International Journal of Advances in Engineering and Management (IJAEM)
Volume 2, Issue 9, pp: 389-394 www.ijaem.net ISSN: 2395-5252

Web Application Security Using VAPT

JatinKushwah, Kushagra Dutt Sharma, Raj Jhunjhunwala, Tanisha Duggal

Department of Computer Science and Engineering, Dr Akhilesh Das Gupta Institute of Technology and Management, New Delhi

----
Date of Submission: 10-11-2020 Date of Acceptance: 25-11-2020

ABSTRACT: By taking advantage of a vulnerability, Cybercriminals are easily able to steal confidential data of the ICT, resulting in heavy loss. Vulnerability Assessment and penetration testing is a special technique to get rid of various security threats from web applications. By focusing on high-risk vulnerabilities such as SQL Injection, Cross-Site Scripting, Local File Inclusion and Remote File Inclusion, in this paper, we have examined the general mechanics of the VAPT process and gather tools which can be useful during the VAPT process.

I. INTRODUCTION

The web application can be harmed by various logical and technical vulnerabilities. SQL injection, cross-site scripting, remote file inclusion and local inclusion are some of the examples of technical vulnerabilities. These are the crucial vulnerabilities that affect the security of web applications. A vulnerability can occur due to various reasons such as poor programming, or due to an outdated system. Web applications can be secured by various methods, wherein Vulnerability Assessment and Penetration Testing (VAPT) process is a special approach. This approach audits web application security and can also be used to provide security to the associated layers. VAPT includes an audit system for finding vulnerabilities, which might exist in the system, and exploit those vulnerabilities in the same manner as an attacker and produces data which describes the risk level of the system.

With the purpose of diving a little deeper in the area of vulnerability assessment and penetration in this paper, we have analysed an overview of the penetration testing process and its limitations and included various tools which are helpful in conducting the VAPT process of high-risk vulnerabilities. The paper is presented as follows: Section 2 provides background information on our study, section 3 describes literature survey, section 4 gives an overview of the VAPT process, section 5 states the tools used in VAPT, section 6 summarizes our research and proposes the future work.

II. BACKGROUND

The research regards SQL Injection, Cross-Site Scripting, Remote File Inclusion and Local file inclusion as high-risk vulnerabilities. OWASP also included these threats in Top 10 High Risk of a web application. These technical vulnerabilities occurred when the web application processes data without proper filtration or validation.

2.1. SQL Injection

SQL Injection vulnerability may affect dynamic web applications which stored data in the associated database. Through SQL Injection, attackers pass malicious code to SQL Server through inserting it in the strings. This malicious code is commonly known as payloads that instruct the database server to retrieve specific information from the database. By taking advantage of an SQL injection vulnerability, an attacker can download the entire database on his computer machine and enumerate important information such as database version, database user name, table information and sensitive data available in database columns such as password, username etc. In several cases, attackers can perform various operations such as add, modify and delete records in a database or an attacker may be able to execute system-level commands and can be successful to launch additional attacks such as the denial of service. These additional attacks are dependent on the role and privilege set in the SQL server of the target machine.

2.2. Cross-Site Scripting

Cross-site scripting also known as XSS is scripting attack in which attacker injects or executes code through the browser at user side for the purpose to steal information of the user's credential. Attackers attempt to steal the user's credential through the vulnerable web application by executing the payloads at client side. The typical example is one where the attacker may inject the payload to the vulnerable field of a web application, and when the user visits the page, at that time, the payload placed in the page steals the user's cookies and sends it to the attacker, or may redirect the users to phishing sites. There are three types of XSS attacks known which are persistent, not persistent and DOM-based cross-site scripting.

In non-persistent XSS also refer to reflected XSS, in which an attacker crafts malicious URL and then tries to execute in the user's browser to steal the data or may redirect users to a phishing page. Persistent XSS is a more powerful attack in which code injected by an attacker is stored in secondary storage such as databases. DOM-based XSS occurs when an application accesses the user's information and writes it in HTML format. This type of vulnerability is commonly seen in RSS feed.

2.3. Local File Inclusion

Local File Inclusion (also known as LFI) is the high-risk web application vulnerability as it affects widely to an application. Generally, this vulnerability occurs due to the inputs not being properly sanitized. An attacker might use an LFI attack to access different files and gather confidential information, thus harvesting useful information. LFI vulnerability also leverages an attacker to place a backdoor (A Shell) in the target server through a vulnerable web application. Besides, an attacker may remotely execute commands by combining this vulnerability with some other attack vectors, such as file upload vulnerability or log injection.

2.4. Remote File Inclusion

Remote File Inclusion (also known as RFI) is a very critical vulnerability. By leveraging this vulnerability, attackers can execute backdoor programs on the server through a vulnerable web application. Thus, an attacker can retrieve confidential information through the backdoor. RFI and LFI vulnerabilities are very similar to each other with the difference being that LFI provides an opportunity for an attacker to directly place a backdoor in the target server while in RFI, the attacker uses a remote location to execute and retrieve the backdoor.

2.5. Web Application Security

There are various approaches available to resolve vulnerabilities in web applications, such as code review, secure coding practices, web application firewall. All these techniques are providing an option to secure the web application at each phrase since the development of deployment of a web application.

Nowadays due to these multiple security options, these technical vulnerabilities can be eliminated from the web application. However, each security approach has its advantages and limitations. For instance, the secure coding approach required additional knowledge of secure programming. In addition to another approach, VAPT can also be used as a specialized approach to a secure web application. VAPT is a single test process. VAPT performs a more comprehensive test on the entire system.

This test provides more detailed information about the risk level of a web application. The digital infrastructure of an organisation is protected by scanning all parts of ICT infrastructure in the VAPT process. As an on-demand solution, VAPT can be carried out from anywhere through the internet, anytime. VAPT is considered as a hybrid solution because this process is performed by the external security expert who uses automated tools and techniques to test the system. Thus, VAPT is a convenient approach as it can be used to secure a web application at any time. In the domain of web application security using VAPT, there have been various researches, with several more undergoing in both academic institutes, as well as in industries since a couple of years. We have surveyed numerous papers and researches which give an overview of VAPT process in the context of web applications to identify the general mechanism of VAPT process and extract some useful free, open-source tools and methods which can be used in VAPT process to eliminate SQLI, XSS, RFI and LFI vulnerabilities.

III. LITERATURE SURVEY

Various models, techniques and tools are available to perform penetration testing and to check for SQL Injection, Cross-Site Scripting, Local File Inclusion and Remote File Inclusion vulnerabilities of websites. The following section illustrates the related work through models, techniques and tools: In 2014, Sugandh Shah1 et al, B. M. Mhetre2 et al create an automated VAPT Testing Tool- NetNirikshak 1.0 at IDRBT, which comprises 8 different modules. This makes it effective in conducting the VAPT process by assessing Services and analysing Security Posture.

It works based on services running in the target system and identifies the vulnerabilities. The tool is capable of detecting SQL injection vulnerability and it reports all the identified vulnerable links. The tool also contains the exploitation process which exploits the vulnerability automatically and steals confidential data from the target system by exploiting SQL injection vulnerability. This tool smartly sends all the findings through a write-protected pdf report to a specified email and removes the copies from the hard disk for the purpose of maintaining confidentiality in the VAPT process.

In 2015, Insha Altaf1 et al Firdous ul Rashid2 et al, Jawad Ahmad Dar3 et al Mohd. Rafiq4 et al talked about various SQL injection methods using the principles of automated testing. They have explained various reasons for conducting the VAPT process and explained Authentication bypass, Union based SQL Injection, Firewall Bypassing attacks mechanics and its patching techniques. A part of their research study includes the explanation for the working procedure of Acunetix Vulnerability Scanner.

In the same year, Jai Narayan Goel1 et al, BM Mehtre2 et al described the entire process to use the VAPT process and demonstrated how it can be used for cybersecurity. They have described the complete life cycle of the VAPT process. They have included the top 15 VAPT tools, both open-source and ones commercially available in the market with their usage and operating system compatibility which can be useful in assessment and exploitation during the VAPT process. They have concluded the necessity to increase the use of VAPT.

In 2016, Kamran Shaukat1 et al, Amber Faisal2 et al, Rabia Masood3 et al, Ayesha Usman4 et al, Usman Shaukat5 et al scrutinised different frameworks that can be secure during the testing level. To secure databases, networks, web applications and Android, they have proposed an entrance testing technique. In the context of entrance testing during their technical review, they identified that studied approaches useful to particular frameworks and not each strategy can be applied to single framework and critics as neglectable. To defeat these issues, they have attempted and proposed another methodology and explained all the components.

During this year, Jai Narayan Goel1 et al, Mohsen Hallaj2 Asghar et al3, Vivek Kumar4 et al, Sudhir Kumar Pandey5 et al propose an Ensemble approach with various VAPT tools which reliable in the prediction of vulnerability for decrease false positive. They have also implemented their study and made a software based on their approach called “VEnsemble 1.0” which is capable of use with several other both open source and commercial tools and included the results.

In the same year, Prashant S. Shinde1 et al, Shrikant B. Ardhapurkar2 et al explained clearly of various techniques used in vulnerability assessment and penetration testing (VAPT). Also, pay attention to cybersecurity awareness and importance in an organization to stay safe. they conclude that there are various tools available for VAPT, with new vulnerability evolution existing tools should be updated to identify new vulnerabilities and make them flexible so that new attack signatures can be added.

In 2017, S. Sandhya, Sohini Purkayastha, Emil Joshua, and Akash Deep, among others discussed the utilization of penetration testing approach using Wireshark tool and demonstrated a method. They have also surveyed several tools for penetration testing to solve security issues.

IV. OVERVIEW OF VAPT

Vulnerability assessment and penetration testing both contain execution of different processes aimed to secure a web application, however, both are closely related to each other. The vulnerability assessment process provides information on possible vulnerability while the penetration testing process includes exploiting the vulnerability to assume risk level. The overall mechanism of VAPT is illustrated in figure 1. as under.

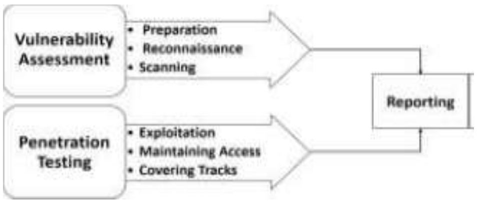

Fig. 1 - General VAPT process model

VAPT is a useful approach which can be used to secure web applications, there are however certain limitations associated with it. Herein we have discussed several key points in this regard. Testing is not a time-bound process. A limitation in time may reduce efficiency levels of the penetration testing process. The success is depending on the skill and efforts of the tester so the test does not guarantee to identify a vulnerability. Generally, VAPT is carried out by an external person or outside firm due to these factors raises the overall budget of the system as VAPT is a repetitive process. Upon change or modification in the system again penetration testing is required. VAPT processes may damage the system for example, during the process testers use various tools and techniques to scan the application which may affect bandwidth. VAPT process also contains legal limitations, for example, a web application hosted on a shared server cannot be penetrated without permission of the web server provider because testing may affect the performance of the overall server and cause damage to another web application which is hosted on the same server. Despite several limitations, VAPT is a far beneficial approach to secure a web application. Various tools and techniques can be used during the penetration testing process.

V. VAPT TOOLS

To support the VAPT process, there are several tools, both commercial and open-source, available to find SQLI, XSS, LFI and RFI vulnerabilities. In this section, we have explored several tools which are helpful in the VAPT process.

5.1. W3af

W3af is a free and open-source automated Black-box web application scanning tool containing various plugins. This tool is available in both GUI and command-line user interface, capable of assessing a web application for a range of vulnerabilities and able to exploit it. Plugins are interlay connected which share information. Special Issue based on proceedings of 4TH International Conference on Cyber Security (ICCS) 2018 INTERNATIONAL JOURNAL OF ADVANCED STUDIES OF SCIENTIFIC RESEARCH (IJASSR) ISSN 2460 4010 ABSTRACTED & INDEXED IN ELSEVIER-SSRN 162

5.2. Fimap

Fimap is written in python language and a command-line security tool helps to find and exploit LFI and RFI vulnerability from a web application. (Joe Beauchamp, 2016).

5.3. Metasploit

Metasploit is available in both commercial and open source platforms. This tool includes a large database of various exploits and methods which provides a smart testing platform to the penetration tester. The tool support in extensive security auditing of a web application.

5.4. Acunetix

Acunetix is a commercial web vulnerability scanner capable of scanning a web application with a black-box approach. It is an automated tools GUI tool which scans a web application for several different vulnerabilities. The tool is to produce a user-friendly and professional report of scanning.

5.5. Nexpose

Nexpose is a GUI and automated vulnerability assessment tool available in both commercial and free versions where the exposed community version is available for a year with limited functionality while the commercial version of exposure is equipped with full functionality.

5.6. Nessus

Nessus is a remote scanner which scans external vulnerabilities, hosted offsite. It also scans all external-forcing ports remotely, and looks for any/all sorts of communication with botnet-infected systems, as well as potentials from external sources.

5.7. Nikto

Nikto is a free software command-line vulnerability scanner that scans webservers for dangerous files/CGIs, outdated server software and other problems. It performs generic and server type specific checks. It also captures and prints any cookies received.

VI. CONCLUSION AND FUTURE WORK

VAPT is a very compressive process. Various research and methods are introduced by researchers to support the VAPT process, we have gone through a literature survey and analysed the overview of the VAPT process and identified several limitations associated with the process. We have also discussed several tools which can be helpful to conduct the VAPT process to find SQLI, XSS, LFI and RFI vulnerabilities. We conclude that VAPT is a very important process that helps in identifying security defects. Many repositories in the form of tools, methods and mechanics available to support VAPT. In the future, we will more closely look for various problems associated with the VAPT process such as identify the factors which slow down businesses to adopt the VAPT process and continue our research and study in this area.

REFERENCES

[1]. Bharti Nagpal; Nanhay Singh; Naresh Chauhan; Angel Panesar. (2015). Tool based implementation of SQL injection for penetration testing. International Conference on Computing, Communication & Automation (IEEE Conference Publications), 746 - 749, DOI: 10.1109/CCAA.2015.7148509.

[2]. Dimitris Mitropoulos; Panagiotis Louridas; Michalis Polychronakis; Angelos D. Keromytis. (2017,). Defending Against Web Application Attacks: Approaches, Challenges and Implications. IEEE Transactions on Dependable and Secure Computing (IEEE Early Access Articles), Volume: PP (Issue: 99), 1 - 1, DOI: 10.1109/TDSC.2017.2665620.

[3]. Insha Altaf; Firdous ul Rashid; Jawad Ahmad Dar; Mohd. Rafiq. (2015). Vulnerability assessment and patching management. IEEE Conference Publications, International Conference on Soft Computing Techniques and Implementations (ICSCTI), 2015, Pages: 16 - 21. Faridabad, India.

[4]. Kamran Shaukat; Amber Faisal; Rabia Masood; Ayesha Usman; Usman Shaukat. (2016). Security quality assurance through penetration testing. 2016 19th International Multi-Topic Conference (INMIC) (IEEE Conference Publications), 1 - 6, DOI: 10.1109/INMIC.2016.7840115.

[5]. M. Ridwan Zalbina; Tri Wanda Septian; Deris Stiawan; Moh. Yazid Idris; Ahmad Heryanto; Rahmat Budiarto. (2017). Payload recognition and detection of a Cross-Site Scripting attack. 2017 2nd International Conference on Anti-Cyber Crimes (ICACC) (IEEE Conference Publications), 172 - 176, DOI: 10.1109/Anti-Cybercrime.2017.7905285.

[6]. Mir Saman Tajbakhsh; Jamshid Bagherzadeh. (2015). A sound framework for dynamic prevention of Local File Inclusion. 2015 7th Conference on Information and Knowledge Technology (IKT) (IEEE Conference Publications), 1 - 6, DOI: 10.1109/IKT.2015.7288798.

[7]. Rahul Johari; Pankaj Sharma. (2012). A Survey on Web Application Vulnerabilities (SQLIA, XSS) Exploitation and Security Engine for SQL Injection. 2012 International Conference on Special Issue based on proceedings of 4TH International Conference on Cyber Security (ICCS) 2018 INTERNATIONAL JOURNAL OF ADVANCED STUDIES OF SCIENTIFIC RESEARCH (IJASSR) ISSN 2460 4010 ABSTRACTED & INDEXED IN ELSEVIER-SSRN 163 Communication Systems and Network Technologies(IEEE Conference Publications), 453 - 458, DOI: 10.1109/CSNT.2012.104.

[8]. Sugandh Shah; B. M. Mehtre. (2014). An automated approach to Vulnerability Assessment and Penetration Testing using NetNirikshak 1.0. 2014 IEEE International Conference on Advanced Communications, Control and Computing Technologies(IEEE Conference Publications), 707 - 712, DOI: 10.1109/ICACCCT.2014.7019182.

[9]. Hossain Shahriar, Mohammad Zulkernine. (June 2012). Mitigating program security vulnerabilities: Approaches and challenges. ACM Computing Surveys (CSUR), Volume 44(Issue 3), Article No. 11. DOI:10.1145/2187671.2187673

[10]. Hugo F. González Robledo. (2008). Types of Hosts on a Remote File Inclusion (RFI) Botnet. Electronics, Robotics and Automotive Mechanics Conference, 2008. CERMA '08(IEEE Conference Publications), 105 - 109, DOI: 10.1109/CERMA.2008.60.

[11]. Jai Narayan Goel, B.M. Mehtre. (2015). Vulnerability Assessment & Penetration Testing as a Cyber Defence Technology. Procedia Computer Science, 57(http://www.sciencedirect.com/science/article/pii/S1877050 915019870), 710-715, ISSN 1877-0509, http://dx.doi.org/10.1016/j.procs.2015.07.458.

[12]. Jai Narayan Goel, Mohsen Hallaj Asghar, Vivek Kumar, Sudhir Kumar Pandey. (2016). Ensemble Based Approach to Increase Vulnerability. Innovation and Challenges in Cyber Security (ICICCS-INBUSH), 2016 International Conference on. DOI:10.1109/ICICCS.2016.7542303

[13]. Joe Beauchamp. (2016). VisualFI: File Inclusion Identification, Exploitation and Reporting Tool. England: The University of the West of England. Retrieved from http://www.cems.uwe.ac.uk/~palegg/teaching/fyp/projects/2016beauchamp.pdf

[14]. OWASP. (2016, April 26). Testing for SQL Injection (OTG-INPVAL005). (OWASP Foundation Inc) Retrieved Jun 6, 2017, from https://www.owasp.org/index.php/Testing_for_SQL_Injection _(OTG-INPVAL-005)

[15]. OWASP Testing Guide v2. (n.d.). Testing for Cross-site scripting. Retrieved May 5, 2017, from https://www.owasp.org/index.php/Testing_for_Cross_site_scripting

[16]. Prashant S. Shinde; Shrikant B. Ardhapurkar. (2016). Cybersecurity analysis using vulnerability assessment and penetration testing. 2016 World Conference on Futuristic Trends in Research and

Innovation for Social Welfare (Startup Conclave)(IEEE Conference Publications), 1-5, DOI:10.1109/STARTUP. 2016.7583912.

[17]. S Sandhya, Sohini Purkayastha, Emil Joshua, Akash Deep. (6-7 Jan. 2017). Assessment of Website Security by Penetration. Advanced Computing and Communication Systems (ICACCS), 2017 4th International Conference on.

AUTHORS

First Author – JatinKushwah, B.Tech. (Computer Science and Engineering), Dr Akhilesh Das Gupta Institute of Technology and Management, New Delhi

Second Author – KushagraDutt Sharma, B.Tech. (Computer Scienceand Engineering), Dr Akhilesh Das Gupta Institute of Technology and Management, New Delhi

Third Author – Raj Jhunjhunwala, B.Tech. (Computer Scienceand Engineering), Dr Akhilesh Das Gupta Institute of Technology and Management, New Delhi

Correspondence Author– Tanisha Duggal,B.Tech. (Computer Scienceand Engineering), Dr Akhilesh Das Gupta Institute of Technology and Management, New Delhi