IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661, p- ISSN: 2278-8727Volume 16, Issue 3, Ver. IV (May-Jun. 2014), PP 126-129
www.iosrjournals.org

Web Penetration Testing using Nessus and Metasploit Tool

[1]Indraneel Mukhopadhyay, [2]Shilpam Goswami, [3]Eshita Mandal
[1,2,3,] Institute of Engineering & Management, Y-12 Saltlake Electronics Complex, Sector V, Kolkata -91 INDIA

Abstract: *Web Penetration Testing is a tool that is being used widely to see how the website reacts when an vulnerability attack is done. Now days many ethical hackers use web penetration tool to predict the vulnerabilities of the website. We have done a survey of some of the web penetration tools that are available and then we have proposed a architecture using nesus and metasploit tool to do scan vulnerabilities of an website.*

General Terms*: Web Penetration Tool, Vulnerability Scanner, Security*
Keywords: *Web Penetration Tool, Vulnerability Scanner, Nessus, Metasploit, Kali Linux, ethical hacking*

I. Introduction:

Penetration Testing is a great way to identify vulnerabilities that exists in a system or network that has an existing security measures in place. A penetration test, or the short form pen test, is an attack on a computer system with the intention of finding security weaknesses, potentially gaining access to it, its functionality and data˙.A penetration test involves identifying the target systems and the goal, then reviewing the information available and undertaking available means to attain the goal. A penetration test will advise if a system is vulnerable to attack, if the defenses were sufficient and which defenses (if any) were defeated in the penetration test. Effective penetration tests will couple this information with an accurate assessment of the potential impacts to the organization and outline a range of technical and procedural countermeasures to reduce risks [1].The results of these tests or attacks are then documented and presented as report to the owner of the system and the vulnerabilities identified can then be resolved after the vulnerability scan of a host.

Penetration testing using commercially available automated tool scan help avoid such faults, but analysis of several popular testing tools reveals significant drawbacks in their performance. We surveyed on different types vulnerabilities by simulating attacks from malicious users on a target application. Identifying the input vectors of a web application and checking the results of an attack are important parts of penetration testing, which is called web penetration testing as they indicates where an attack could be introduced and whether an attempted attack was successful or not. Vulnerability assessments are necessary for discovering potential vulnerabilities throughout the environment. Full exploitation of systems and services is not generally in scope for a normal vulnerability assessment engagement. Systems are typically enumerated and evaluated for vulnerabilities, and testing can often be done with or without authentication. Most vulnerability management and scanning solutions provide actionable reports that detail mitigation strategies such as applying missing patches, or correcting insecure system configurations.

This paper has been divided into following sections. Section 2 deals with previous work that has happened, section 3 deals with the proposed architecture, next section deals with examples of different Web Penetration Testing tools. Section 5 deals with implementation details and we conclude the paper with future scope.

II. Previous Work:

In our survey, we found a tool which is used to find all the known vulnerabilities. The prototype implementation of our approach consists of three main components: the dynamic analysis module, which is an extension of the Python interpreter that collects traces of the executed application, the analyzer, which performs analysis thereof, and the penetration testing module that submits input data (both normal and malicious) to the web application. Number of reported web applications vulnerabilities is increasing dramatically. Most existing approaches are based on the Tainted Mode vulnerability model which cannot handle inter-module vulnerabilities [2].

In another paper two important steps of penetration testing are identifying the IVs of a web application and determining whether an attempted attack was successful. This paper proposed a new approach to penetration testing that improves both of these steps. The approach incorporates a conservative static analysis of the web application that identifies IVs directly from the application’s code. The approach improves the response analysis by leveraging automated dynamic analyses that accurately detect when an attack has succeeded. The authors have compared its performance against two state-of-the-art penetration testing tools and found the result satisfactory [3].

In another paper, we see that, each of the different penetration testing tools reported a different number of vulnerabilities. False positives were another issue. A vulnerability reported by a tool as a false positive if the team of security experts did not report it and a true positive if it matched one reported by the team. There was also coverage analysis. the number of vulnerabilities detected compared with the total number of actual vulnerabilities. So we are trying to develop our own penetration testing tool that can overcome the said difficulty [4].

III. Proposed Architecture:

The process of penetration testing can be broadly divided into three phases: information gathering, attack generation, and response analysis. Figure 1 shows a high-level overview of a generic penetration testing process. In the information gathering phase, testers use a wide variety of techniques, such as automated scanning, web crawlers, and social engineering, to gain information about the target application. This information is used to drive the attack generation phase, in which testers use the identified information, together with domain knowledge about possible vulnerabilities, to generate attacks. Penetration testers typically use a range of commercial and open-source tools to automate the generation of attacks. Finally, the response analysis phase checks whether an attack has succeeded and, if so, logs information about the attack. The final result of the penetration testing process is a report that details the discovered vulnerabilities and corresponding attacks. Developers can use this information to eliminate the vulnerabilities and improve the security of their software[5].

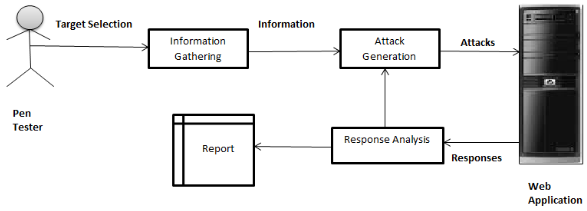


Figure1: Proposed Penetration Testing process

IV. Web Penetration Testing Tool:

For our proposed architecture we have worked with different off-the-shelf Web Penetration Testing tool like…

4.1 Skipfish: Skipfish is an active web application security reconnaissance tool. It prepares an interactive sitemap for the targeted site by carrying out a recursive crawl and dictionary-based probes. The resulting map is then annotated with the output from a number of active (but hopefully non-disruptive) security checks. The final report generated by the tool is meant to serve as a foundation for professional web application security assessments.

4.2 Wapiti: Wapiti allows you to audit the security of your web applications .It perform "black-box" scans, i.e. it does not study the source code of the application but it will scan the web pages of the deployed web app, looking for scripts and forms where it can inject data. Once it gets this list, Wapiti acts like a fuzzer, injecting payloads to see if a script is vulnerable.

4.3 Arachni: Arachni is a feature-full, modular, high-performance Ruby framework aimed towards helping penetration testers and administrators evaluate the security of web applications. Arachni is smart, it trains itself by learning from the HTTP responses it receives during the audit process. Unlike other scanners, Arachni takes into account the dynamic nature of web applications and can detect changes caused while travelling through the paths of a web application's cyclomatic complexity. This way attack/input vectors that would otherwise be undetectable by non-humans are seamlessly handled by Arachni. Finally, Arachni yields great performance due to its asynchronous HTTP model (courtesy of Typhoeus). Thus, you'll only be limited by the responsiveness of the server under audit and your available bandwidth.

4.4 Nessus:. Nessus is a network security scanner. It utilizes plug-ins, which are separate files, to handle the vulnerability checks. This makes it easy to install plug-ins and to see which plug-ins are installed to make sure that your are current. Nessus uses a server-client architecture. The main server will need to be built on a supported Unix-like operating system. The client is available for Unix, Linux, and Windows. The server is not an option because “it performs the security checks.[6]

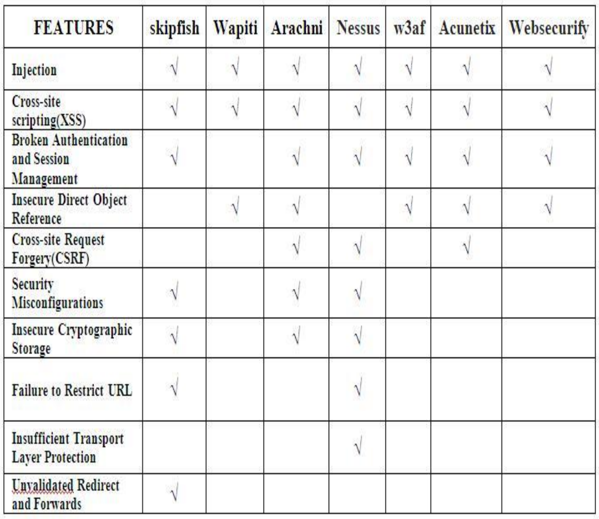
One of the most attractive features of Nessus is open source and many people contribute to Nessus everyday and that helps to keep it up-to-date. There will be plug-ins for new vulnerabilities within days of the vulnerabilities being released to the public. Another feature is that Nessus scans for vulnerabilities on Windows and Unix systems. This helps make it a good all-around tool so that you can scan a mixed environment in one session. Next, Nessus utilizes Nmap for port scanning. Nmap has become a standard in the Security Industry for good reasons. Nmap is known as an extremely fast port scanner.

4.5 w3af: w3af is a Web Application Attack and Audit Framework. The project's goal is to create a framework to find and exploit web application vulnerabilities that is easy to use and extend. To read our short and long term objectives, please click over the Project Objectives item in the main menu. This project is currently hosted at SourceForge, for further information, you may also want to visit w3af SourceForge project page..

4.6 Acunetix: Acunetix web vulnerability scanner is a tool designed to discover security holes in your web applications that an attacker would likely abuse to gain illicit access to your systems and data. It looks for multiple vulnerabilities including SQL injection, cross site scripting, and weak passwords. The application can be used to perform scanning for web and application vulnerabilities and to perform penetration testing against the identified issues. Mitigation suggestions are then provided for each weakness and can be used to increase the security of the web server or application being tested.

4.7 Websecurify: Web security is a powerful web application security testing platform designed from the ground up to provide the best combination of automatic and manual vulnerability testing technologies. It is available for all major desktop platforms including mobile devices and web.

Table 1 compares different Web Penetration Tools that are available with respect to different properties.



FEATURES	skipfish	Wapiti	Arachni	Nessus	w3af	Acunetix	Websecurify
Injection	√	√	√	√	√	√	√
Cross-site scripting(XSS)	√	√	√	√	√	√	√
Broken Authentication and Session Management	√		√	√	√	√	√
Insecure Direct Object Reference		√	√		√	√	√
Cross-site Request Forgery(CSRF)			√	√		√	
Security Misconfigurations	√		√	√			
Insecure Cryptographic Storage	√		√	√			
Failure to Restrict URL	√			√			
Insufficient Transport Layer Protection				√			
Unvalidated Redirect and Forwards	√						

Table1: Comparison of Few Web Penetration Testing Tools

V. Implementation Details:

To implement our proposed architecture our initial analysis we started using Nessus tool for vulnerability scanning. After the vulnerability analysis using Nessus we get the the vulnerability list then we attack the site using Metasploit tool. The Metasploit Framework is a tool that collectively combines exploits into one central location ideally for security researchers. Using Metasploit tool we can ethically hack a site. Equally important is the ability to correlate information provided by different types of requests Furthermore ,tools should be based on standardized and consistent procedures, implementing a well defined set of testing components to provide integrated support for detecting a maximum number of vulnerabilities and a minimal number of false positives. A generic penetration testing tool for Web services that combines all these attributes is a goal for the future. In our work we found vulnerability of a given website and attack to determine the vulnerability of the website.

VI. Future Scope:

Till now we have used Nessus and Metasploit tool to find out the vulnerability of a site. Our future scope will be to implement a tool that is combination of Nessus and Metasploit that can successfully find vulnerability of a site and ethically hack into any website.

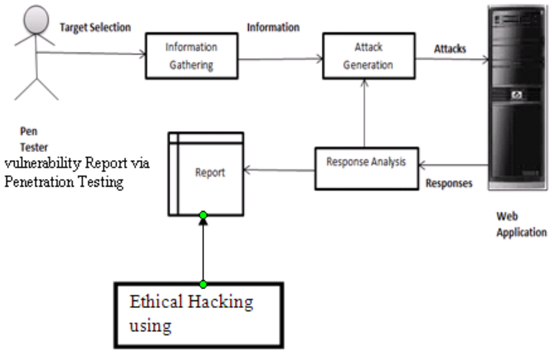


Figure 2: Penetration testing via Metasploit tool

References

[1]. Joseph Muniz Aamir Lakhani. "Web Penetration Testing with Kali Linux", PACKT publishing. September 2013.

[2]. Andrey Petukhov, Dmitry Kozlov "Detecting Security Vulnerabilities in Web Applications Using Dynamic Analysis with Penetration Testing", https://www.owasp.org/images/3/ 3e/OWASP-AppSecEU08-Petukhov.pdf

[3]. William G. J. Halfond1,Shauvik Roy Choudhary and Alessandro Orso "Improving penetration testing through static and dynamic Analysis". http://www-bcf.usc.edu/~halfond/ papers/halfond11stvr.pdf

[4]. Nuno Antunes and Marco Vieira, "Penetration Testing For Web Services", http://ieeexplore.ieee.org/stamp/stamp .jsp?tp=&arnumber=6681866

[5]. William G.J. Halfond, Shauvik Roy Choudhary, and Alessandro Orso "Penetration Testing with Improved Input Vector identification", http://www.cs.columbia.edu/~junfeng /reliable-software/papers/pen-test.pdf

[6]. Paul Schmelzel Nessus: "Vulnerability Scanning and Beyond" https://cyber-defense.sans.org/resources/papers/gsec/ nessus-vulerability -scanning-103152

[7]. Carlos Joshua Marquez."An Analysis of the IDS Penetration Tool: Metasploit", http://www.infosecwriters .com/text_resources/pdf/jmarquez_ Metasploit.pdf