Effective Penetration Testing Approach for Modern Web Application Vulnerabilities

Leelark Sharan Saxena
Department of CSE
Rajiv Gandhi Proudyogiki Vishwavidyalaya
Bhopal, India

ABSTRACT

Now days, every business of any domain that is education, sports, heath, gaming, service etc or any government organization are online i.e. they have a web application. Each and every web application have large amount of confidential data related to their users or important data about their organization and it can be extremely destructive if it goes in the hand of wrong and unauthorized person.

This paper focuses on determining whether the developed web application is secured against different and most destructive types of web attacks or not. This paper not only describes about destructive web application attacks but it also elaborates each and every step a pen tester need to follow to detect each type of vulnerability, and how to exploit it to perform unauthorized actions as firstly it is necessary to find whether an application is vulnerable to any attack or not before directly going towards taking all precaution steps towards all type of vulnerability. And moreover penetration testing also gives a clear idea of the specific part or the functionality of the targeted web application which is vulnerable to which particular type of attack.

Keywords

Attacks, penetration testing, security, threats in web application, vulnerability, web application, web application testing

1. INTRODUCTION

Web Application has grown from being purely static information repositories into highly functional applications that process susceptible information and carry out powerful events with real world outcomes. Due to this development, numbers of factors are responsible for weak security but still the core factor for vulnerabilities is that user is allowed to submit arbitrary input.

Before studying penetration testing of web application, one should know about the 2 basic terms that are vulnerability and vulnerability assessment. The term **Vulnerability** is defined as the flaw or the weakness in the web application that could be subjugated to compromise its security. Attacks against web applications vulnerabilities are mainly to expose sensitive data or in order to gain unhampered access to the back end systems on which the application is running. While the second term **Vulnerability Assessment** refers to the 3 step process which includes identification, quantification and report creation phase. In identification phase, numbers of vulnerabilities are acknowledged and then in quantification phase, task of rating them according to technical severity rather than taking into account the affected business and its mission critical process is done. At last documentation or report is created. Vulnerability Assessment is done using both automated tools and manual vulnerability scanning. [1] [2]

Now let us know what Penetration Testing means?

Penetration testing is defines as action of evaluating, exploiting the vulnerabilities in any system. It is the extended process of vulnerability assessment. It includes scanning, enumerating, exploiting and reporting the vulnerabilities. Penetration testing is done in 5 different phases –

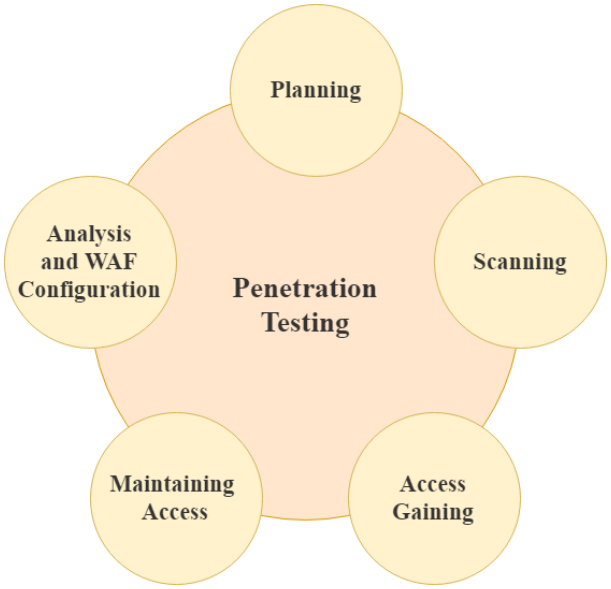

Fig 1 : Penetration Testing Phases

Phase 1 – Planning and Reconnaissance

The main aim of this phase is to determine the scope as well as goal of the test. In this phase only, tester specify the targeted system and testing methodology.

Tester also gathers information like network, mail server and domain and sub domain names of target in order to understand its working and potential vulnerabilities that it may have.

Phase 2 - Scanning

The main goal of scanning phase aims at determining the response of targeted web application to various intrusion shot. This is done by static and dynamic analysis. In static analysis, penetration tester examines the entire targeted web application's code in order to find its run time behavior. The tools used for static analysis scan the entire code in one pass. While on the

other hand, examining of an application's code is performed in its running state.

Phase 3 – Access Gaining

In this phase, various web application attacks like SQL injection, CSRF attack, cross site scripting etc are used for determining the target's vulnerabilities.

Tester attempts for exploiting the vulnerabilities by escalating privileges, stealing confidential data, interrupting traffic etc, to know about the damage that be caused.

Phase 4 – Maintaining Access

This phase mainly aims to determine that whether the vulnerability can be used for achieving continual existence in the exploited target that too long enough for gaining in depth access. The scheme is emulated higher persistent threats, which steal the most sensitive data.

Phase 5 – Analysis and WAF configuration

Then the report of penetration test is created which specifies:

- Vulnerabilities that were exploited
- Perceptive data that was accessed
- The time period pen tester was able to be in the targeted web application undetected.

And further this information is analyzed in order to patch vulnerabilities as well as protect the application against future attacks.

2. SCANNING THE TARGETED WEB APPLICATION

For gathering the complete information about the targeted web application, **whois.sc** website is used. For checking how many other website are running on the same server on which our target is running, **youngetsignal.com** website is used.

For finding out the sub domains, **knockpy** which is a python tool is used and the command for it is –

$ knockpy <domain_name of targeted web application>

In order to determine operating system on which web server is running pen tester have to ping it and if the TTL(Time To Live) value is less than 60 then it is running on Linux and if TTL value is greater than 60(about 110) then it is running on windows. Ping command that is used is written below :

ping host name or ping IP address.

For gathering more details of target, NMAP (**Network Mapper**) is used for performing deep scan. It is used to find the open ports and the running services along with their version number running on the particular port. It is used to fetch information like operating system on which web server is running, metadata, internal files and disallowed directories and lot more. Command to find open ports and running services on them is:

nmap <hostname of target> or nmap <target IP address> and nmap –sV <hostname> command is used for determining the version number of running services on free port. In order to perform aggressive scanning of targeted web application, command is - nmap –A <hostname of target>.

NMAP Scripting Engine, which is one of the most powerful features of NMAP, is used for to execute the scripts for performing networking task and getting more detail of target. Command to execute **Auth** scripts against a target in our network for discovering users accounts on remote machine is: # nmap –script auth <target host address>.

Executing default script will give information regarding operating system, work group name, net bios name etc. Running http-enum script will give all directories and file information of the target. In order to find all sub domains (same information that was gathered using knockpy), dns-brute script is used. [3]

Another tool used for performing scanning task is **metasploit,** basically it's a framework. Steps to use metasploit framework are:

- Scanning open ports, services and their version number of metasploitable Linux from metasploit. Command for performing nmap scan is –nmap –sV <ip of metasploitable linux>
- Then penetration tester has to search exploit related to particular found service from above list through msfconsole. Command to perform this task : Search <service version from above list>

  e.g., search vsftpd 2.3.4
- Activating the particular related exploit environment and command for that is

  use <exploit_name from the above list>
- Now configure the exploit according to the need of current scenario. Show option is the command which will display various parameters required in order to launch exploit properly. Penetration tester has to set RPORT and RHOST accordingly. For e.g., set RHOST 192.168.1.106.
- Setting of payload for exploit. For this show payload command is first used to list all payloads compatible with selected exploit and then set the exploit accordingly. For e.g., set PAYLOAD windows/meterpreter/reverse_tcp.
- Now again running the command "show options" in order to find whether all section are properly set for proper launching of exploit or not and if not then set them.
- Running **"exploit"** command that will launch the attack. If exploit will be successfully launched against targeted remote machine then prompt will change to "meterpreter>".
- At last in order to have complete control of server, make use of "help" command to find the commands to perform related action on remote server. [4]

Other Tools used to gather more information about the targeted web application is **wireshark**. It is a network protocol analyzer. It captures the packets and display then in readable format. It provides details about network protocols, decryption and accurate packet details. It has the feature of live capturing the packets.

3. PENETRATION TESTING STEPS FOR VULNERABILITIES FOUND IN MODERN WEB APPLICATIONS

Table 1: Classification of Vulnerabilities

Classification	Vulnerability
Cross-Site Scripting	Cross-Site Scripting
SQL Injection (SQLI)	SQL Injection
Cross Chanel Scripting	File Upload Remote File Inclusion OS command Execution Code Injection
Session Management	Session Fixation Session Hijack Authentication Bypass
Cross-Site Request Forgery	Cross-Site Request Forgery
Server Configuration	SSL misconfiguration Insecure HTTP methods
Information Leakage	Path Traversal Error Message Disclosure

Some of the most common vulnerabilities found in a modern web application are –cross-site scripting, SQL injection, XXE attack, XPath injection, File Inclusion vulnerabilities, CSRF attack, broken authentication and session management vulnerability. [5]

3.1 XSS Vulnerability

XSS (Cross Site Scripting) is basically a client side injection attack. Input receiving areas like search box, feedback areas are vulnerable to XSS attack. JavaScript code is injected in the vulnerable areas that are used for stealing cookies and session id's. XSS can be combined with other type of vulnerabilities leading to extremely destructive effect. XSS attack can be converted into self propagating worms. Reflected and stored XSS are two most common cross scripting attacks. [6]

3.1.1 Reflected XSS

It is also known as non persisted XSS. It is the most familiar type of cross-scripting attack in which attacker's malicious JavaScript is integrated as the part of HTTP request and reflected back as a HTTP response rather than storing the script code.

3.1.1.1 Steps Involved in Reflected XSS Attack

1. A user log in the web application and a cookie containing the unique session id is issued.
2. User visits to malicious page by making request to the attacker's crafted URL. For example - URL looks like ><SCRIPT>var+img=new+Image();img.src="http://pentesting/"%20+%20document.cookie;</SCRIPT> , it will return user's cookie to attacker.
3. Response given by server is along with attacker's JavaScript.
4. Attacker's JavaScript executes in the client's browser.
5. User's browser sends session token to attacker.
6. Attacker takes possession of the user's session.

3.1.2 Stored XSS

It is also known as persisted XSS. It is one of the most destructive variety of XSS attack in which attacker introduce a JavaScript which is permanently stored in the targeted web application's database.

3.1.2.1 Steps Involved in Stored XSS Attack

1. Attacker submits questions containing malicious JavaScript.
2. User log in and view attacker's question.
3. In this also, response is along with attacker's malicious JavaScript.
4. Attackers injected JavaScript code gets executed in the client's browser.
5. As soon as JavaScript executes, client's browser sends session token to attacker.
6. Finally, Attacker takes possession of the user's session.

3.1.3 DOM Based XSS

Document Object Model Based XSS of attack is possible only when Client side scripts of targeted web application write user input to DOM and if data is not properly handled then the attacker can easily inject payload which gets stored in the DOM only and implements when the web application read the data from DOM.

3.2 SQL Injection

In SQL injection, attacker try to inject malicious SQL query as input by which they gain unauthorized access to the sensitive and the confidential data stored in the targeted web application's database server. Attacker can easily retrieve, manipulate and delete the important data after gaining access. Attackers can even bypass log in and shutdown SQL server. [7]

3.2.1 Steps Involved in SQL Injection

1. Attacker first finds out input within a targeted application that is integrated in the SQL query.
2. Next step is to check vulnerability and for this single quote is added at the end of URL and after running it if error occurs regarding SQL query then only that web application is vulnerable to SQL Injection.
3. Now next task is to find the number of column by using "order by" clause. Just at the end of URL add order by n where n=1, 2, 3…. until you get an error of < unknown column>. As soon as you get error, stop and the number of column is nothing but n-1.
4. Sometimes above step may not work in this case, then comment out the rest part of query using "--".
5. Then penetration tester has to find out which column is displayed to user by running another SQL query along with original one by using "union" but note that set fussing value to the main input parameter of the original SQL query.
6. Now next task is to find out database name and user name for that database() and user() keywords are used in place of vulnerable column. Query will be—

Union select database(),user(),….—

7. After finding out database name, find out table name for that query is—

   Union select group_concat(table_name) from information_schema.tables where table_schema=<database_name>

8. Now find out column name, for that SQL query is: Select group_concat(column_name) from information_schema.columns where table_name=<table_name>

NOTE- while executing another SQL query along with original one always give extremely large or negative value for the input for fussing the web application.

3.3 XXE Attack

XXE stands for XML external Entity. Those web applications which parse data from client to server into XML and then server performs action on it and then return response in the form of XML are vulnerable to XXE attack. Weak parsing of data into XML allows an attacker to easily abuse this parsing by reading system internal files.XML is basically used for communicating between two systems which are running on different technologies. An XML document contains entities which are defined using "system" identifier and these are present within a DOCTYPE header. These entities are capable of accessing local as well as remote content.

Thus, with the help of these entities attacker can easily retrieve the important web application data.

XML format allows custom entities to be defined in XML file. So attacker can create his own entity for accessing file of his interest or change the previously present entities in the web application and make the application to show that file details.

For e.g.,

<!DOCTYPE test [<!ENTITY xxe SYSTEM "file:///etc/passwd" >]>

<test>&xxe;</test>

Encoding the above statement and passing as XML attribute value in URL.

3.4 XPath Injection

The XPath i.e. XML Path Language is an interpreted language which is used for navigating around XML documents and for retrieving content present within them. Those web application which uses XML for storing input from user as well as for response, they may use XPath for accessing the data in response to user input. And if there is no proper filtering and sanitization of the input that is inserted in XPath query, then an attacker can manipulate the query for retrieving or accessing the data for which he/she is not authorized. [8]

XPath injection is similar to SQL injection. XPath has two useful functions that are mainly used by attackers for quickly iterating through each and every node and data present in the XML document:

- **count()** –This function gives the number of child nodes of any given element, which can be further used in order to get range of position() values for iterating.
- **string-length()** –This function is used for determining the supplied string length, which can be used to get the range of substring() values to iterate over. [9]

3.4.1 Steps for Penetration Testing for XPath Vulnerability

1. Determine whether the below mentioned values result in abrupt or different application behavior but without causing an error:

- ' or count(parent::*[position()=1])=0 or 'a'='b
- ' or count(parent::*[position()=1])>0 or 'a'='b

   In case the parameter is numeric, then also test for following values:

- 1 or count(parent::*[position()=1])=0
- 1 or count(parent::*[position()=1])>0

2. If any of the above strings causes deferential behavior of targeted web application without resulting into any error, then there may be vulnerability of extracting confidential data by crafting test conditions in such a way that only one byte of information or data is drawn out at a time. In order to determine the name of the current node's parent, apply a series of test conditions of the following format:

- substring(name(parent::*[position()=1]),1,1)='a'

3. After determining the name of the parent node, last task is to extract all data within XML tree by using series of conditions of following form:

- substring(//parentnodename[position()=1]/child::node()[position()=1]/text(),1,1)='a'

3.5 File Inclusion Vulnerability

File inclusion is one of the most necessary features that is provided by many scripting languages for the development of complex web application. File inclusion helps the developer to keep the reusable code components into separate files and to include them within the function specific code files whenever they are needed. The code of the file which is included is interpreted just as if it has been placed at the include directive's location.

Attackers can abuse this functionality by including their malicious files in the targeted web app or by including server files to view its content.

File Inclusion Attack can lead to:

1. Disclosure of sensitive information.
2. Execution of remote command

3.5.1 Steps to Test for Different Types of File Inclusion Vulnerabilities

3.5.1.1 Remote File Inclusion Vulnerability

Those web applications are RFI attack vulnerable where an attacker can include the remote files such as web shells by abusing the application that dynamically take in external files.

RFI generally occurs when a web application receives path to included file as an input without perfectly sanitizing it, which allow attacker to supply external URL to include statement. Web applications developed using PHP languages are mostly vulnerable to RFI attack. [10]

Steps to test for remote file inclusion flaws:

1. Determining whether any requests are acknowledged from server hosting the targeted web application after submitting URL intended for resource on web server which was controlled in each and every targeted parameter.
2. If the above case fails, then determine whether a timeout occurs while server tries to connect on submitting the URL of nonexistent IP address.
3. If any of the above test condition is true, then the targeted web application is vulnerable to Remote File Inclusion. Now for further testing, try to include any web shell of particular language in which the application had been developed.

3.5.1.2 Local File Inclusion Vulnerability

Those web application in which attacker can include local files which are already present in server such as user-configuration files, user files etc are LFI vulnerable.

LFI attack can lead to directory traversal and information disclosure. [11]

Steps to test for local file inclusion vulnerabilities:

1. Determining whether any changes occurs in the targeted web application behavior on submitting name of known executable resource already present on the server.
2. Find out that on submitting acknowledged static resource name on the server on which targeted web application is running, its content get included into application's response or not.
3. If all above test conditions runs in your favor, then the targeted web application is vulnerable to Local File Inclusion. Now try to access any resource which cannot be accessed directly via web server.
4. Even test for gaining access to files in other directories using traversal techniques.

3.6 CSRF Attack

In cross site request forgery (CSRF) attacks, an attacker first develops an innocent and real looking website that causes victim's browser to submit a request to the vulnerable web application to perform some weird action that is useful to the attacker. Same origin policy does not forbid one web application from issuing requests to a different field. However, it prevents originating website from processing the responses to cross domain requests. Therefore, CSRF attacks are one way only. [12]

Those web applications which rely completely on HTTP cookies in order to track sessions are vulnerable to CSRF attack. As application set cookie in a client's browser, the browser automatically forward that cookie to the web application in every consequent request. If application does not acquire any protection against an attacker's "riding" on its users' sessions, then it is most likely to be CSRF vulnerable. [13]

3.6.1 Steps Involved in Penetration Testing for CSRF Vulnerability

1. Analysis of key functionality within the application.
2. Next task is to determine that web application function which can be used to carry out some sensitive actions on behalf of an innocent user, that depend completely on cookies for tracking sessions, and that take up request parameters that an attacker can easily detect in advance which means that do not hold any other tokens.
3. Create an HTML page that issues the preferred request without making any interaction with user.
   - For GET requests, set an <img> tag with src attribute set to the vulnerable URL.
   - For POST requests, develop a form that include hidden fields for all the appropriate parameters needed in order to attack and that has its target value set to susceptible URL. Even JavaScript can be used for auto submitting the form as soon as page loads.
4. Make use of same browser to load crafted HTML page, while logged in to the targeted web application. Also verify that the desired action is proceeded within the application.

3.7 Broken Authentication and Session Management

When a web application functions which are associated to authentication and the session management tasks are not implemented appropriately then it gives a chance for attackers to gain access over passwords, keys, session tokens, or to abuse other implementation weakness for gaining access to other user's identity details. This ultimately results in undermined authorization and liability controls and even results in privacy violation as well as identity stealing.

3.7.1 Broken Authentication

Weakness that may be present in authentication -

1. Some web application allows user to have weak and short password for their account, thereby making it easy for attacker to break authentication by predicting the password.
2. Some web application allows user to have n number of attempts to log in. Attackers use this weakness in order to determine correct password by making repeated login attempts with different password. This is known as Brute Force attack.
3. Many authentication mechanisms display the user name either implicitly or explicitly as error message and as a result it makes easy for the attacker to predict the password.
4. Sometimes the attackers use the forget password feature provided by web application to steal the identity details as security questions' answers are easy to guess.

3.7.2 Session Management

Some more severe attacks against authentication by gaining access over victim's session. [14]

3.7.2.1 Session Fixation

Those web application which creates anonymous session for each and every user when he/she accesses the application for the first time are likely to have session fixation vulnerabilities. In a session fixation attack, attacker first tries to get an anonymous token directly from web application and then apply some technique for fixing it within a victim's browser. Then as

soon as user log in, the attacker can hijack the victim's session using the token.

Steps involved in session fixation attack:

1. Attacker first request for login page of the targeted web application and a session token is issued.
2. Attacker feeds the session token to the user.
3. User logs in to the targeted web application through the token he/she has received from the attacker.
4. At last, attacker hijacks the user's session using same token which was fed to the user.

3.7.2.2 Session Hijacking

In session hijacking, attacker just tries to get session token of any application user. It is a 5 step process which includes:

1. **Sniffing**

   Placing yourself between the victim and the target. Then capture traffic and attempt for session information gathering with the help of sniffers like wireshark.

2. **Monitor**

   Observe the flow of packets and try to find out valid authentication packet and also predict the sequence number

3. **Session ID prediction**

   Try to guess the session id which is a difficult task. This is usually done by brute forcing. While on the other hand, sequence number can be determined easily either manually as well as with the help of tools.

4. **Session Desynchronization**

   Breaking the connection to the victim's machine

5. **Command Injection**

   The last task is to take over session communication between the workstation and the server. In order to stay away from detection, attacker spoof IP address of client and then include the session sequence number. Session is hijacked if the server agrees to the attacker provided information. [15] [16]

4. CONCLUSION

Modern web applications are max exposed and min protected, therefore vulnerable because standards are focusing more in providing number of functionality rather than on security. Penetration testing is the best approach in order to accurately examine the effectiveness of the security measures deployed in the targeted web application.

This paper proposes the best vulnerability scanning tools and the way to use them that can help to test for the existence of common vulnerabilities but these automated tools are not enough for conducting high quality penetration testing. So in order to perform a high level testing against vulnerabilities requires manual approach and this paper focused on each and every steps needed to check for every specific vulnerability that may be present in the targeted modern web application. Penetration testing can help to reduce the attacks on web applications by giving the complete overview on the vulnerabilities found in target and the functionality which is vulnerable. Further security measures will be taken to prevent the target against vulnerable attacks as analyzed in penetration testing phase.

Future task is to uncover more vulnerabilities that are destructive and to build an efficient common open source tool based on those vulnerabilities.

5. REFERENCES

[1] Xiaowei Li and Yuan Xue, A Survey on Web Application Security. Available : http://www.isis.vanderbilt.edu/sites/default/files/main_0.pdf

[2] Gopal R. Chaudhari and Prof. Madhav V. Vaidya, "A Survey on Security and Vulnerabilities of Web Application", (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 5 (2), (2014)

[3] "Nmap: the network mapper" [online]. Available: https://nmap.org/

[4] "metasploit" [online]. Available: https://www.metasploit.com

[5] OWASP Top 10 Web Application Vulnerabilities, https://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project

[6] Shrivastava, Ankit, Santosh Choudhary, and Ashish Kumar. "XSS vulnerability assessment and prevention in web application" Next Generation Computing Technologies (NGCT), 2016 2nd International Conference on. IEEE, 2016.

[7] What is SQL Injection and How to Prevent it | Netsparker - (2018, June 5). [Online] Available: https://www.netsparker.com/blog/web-security/sqlinjection-vulnerability/

[8] www.w3schools.com/xpath - Web Element Locator.

[9] XPATH Injection – OWASP (2018, May 19). [Online] Available: https://www.owasp.org/index.php/XPATH_Injection

[10] What is the Remote File Inclusion vulnerability? [Online] Available: https://www.netsparker.com/blog/web-security/remote-file-inclusion-vulnerability/

[11] Afasana Begum and Md. Maruf Hassan,"RFI and SQLi based Local File Inclusion Vulnerabilities in Web Applications", *International Workshop on Computational Intelligence (IWCI)*, 12-13 Dec 2016.

[12] Nikunj Tande and Kalpesh Patel,"Mitigation of CSRF Attack", International Journal of Science and Research (IJSR), (2012).

[13] "Threat Modelling for CSRF Attacks", Xiaoli Lin, Pavol Zavarsky, Ron Ruhl and Dale Lindskog, 2009 International Conference on Computational Science and Engineering.

[14] C. Visaggio,"Session Management Vulnerabilities in Todays Web", in IEEE Security and Privacy,48-56,

2010Tavel, P. 2007 Modeling and Simulation Design. AK Peters Ltd.

[15] Jerry, Louis, Detection of session hijacking, 2011.

[16] Vishnoi, Monika and Tech,Laxman and Agarwal, MIT, Session Hijacking And Its Countermeasures, International Journal of Scientific Research Engineering and Technology (IJSRET), (2013)250–252.