Oct 28th, 9:00 AM - Oct 30th, 5:00 PM

Web application penetration testing

Besnik Qehaja
University for Business and Technology, besnik.qehaja@ubt-uni.net

Gazmend Krasniqi
University for Business and Technology, gazmend.krasniqi@ubt-uni.net

Ardian Bajraliu
University for Business and Technology, ardian.bajr@gmail.com

Amet Shabani
University for Business and Technology, ametshabani@gmail.com

WEB APPLICATION PENETRATION TESTING

Besnik Qehaja[1] , Gazmend Krasniqi[2], Ardian Bajraliu[3], Amet Shabani[4]

University for Business and Technology
{besnik.qehaja, gazmend.krasniqi}@ubt-uni.net,
{ardian.bajr, ametshabani}@gmail.com

Abstract. Safety of information is needed either in private sector or business for protection from market with competitive secrets or only for privacy. Advantages of internet and web applications is that they are accessible from everyone, but in business word data should be safe, reliable accessible. Although these are not new problems and always had different solutions to these problems, we always need to be on the cutting edge with new attacks that appear every day and to try to achieve a greater security. In this paper we present some of the most dangerous forms of risk which are risking web applications in year 2015/2016.we will demonstrate step by step how to achieve unauthorized access from web application inside server system and we will explain why is happened for our analysis that we have done. In testing stages we used some parts of real tests that we have done on several web applications, with Penetration Testing Methods which is procedure for testing and documentations including infrastructure of Networks, servers, Web applications, Wireless communications and all other technological parts. Penetration Testing is Testing Procedure for Web applications usually made on port 80 and 443.In this paper we will explain the real analyzing of tests with all the procedures for one web applications, including all the attached stages which are used in real life for testing the safety of web applications from safety testers.

Keywords: Security, Testing, Network Security , Web Applications.

1. Introduction

During the development of a web application, not all companies pay attention to proper safety key functions on which to concentrate the most delicate parts of a web application. This very important issue and may result in the total destruction of a company, so we have developed with great care these parts of a web application and security must always be the top priority. The following document will deal with all the testing procedures of some of membership by a web application include here all phases of construction attached which are used in the real lives of membership tested the security of web applications security testing

2. Steps in Penetration Testing

Information Gathering In this step, the testers collect as much information about the web application as possible and gain understanding of its logic. The deeper the testers understand the test target, the more successful the penetration testing will be [3]. The information gathered will be used to create a knowledge base to act upon in later steps. The testers should gather all information even if it seems useless and unrelated since no one knows at the outset what bits of information are needed. This step can be carried out in many different ways: by using public tools

such as search engines; using scanners; sending simple HTTP requests or specially crafted requests [4] Vulnerability analysis Step Using the knowledge collected from the information gathering step, the testers then scan the vulnerabilities that exist in the web application. The testers can conduct testing on configuration management, business logic, authentication, session management, authorization, data validation, denial of service, and web services [4]. In this step, web server vulnerabilities, authentication mechanism vulnerabilities, input-based vulnerabilities and function-specific vulnerabilities are examine[4]

2.1. Calculating risk

It is important to understand how to calculate risk associated with vulnerabilities found, so that a decision can be made on how to react. Most customers look to the CISSP triangle of CIA when determining the impact of risk. CIA is the confidentiality, integrity, and availability of a particular system or application. When determining the impact of risk, customers must look at each component individually as well as the vulnerability in its entirety to gain a true perspective of the risk and determine the likelihood of impact.[1]
It is up to the customer to decide if the risk associated to vulnerability found justifies or outweighs the cost of controls required to reduce the risk to an acceptable level. A customer may not be able to spend a million dollars on remediating a threat that compromises guest printers; however, they will be very willing to spend twice as much on protecting systems with the company's confidential data.[6]
The Single Loss Expectancy formula:
Risk = Asset Value * Threat * Vulnerability * Impact
The next important formula is identifying how often the SLE could occur. If an SLE worth a million dollars could happen once in a million years, such as a meteor falling out of the sky, it may not be worth investing millions in a protection dome around your headquarters. In contrast, if a fire could cause a million dollars' worth of damage and is expected every couple of years, it would be wise to invest in a fire prevention system. The number of times an asset is lost is called the Annual Rate of Occurrence (ARO)

	Last revised	Focus			Tools	Easy to use	Integration to the context of IS/IT management
		Management	High level	Technical			
OSSTMM	2010	No	Yes	No	No	No	No
ISSAF	2006	Partially	Yes	Yes	Yes	No	Partially
PTES	2014	No	Yes	Yes	Yes	Yes	No
OWASP	2014	No	Yes	Yes	Yes	Yes	No
NIST SP 800-115	2008	No	Yes	No	No	Yes	No

***Tab. 1**. Comparative analysis of current methodologies. Source: author.*

2.2. Integration of penetration tests into context of IT management

Integration of penetration tests into IT management context is based on the process model from COBIT, which is tailored to suit the specific needs of penetration tests (an original model which contains 34 processes is reduced to 16 processes. This reduced process model is an ideal baseline as the structure of the processes covers essential areas that can be effectively tested. [2]
Every process from this model can be tested by one or more steps from the detailed level and one or more steps described in specific topics (relevant mapping table is too large to be included in the article). This reduced model is particularly useful for planning the tests (decision which areas should be tested) and for remediation of vulnerabilities. In practice, the manager (usually chief information officer, chief information security officer or chief security officer) can easily benchmark security level of each process based on the results of penetration tests. He can also monitor the progress of remediation activities in specific areas.[7]

3. Detailed level

On a detailed level, the processes that take place during a penetration test from the first touch with tested infrastructure to a complete compromise (if desired) are presented. Also, the work breakdown structure is introduced. See Table 2 to understand three basic steps (planning, testing and reporting) of the detailed level. [3]

Planning	**Testing**	**Reporting**
1.1) Requirements identification	2.1) Information gathering	3.1) Cleanup
1.2) Stakeholder identification	2.2) Perimeter mapping	3.2)Document analysis
1.3) Project management team creation	2.3) Penetration	3.3) Report creation
1.4) Defining scope	2.4) Network scanning	3.4)Report presentation
1.5) Defining rules	2.5) Vulnerability scanning	
1.6) Testing team appointment	2.6) Penetration further	
1.7) Role description	2.7) Gaining access and escalation	
1.8) Kick off meeting	2.8) IS compromise	
	2.9) Maintaining access	
	2.10) Covering the tracks	

Tab. 2. Detailed level

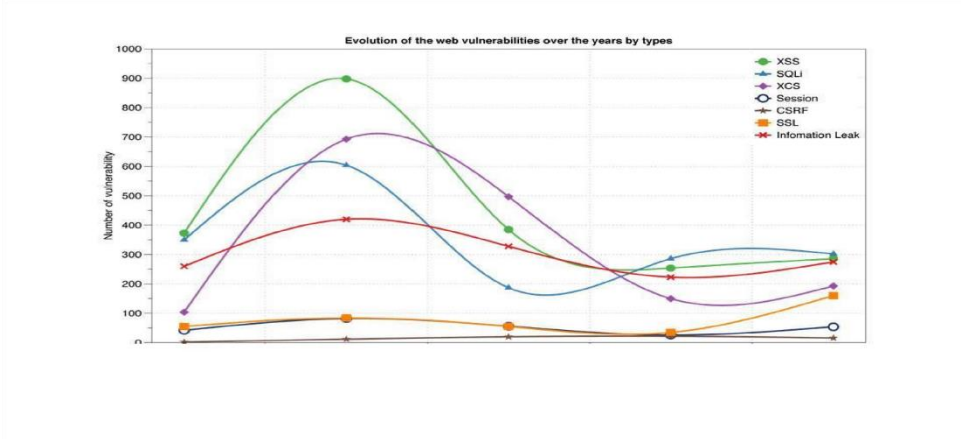
At first, during the planning step, the requirements for the test must be identified. Requirements can result from a need to adhere to some compliance standard (like PCI DSS), or a long-term security plan. Stakeholders and their needs must be also identified. Then the management team has to be created. This team appoints the testing team, no matter if the team is created from internal or external resources. The scope and rules are an important part of the test description and the test plan. The test description should include the tools and processes that the tester is (un)authorized

to use and how deep should the tester compromise the infrastructure in case of successful penetration. The test is started by a kick-off meeting.[5]

3.1. Web Applications testing

The most common types of errors encountered during testing are presented in Figure1

Tab 3: The evolution of Vulnerabilities in Web Applications 2010-2016



Tab 2 The evolution of Vulnerabilities in Web Applications 2010-2016

One of the most critical vulnerabilities that a penetration controller can possess is to find an application that will allow the execution of commands on systems. The aim of this vulnerability is high because it can allow any unauthorized users and malicious to execute commands from the application on the system and collect large amounts of information or take control of the host[2]

As you can see we have a part in the application which allows us to pinging every IP address Figure 2.

Tab 4 Pinging Addres 172.16.212.133

3.2. Way the web applications a breaking

This question will easily be able to answer if we look at the code function that allows ping-un Figure 5.

```
<?php

if( isset( $_POST[ 'submit' ] ) ) {

    $target = $_REQUEST[ 'ip' ];

    // Determine OS and execute the ping command.
    if (stristr(php_uname('s'), 'Windows NT')) {

        $cmd = shell_exec( 'ping  ' . $target );
        echo '<pre>'.$cmd.'</pre>';

    } else {

        $cmd = shell_exec( 'ping  -c 3 ' . $target );
        echo '<pre>'.$cmd.'</pre>';
```

Figure 5. Code which was implemented with the execution of the command PING

Variables seen by the code "$ target" only accepted by the user and join the command "ping" but not controlled us what kind of input is, so we were able to execute other commands and achieved the goal.[1]

3.3. SQL Injection

Injection SQL vulnerability is considered a high risk due to the fact that can lead to taking full control of the system. This is why almost all the commitments in web penetration testing, applications should always check for SQL injection. Demand is vulnerable to SQL injection when the application allows you to interact with the database and execute the query in the database.[3]

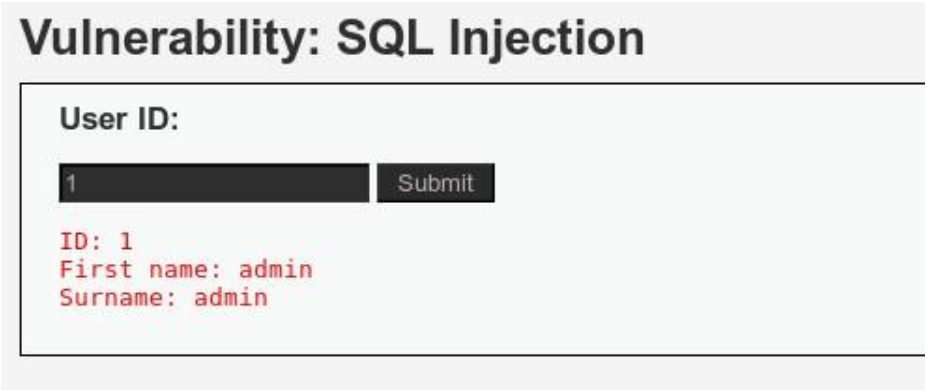


Figure 6. Implementation of the code that turns if given name and surname ID

This means that the query that was executed in the database has been as follows:
SELECT FIRST_NAME, LAST_NAME FROM users WHERE ID = '1'; Let's look at the URL:
http://172.16.212.133/dvwa/vulnerabilities/sqli/?id=1&Submit=Submit#[4]

Conclusion

The main goal of this degree project was as previously stated in the problem description to explore penetration testing in a web application environment. In order to grasp the field of security testing one has to understand the threat given by the attacker community. One of the first steps was to find out who the person behind the attacks was. Did the stereotypical image given by media of the hacker correspond to the reality? Shows a wide span of attacking types ranging from the restless teenager with little knowledge to the malicious black hat who knows all about the internal workings of every attack. Furthermore, the results also point towards very different reasons for committing the exploit. In the case of the script kiddie, the main argument for attacking a target is peer respect and status in a certain community while a black hat would perform the same illegal action for pure financial gain. A large gray zone exist between these two extreme characters where some hackers can be found who merely brake into a site to later inform the owner of the insecurity. Another problem formulated in the beginning of this work questioned the side effects of the various exploits. Every injection, and scripting attack could give examples of scenarios where all the attacker would achieve was of low security impact. Despite of this, exploit scenarios from the same flaw could also show very high impact consequences on the application. In a worst-case scenario, a company could lose credibility, sensitive information and therefore its customers.

References

1. Mike Shema 2012. “Hacking Web Apps: Detecting and Preventing Web
2. Application Security Problems”
3. Bru Dafydd Stuttard 07/October/2011 “The Web Application Hacker's Handbook: Finding and Exploiting Security Flaws”
4. Sheelah K. (2015, August 24). I Was Harassed After the Ashley Madison Hack. From http://www.bloomberg.com/news/articles/2015-08-24/i-was-harassed-afterthe- ashley-madison-hack
5. RY4N C0R3Y. (2015, May 6). Summarizing The Five Phases of Penetration Testing. From https://www.cybrary.it/2015/05/summarizing-the-five-phases-ofpenetration-testing/
6. Eric B. (2013, October 13). What Is a Penetration Test And Why Would I Need One For My Company? From http://www.forbes.com/sites/ericbasu/2013/10/13/what-isa-penetration-test-and-why-would-i-need-one-for-my-company/
7. Justin Seitz 14/December/2014 “Black Hat Python: Python Programming for Hackers and Pentesters”
8. Peter Kim 20/June/2015 “The Hacker Playbook 2: Practical Guide To Penetration Testing”