the methodology for this proposed framework. Following that, in Section 4, we will explain in detail the proposed framework of VAPT in cloud computing, and in Section 5, we will elaborate more on the result and discussion

[1,2,3,4] *Faculty of Ocean Engineering Technology and Informatics, Universiti Malaysia Terengganu.*
[5] *Department at CyberSecurity Malaysia Security Evaluation Facility CyberSecurity Malaysia*
* *Email: aida@umt.edu.my*

of this research. Finally, in the final section, we will outline our next steps for future research to ensure that this research is ongoing and relevant.

2. Research Background

In this section, we will cover the study in two (2) research fields: 1) the cloud computing service model focusing more on IaaS, and 2) the reported injection approach and attack in the web application deployed in the Cloud such as SQL injection, Cross-Site Scripting (XSS) and many more.

There are three cloud computing service models [NIST SP 800-145] [4][23]:

- Infrastructure as a Service (IaaS) also referred as Resource Clouds [23] - The provider provides hardware and network connectivity. The subscriber is responsible for the virtual machine and everything that runs within it.

- Platform as a Service (PaaS) - The provider provides all the components needed to operate the application, and the subscriber provides the application they want to deploy.

- Software as a Service (SaaS) - It is also referred to as Application or a Service Clouds [23]

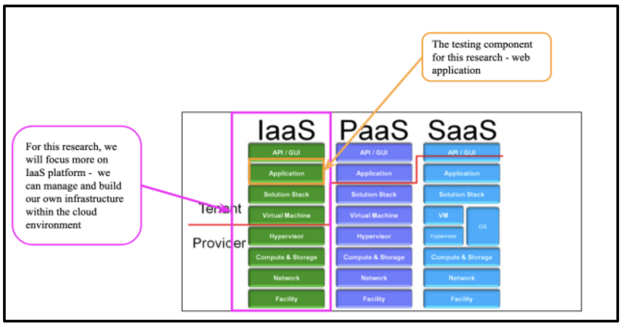
The testing component for this research - web application
For this research, we will focus more on IaaS platform - we can manage and build our own infrastructure within the cloud environment
IaaS
PaaS
SaaS
API / GUI
Application
Solution Stack
Virtual Machine
Hypervisor
Compute & Storage
Network
Facility
VM
OS
Tenant
Provider

Fig 1: The type of cloud service models

Injection attacks, such as SQL injection and Cross-Site Scripting (XSS), are critical security vulnerabilities that can lead to unauthorized access, data breaches, and potential service disruptions in web applications [6][7][8][9]. With the increasing adoption of cloud computing, web applications deployed on cloud platforms like Amazon Web Services (AWS) are becoming more prevalent and vulnerable to such attacks. Therefore, it is crucial to develop practical Vulnerability Assessment and Penetration Testing (VAPT) techniques specifically tailored to identify and detect injection vulnerabilities in web applications deployed in AWS.

There are various types of attacks or vulnerabilities, as mentioned in OWASP Top 10 (web) [10] and OWASP Code Review [11]. However, we focus on injection attacks (No.3 in OWASP Top 10 for web and No.1 in OWASP Code Review) as illustrated in Fig 2.

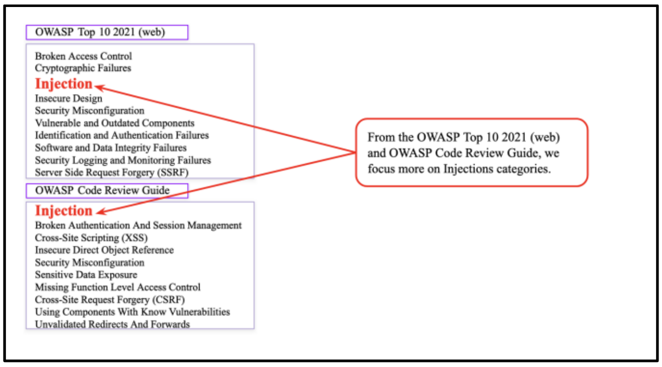
OWASP Top 10 2021 (web)
Broken Access Control
Cryptographic Failures
Injection
Insecure Design
Security Misconfiguration
Vulnerable and Outdated Components
Identification and Authentication Failures
Software and Data Integrity Failures
Security Logging and Monitoring Failures
Server Side Request Forgery (SSRF)
OWASP Code Review Guide
Injection
Broken Authentication And Session Management
Cross-Site Scripting (XSS)
Insecure Direct Object Reference
Security Misconfiguration
Sensitive Data Exposure
Missing Function Level Access Control
Cross-Site Request Forgery (CSRF)
Using Components With Know Vulnerabilities
Unvalidated Redirects And Forwards
From the OWASP Top 10 2021 (web) and OWASP Code Review Guide, we focus more on Injections categories.

Fig 2: Existing standard for web application security project and code review guideline.

Table 1 discusses the type of injection attacks in web applications based on OWASP Top 10 2021 for web applications [10], such as SQL injection, Cross-Site Scripting (XSS), command injection, XML External Entity (XXE) injection, and Server-Side Request Forgery (SSRF).

Table 1: type of injection attacks based on Owasp top 10 2021

Injection attacks	**Description**
SQL Injection	This attack happens when a hacker manipulates SQL queries to get access to the application's database and executes unauthorized instructions or retrieves data.
Cross-Site Scripting (XSS)	Though not technically an injection attack, cross-site scripting (XSS) entails inserting malicious scripts into user-viewed web pages, giving attackers the ability to steal confidential data or carry out unwanted operations on the victim's behalf.
Command Injection	By inserting malicious commands into weak system command calls, command injection attacks give an attacker the ability to potentially run any command on the host operating system.
XML External Entity (XXE) Injection	XXE injection attacks exploit vulnerable XML parsers, enabling attackers to read sensitive data, perform SSRF attacks, or execute arbitrary code.
Server-Side Request Forgery (SSRF)	SSRF attacks occur when an attacker can manipulate the application to request unintended internal or external resources, potentially bypassing access controls and retrieving sensitive information.

Table 2 illustrates the summary of the literature review for this research, including the purpose of each existing research study and the discussion or limitation.

Table 2: The Summary of previous work

No	Title, Year	Propose	Discussion or Limitation
1	Main paper: Vulnerability Assessment and Penetration Testing Approach Towards Cloud-Based Application and Related Services, 2021 [3]	Discussed the real-world cloud attack against each type of cloud service model and mentioned the techniques hackers use to perform those attacks. Examples of attacks on the Cloud mentioned in this paper are access control weaknesses, **XSS, SQL injection**, Insecure storage, insecure settings, cookie manipulation, hidden data field manipulation, and Cross-Site Request Forgery (CSRF).	Not discuss in detail how to perform VAPT but only focus on the type of attack against each cloud service model.
2	A SQL Injection Detection Method based on Adaptive Deep Forest, 2019 [12]	Suggests using an adaptive deep forest technique to find sophisticated SQL injection attacks. Compared to traditional machine learning techniques, ADF allows the model parameters to be automatically changed during training, increasing detection accuracy.	From these papers, we will use as our basic techniques to conduct the injection attacks (SQL injection and Cross-Site-Scripting)
3	Cross-Site Scripting (XSS) Attacks And Mitigation: A Survey, 2019 [13]	This paper analyzes the current trend and various patents of the XSS payload crafted by the attacker from multiple resources such as Xbuster, snort, Webkit XSS Auditor, and many more.	
4	Exploring Cross-Site Scripting (XSS): Attack Payloads, Prevention, and Mitigation Techniques, 2022 [14]	This paper discusses the type of XSS attack, detection techniques such as static analysis and dynamic analysis, the impact of XSS, and prevention techniques like input validation and output encoding.	
5	An Analysis of XSS Vulnerabilities and Prevention of XSS Attacks in Web Applications , 2023 [15]	This paper discusses the type of XSS attack, the impact of XSS, and how to prevent XSS.	Not discuss in detail how to perform VAPT but only focus on how to mitigate this issue.
6	Testing Approaches and Tools for AWS Lambda Serverless-Based Applications, 2022 [19]	The testing strategies and resources for serverless apps based on AWS Lambda are the main topics of this research study. With the help of a full-stack application, the authors hope to delineate testing procedures for	The paper highlights the limited testing practices and patterns available for serverless-based applications.

No	Title, Year	Propose	Discussion or Limitation
		serverless functionalities developed on the Amazon Web Services (AWS) cloud platform.	
7	NoSQL Racket: A Testing Tool for Detecting NoSQL Injection Attacks in Web Applications, 2017 [23]	A testing tool called 'NoSQL Racket' is proposed to detect NoSQL injection attacks in web applications. To look for any fraudulent query segments, the tool compares the NoSQL query structure in the runtime query statement with the code query statement.	The tool was tested on four vulnerable web applications and compared against three other well-known testers, and it was found to be effective in detecting NoSQL injection attacks.

However, existing VAPT methodologies often need more comprehensive coverage for injection vulnerabilities in cloud-based web applications [3], and they may not consider the unique characteristics and challenges associated with the AWS environment related to the serverless technologies such as AWS Lambda and Amazon API Gateway. [19]. This research addresses this gap by proposing an advanced VAPT framework focusing specifically on injection attacks in web applications deployed on AWS.

There are several VAPT approaches that can be used, depending on the specific needs of the organization or software being assessed. Here are some of the common approaches [16].

Software testing using "Black Box" testing involves the pentester not knowing anything about the internal workings of the program being tested. The focus of this testing method is on the system's inputs and outputs to ensure that they function correctly. The pentester examines the software from an external perspective, treating it as a "Black Box" where the internal workings are unknown. This approach allows for a thorough evaluation of the software's functionality and behavior without any bias or preconceived notions about how it should work [16].

White Box Testing, on the other hand, gives the pentester full access to the software's internal workings. This means that the pentester can examine the code and the software's structure to ensure that it is working correctly. This testing method is used to verify the software's logical flow and to ensure that all code paths are executed correctly. By having access to the internal workings of the software, the pentester can identify any potential vulnerabilities or weaknesses that may exist [16].

Gray Box Testing combines elements of both black box and white box testing techniques. In this approach, the pentester has some knowledge of the internal workings of the software being tested but does not have full access to the code. This testing method is used to verify that the software is working correctly and to identify any potential security vulnerabilities. By having partial knowledge of the internal workings, the pentester can focus on specific areas of concern while still maintaining an external perspective. This allows for a more targeted and efficient testing process [16].

Dynamic Testing involves executing the software application and observing its behavior in different scenarios. This testing method focuses on the functionality and behavior of the software in a runtime environment. Examples of dynamic testing include unit testing, integration testing, software testing, and acceptance testing. Through the process of software execution and behavior analysis, the pentester is able to find any problems or bugs that might occur during real-world use. This kind of testing is necessary to make sure the program works as intended and satisfies end users' needs. [17]

Static Testing, on the other hand, involves testing the code without executing it. This can include manual reviews, automated analysis, and code inspection to identify syntax errors, code vulnerabilities, and logic errors. Examples of static testing include code reviews, code inspections, and static code analysis. By analyzing the code itself, the pentester can identify any potential issues or weaknesses that may exist. This testing method is crucial for identifying and addressing any coding errors or vulnerabilities before the software is deployed [17]

3. Methodology

The process flow of this proposed framework development involves three (3) stages of the process, as discussed in Fig. 3, and the details for the process will be elaborated in the next paragraph.

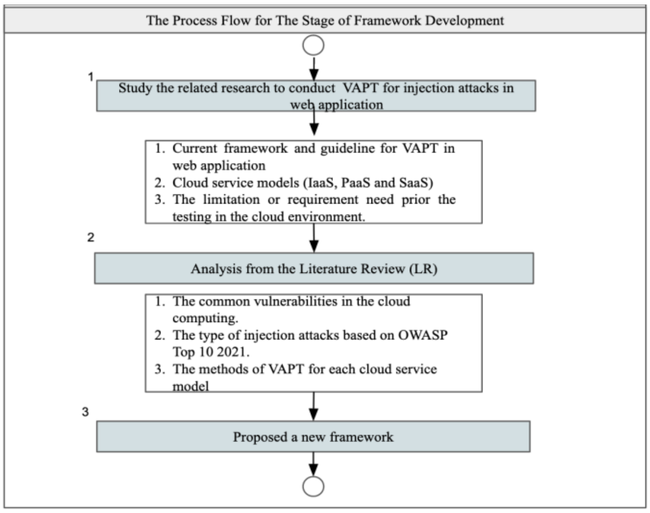
The Process Flow for The Stage of Framework Development
1
Study the related research to conduct VAPT for injection attacks in web application
1. Current framework and guideline for VAPT in web application
2. Cloud service models (IaaS, PaaS and SaaS)
3. The limitation or requirement need prior the testing in the cloud environment.
2
Analysis from the Literature Review (LR)
1. The common vulnerabilities in the cloud computing.
2. The type of injection attacks based on OWASP Top 10 2021.
3. The methods of VAPT for each cloud service model
3
Proposed a new framework

Fig 3: The overview of process flow for each stage of framework development

The initial step in each stage of framework development involves conducting a thorough study of relevant research to perform VAPT for injection attacks in web applications. This includes gathering related information, researching and studying existing frameworks or guidelines for testing web applications, identifying the cloud user and the roles of the penetration tester, and addressing any limitations or requirements that are needed before conducting the testing in the cloud environment.

The next process is analyzing the literature review, which involves identifying common vulnerabilities in cloud computing, studying the types of injection attacks, and determining the necessary tools and approaches for VAPT. The literature review analyzed common vulnerabilities in cloud computing, including gathering information on reported issues and vulnerabilities for each type of cloud service model, determining pre-engagement requirements for VAPT for each cloud service model, and identifying tools to simulate vulnerabilities. During this phase, we also focused on studying injection attacks such as SQL injection and XSS attacks based on the OWASP Top 10 2021. Then the approach to conducting VAPT is to confirm the specific needs for VAPT, identify the appropriate penetration testing methodology before the VAPT is conducted, such as the cloud service model in use, and obtain proper permission before starting the testing.

Finally, from the previous stage, a framework is proposed for conducting VAPT in web applications hosted in IaaS service models.

4. Proposed Framework for VAPT in Cloud

This section will briefly explain the proposed VAPT framework in the cloud computing environment for each type of cloud service model. Fig. 4 illustrates the proposed framework to conduct VAPT in the cloud computing environment for identified cloud service models such as IaaS, SaaS, and PaaS based on each type of cloud user. The pen-testers or external parties' roles and responsibilities depend on who engages with them.

For example, suppose the subscriber hires the pentester to test the IaaS model. In that case, all the testing components can be tested by the pentester (but the scope of testing needs to be determined during the pre-engagement between the pentester and subscriber). However, if the cloud provider hires the pentester to conduct the testing on the SaaS model, they cannot test all the testing components except for APIs only.

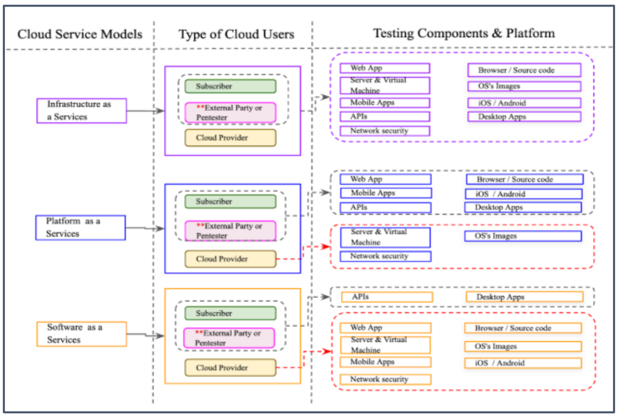

Fig. 4: The overall of the proposed framework for VAPT in the Cloud Computing environment

However, for this research, we will focus more on one (1) of the testing components: the web application. We chose this testing component since it faces public access and thus is more vulnerable to compromise by malicious users. Some vulnerabilities discovered or exploited in the web application layer might be a pivot point to other components. Fig. 5 illustrates the proposed framework to conduct VAPT in the cloud computing environment for web applications deployed in IaaS.

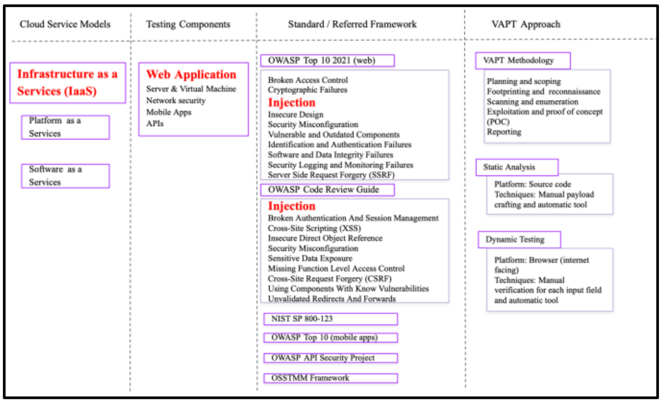
Cloud Service Models
Testing Components
Standard / Referred Framework
VAPT Approach
Infrastructure as a Services (IaaS)
Platform as a Services
Software as a Services
Web Application
Server & Virtual Machine
Network security
Mobile Apps
APIs
OWASP Top 10 2021 (web)
Broken Access Control
Cryptographic Failures
Injection
Insecure Design
Security Misconfiguration
Vulnerable and Outdated Components
Identification and Authentication Failures
Software and Data Integrity Failures
Security Logging and Monitoring Failures
Server Side Request Forgery (SSRF)
OWASP Code Review Guide
Injection
Broken Authentication And Session Management
Cross-Site Scripting (XSS)
Insecure Direct Object Reference
Security Misconfiguration
Sensitive Data Exposure
Missing Function Level Access Control
Cross-Site Request Forgery (CSRF)
Using Components With Know Vulnerabilities
Unvalidated Redirects And Forwards
NIST SP 800-123
OWASP Top 10 (mobile apps)
OWASP API Security Project
OSSTMM Framework
VAPT Methodology
Planning and scoping
Footprinting and reconnaissance
Scanning and enumeration
Exploitation and proof of concept (POC)
Reporting
Static Analysis
Platform: Source code
Techniques: Manual payload crafting and automatic tool
Dynamic Testing
Platform: Browser (internet facing)
Techniques: Manual verification for each input field and automatic tool

Fig. 5: The Framework to conduct VAPT in web applications deployed in IaaS.

5. Result and Discussion

This section will discuss in detail the proposed framework of VAPT in the cloud computing environment for web applications deployed in the IaaS. Throughout this section, we will state the basic penetration testing often used by the penetration tester to conduct the testing, then we will discuss the penetration testing methodology for conducting VAPT for web application.

The basic penetration testing methodology consists of several stages [20][21][22]. The first stage is planning and scoping, which involves establishing the scope of the testing and obtaining permission from the target's owner. This stage also includes signing any necessary agreements, such as a Non-Disclosure Agreement (NDA).

The second stage is footprinting and reconnaissance, which is the process of gathering information about the target. This can be done passively using search engines like Google or Bing or actively using tools like nslookup, dig, and whois.

The third stage is scanning and enumeration, where the penetration tester uses appropriate tools to scan for vulnerabilities and flaws within the target. This stage also involves gathering as much information as possible about the target.

The next methodology for basic penetration testing involves two key steps: exploiting and verifying. Each test case must undergo manual verification and exploitation based on the results from previous steps to ensure that there are no false positives or negatives. Manual testing is essential, particularly for vulnerabilities related to logic functions or process flow that cannot be detected by automatic tools.

The final step in the methodology is reporting. The penetration tester must provide a detailed report that outlines the testing performed and how all issues were discovered. The report should be replicable, allowing the project owner to reproduce the testing steps as outlined by the penetration tester.

Overall, the basic penetration testing methodology involves planning and scoping, followed by footprinting and reconnaissance, scanning and enumeration, then a thorough process of exploiting and verifying, and finally reporting. Each stage plays a crucial role in identifying and assessing vulnerabilities within the target; manual testing is crucial for detecting certain vulnerabilities; and the report must be detailed and replicable to ensure that all issues are properly addressed.

Next, we give more detail about the penetration testing methodology for the specific testing component for this research web application and the activities involved in each methodology phase, from planning to report delivery by pen-testers to their clients.

Planning and scoping involve determining the scope of testing and the objectives of the engagement. This includes identifying the reasons for conducting the testing, such as recent attacks or defacements, new web application development, or significant enhancements to the current web application deployed in the IaaS environment. Additionally, it is important to consider the VAPT approaches that will be used, such as black-box testing, white-box testing, gray-box testing, dynamic testing, or static analysis. These approaches need to be approved by the respective person or authority to ensure their effectiveness and alignment with the testing objectives.

In order to conduct VAPT, it is crucial to identify and confirm the target that will undergo the testing activities. This is typically done by the subscriber or client who demonstrates the target to the penetration tester. This demonstration ensures that the scope of the testing aligns with the expectations of the subscriber or client. Furthermore, it is necessary to prepare specific requirements for the testing, such as providing all the IP addresses, endpoints, or domains that are included in the scope. If static analysis is part of the testing, the source code of the application needs to be provided. Additionally, any limitations or special requirements, such as conducting the testing during office hours or outside business hours, should be clarified and agreed upon by both parties involved.

Another important aspect of planning and scoping is understanding the architecture and components of the application. This includes analyzing the backend, frontend, and any AWS services that are being used. By gaining a comprehensive understanding of the application's architecture, the penetration tester can effectively plan and execute the testing activities. This knowledge allows for a more targeted approach, ensuring that all relevant components are thoroughly tested for vulnerabilities.

In the process of footprinting and reconnaissance, the penetration tester gathers all the necessary information about the target. This includes conducting static analysis if the source code is provided, in order to identify any potential vulnerable areas at the source code level. This step is crucial in identifying potential weaknesses and vulnerabilities that can be exploited during the testing process. By conducting thorough reconnaissance, the penetration tester can gather valuable information that will inform the subsequent testing activities and help ensure a comprehensive and effective VAPT process.

During the scanning and enumeration stage of the Vulnerability Assessment and Penetration Testing (VAPT) methodology, the penetration tester aims to identify any potential vulnerabilities in the target system, particularly in fields that require user input. To achieve this, the penetration tester utilizes various scanning tools to thoroughly examine the target and gather as much information as possible. By employing a combination of scanning tools, the penetration tester can effectively identify all potential injection points and enumerate the files present in the web server. This comprehensive approach ensures that no vulnerabilities are overlooked and provides a solid foundation for the subsequent stages of the VAPT process.

Once the scanning and enumeration phase is complete, the penetration tester proceeds to the exploiting and verifying stage. At this point, all the findings or security issues that were discovered during the testing process must be manually verified to ensure their accuracy. This verification process is crucial in order to eliminate any false positives and ensure that the reported vulnerabilities are indeed valid. Additionally, the penetration tester is responsible for providing evidence of the reported findings or security issues, demonstrating how the system or target can be compromised. This evidence serves to validate the identified vulnerabilities and helps the project owner understand the potential impact of these issues.

The final stage of the VAPT methodology is reporting. In this stage, the penetration tester prepares a detailed report that documents the testing process and the discovered vulnerabilities. The report should provide a comprehensive account of how the penetration tester identified the issues, including the steps taken and the tools used. This level of detail allows the project owner to replicate or reproduce the testing steps outlined in the report, ensuring that the vulnerabilities can be properly addressed. The report serves as a valuable resource for the project owner, providing them with the necessary information to mitigate the identified vulnerabilities and improve the overall security of the system.

6. Conclusion

In conclusion, from this research, we proposed a new framework and methodology of VAPT for injection attacks in web applications deployed in the IaaS platform. We chose this cloud user due to the nature of the responsibilities of penetration testers who will perform the VAPT according to who is engaged with them and the web application testing component since it is facing a public user (all the authorized users can view the web application). This framework also discusses the type of injection attack and helpful methodology to reproduce the injection attack in web applications hosted in the cloud environment or a physical server.

For future research, we could do simulation testing to ensure the effectiveness of this proposed framework of VAPT and discover the injection points in the cloud computing environment. The simulation testing should be done in a closed environment since we will deploy vulnerable machines in the Cloud before we can use the same testing method in the production environment.

Acknowledgement

This work was funded by the Private Partnership Research Grant (PPRG) sponsored by Universiti Malaysia Terengganu and Cybersecurity Malaysia (Vot 53389), (UMT/PPRG/2020/53389). The authors sincerely thank the Research Management Office, Universiti Malaysia Terengganu for the financial support.

References

[1] SANS GPEN 2013 (Page 10, 560.1)

[2] Soon Bock, Loh. "4 Reasons Why Penetration Testing Is Important." Horangi Cyber Security, https://www.horangi.com/blog/4-reasons-why-penetration-testing-is-important.

[3] International Journal of Scientific Research in Science, E., & IJSRSET, T. (2021). Vulnerability Assessment and Penetration Testing Approach Towards Cloud-Based Application and Related Services. International Journal of Scientific Research in Science, Engineering and Technology. https://doi.org/10.32628/IJSRSET218346

[4] "NIST SP 800-145, The NIST Definition of Cloud Computing." NIST Technical Series Publications, https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf.

[5] "An overview of cloud security." What is cloud security, IBM, https://www.ibm.com/cloud/learn/cloud-security.

[6] "7 Cloud Computing Security Vulnerabilities and What to Do About Them". https://towardsdatascience.com/7-cloud-computing-security-vulnerabilities-and-what-to-do-about-them-e061bbe0faee

[7] "How Does Cloud Penetration Testing Differ from Standard Penetration Testing? "https://www.guidepointsecurity.com/education-center/cloud-penetration-testing/

[8] "Cloud Application Security Checklist And Best Practices". https://www.rishabhsoft.com/blog/cloud-application-security-best-practices.

[9] "Cloud security" https://www.ibm.com/cloud/learn/cloud-security

[10] OWASP Top 10 https://owasp.org/www-project-top-ten/

[11] OWASP Code Review Guide 2.0 "https://owasp.org/www-pdf-archive/OWASP_Code_Review_Guide_v2.pdf"

[12] Qi Li; Weishi Li; Junfeng Wang; Mingyu Cheng "A SQL Injection Detection Method based on Adaptive Deep Forest" https://ieeexplore.ieee.org/document/8854182

[13] German Rodriguez "Cross-Site Scripting (XSS) Attacks And Mitigation: A Survey" https://www.sciencedirect.com/science/article/abs/pii/S1389128619311247

[14] Joshua Crotts "Exploring Cross-Site Scripting (XSS): Attack Payloads, Prevention, and Mitigation Techniques"https://www.researchgate.net/publication/360400563_Exploring_Cross-

Site_Scripting_XSS_Attack_Payloads_Prevention_and_Mitigation_Techniques.

[15] Hansaka Dilshan Jayawardana "An Analysis of XSS Vulnerabilities and Prevention of XSS Attacks in Web Applications" https://www.researchgate.net/publication/371724261_An_Analysis_of_XSS_Vulnerabilities_and_Prevention_of_XSS_Attacks_in_Web_Applications

[16] Jason Firch "What Are The Different Types Of Penetration Testing? "https://purplesec.us/types-penetration-testing/

[17] Neil Dupaul "Static testing vs Dynamic" https://www.veracode.com/blog/secure-development/static-testing-vs-dynamic-testing

[18] Eetu Rinta-Jaskari "Testing Approaches And Tools For AWS Lambda Serverless-Based Applications" https://www.researchgate.net/publication/358138859_Testing_Approaches_And_Tools_For_AWS_Lambda_Serverless-Based_Applications

[19] Tori Thurmond "What Are the Penetration Testing Steps?" https://kirkpatrickprice.com/blog/7-stages-of-penetration-testing/

[20] EC-Council "Understanding the Five Phases of the Penetration Testing Process"https://www.eccouncil.org/cybersecurity-exchange/penetration-testing/penetration-testing-phases/

[21] PCI Data Security Standard (PCI DSS) version 1.0 "Information Supplement: Penetration Testing Guidance "https://listings.pcisecuritystandards.org/documents/Penetration_Testing_Guidance_March_2015.pdf

[22] Ahmed M. Eassa "NoSQL Racket: A Testing Tool for Detecting NoSQL Injection Attacks in Web Applications" https://thesai.org/Downloads/Volume8No11/Paper_78-NoSQL_Racket_A_Testing_Tool.pdf

[23] Ms. Disha H. Parekh "An Analysis of Security Challenges in Cloud Computing"https://thesai.org/Downloads/Volume4No1/Paper_6-An_Analysis_of_Security_Challenges_in_Cloud_Computing.pdf