Oct 27th, 9:00 AM - 10:30 AM

Vulnerability Assessment & Penetration Testing: Case study on web application security

Gazmend Krasniqi
University of Vlora "Ismail Qemali", gazmend.krasniqi@hotmail.com

Veton Bejtullahu
University for Business and Technology, veton.bejtullahu@gmail.com

Vulnerability Assessment & Penetration Testing: Case study on web application security

Gazmend Krasniqi[1], Veton Bejtullahu[2]

[1] University of Vlora „Ismail Qemali“, Faculty of Technical Sciences, Vlorë, Albania
[2] UBT – Higher Education Institution, Lagja Kalabria, 10000 p.n., Prishtinë, Kosovo

[1]gazmend.krasniqi@hotmail.com; [2]veton.bejtullahu@gmail.com;

Abstract. Complexity of information systems are increasing day by day. The security of information systems that are connected to public networks can be compromised by unauthorized, and usually anonymous, attempts to access them. By using public networks businesses and other institutions are exposed to numerous risks. This leads to more and more vulnerabilities in Information Systems. This situation calls for test methods that are devised from the attacker's perspective to ensure that test conditions are as realistic as possible. In this paper we will describe complete stages of Vulnerability Assessment and Penetration Testing on some systems and proactive action taken to resolve that vulnerability and stop possible attack. Also we will describe prevalent Vulnerability assessment techniques and some security tools for one web applications, including procedures which are used in real life for testing the security.

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 security of web applications.

Keywords: Security, Vulnerability Assessment, Penetration Testing, Web Applications

Introduction to Penetration Testing

By using public networks businesses and public authorities are exposed to numerous risks. Public and private entities are often unable to grasp the full extent of today's complex communication structures and frequently have little or no control over them. Enterprises and public authorities connect to the internet, thereby yielding some of their responsibility (e.g. availability of external servers and networks), but also exposing themselves to new threats which need to be tackled appropriately [1].

Penetration Testing or Pen Test is the process of trying to gain access to resources without the knowledge of the username, password, and other normal access tools. Penetration testing is a comprehensive method to test the complete, integrated, operational, and trusted computing base that consists of hardware, software and people [2]. The process involves an active analysis of the system for any potential vulnerabilities, including poor or improper system configuration, hardware and software flaws, and operational weaknesses in the process or technical countermeasures [3]. If the focus is on computer resources, then examples of successful deception would be theft or destruction of confidential documents, prices, databases, and other protected information. The main thing that distinguishes a Penetration Tester from an attacker is the authorization. Penetration Tester will have permission from the owner of those IT resources that are being tested and will be responsible for providing a report.

Computingsystem professionals use penetration testing to address problems inherent in vulnerability assessment, focusing on high-severity vulnerabilities. Penetration testing is a valued assurance assessment tool that benefits both business and its operations [4].

Why Penetration Testing?

Despite the fact that cyber attacks have increased dramatically, many companies or organizations still do not monitor the security of their infrastructure actively so they remain a prey for the perpetrators or hackers. Once connected to the Internet, company systems can be controlled, scanned or even attacked.

The purpose of the penetration test is to recognize the level of security and increase the security of the computing resources being tested. So one of the first steps to prevent cyber attacks is exactly the Penetration Test.

Benefits of Penetration Testing

Penetration testing allows testers to view the client system through the eyes of malicious hackers [5]. Such a process can bring a large number of discoveries and provide the client with the time needed to repair the system before the attack actually occurs. In addition, penetration testing can help organizations and companies to prove the effectiveness or ineffectiveness of security measures being implemented, demonstrating explicit security vulnerabilities in the system. More importantly, this test provides evidence to alert the management of the need to take security of information more seriously [6].

Penetration testing evaluates the effectiveness of existing security products and provides the supporting arguments for future investment or upgrade of security technologies. It provides a "proof of issue" and a solid case for proposal of investment to senior management [7].

Vulnerability Assessment and Penetration Test – How to conduct a test?

Vulnerability Assessment and Penetration Testing is a total 9 step process that is shown in Figure 1.

First of all, the tester must set the scope of the assignment (black / gray / white box). After deciding the scope, the tester gets information about the operating system, network, and IP address in reconnaissance step. After that, he uses various vulnerability assessment techniques on the testing object to find vulnerabilities. Then the tester analyzes the weaknesses found and makes the plan for the penetration testing. The tester uses this plan to penetrate the victim's system. After penetrating the system, the tester increases the privilege in the system. In the results analysis step, the tester analyzes all results and draws up recommendations to solve the system's vulnerabilities.

All these activities are documented and sent to the management to take the appropriate actions. After all these steps, the victim's system and its program get affected and altered. In the cleanup step the system is restored to the previous state as it was before the start of the process [8].

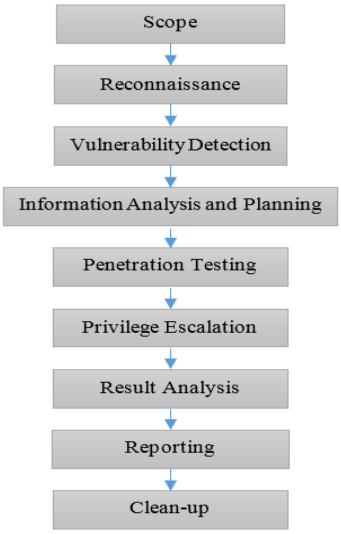


Figure 1 – Vulnerability Assessment and Penetration Testing life cycle

The results of the test are thoroughly investigated during the test analysis phase. These results are provided to the organization so it must be comprehensive and systematic. Preparation of a mitigation plan is important in penetration testing [9].

Penetration testing methodologies

Since most of the network or system penetration tests share some of the main phases, many security professionals have introduced different methodologies to conduct a penetration test ranging from simple ones to more sophisticated and formal processes.
In general, the pen test involves three main stages that imitate the steps that would be used by real hackers to carry out an attack. The three respective phases are: pre-attack, attack, and post-attack. The pre-attack phase attemps to investigate or explore the target. The attack phase involves the actual compromise of the target. Lastly, the post-attack phase which is unique to the penetration testing team, attempts to return any modified system(s) to the previous stage before the test begins [6].
A basic method of penetration testing methodology consists of the 3 following steps shown in Figure 2.

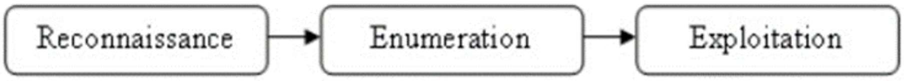


Figure 2 – Basic methodology of penetration testing

Reconnaissance
Also known as the Information Gathering step, this is the process of searching available information that is used in a penetration test. Depending on the scope of a pen test, reconnaissance activities can be ranged from ping sweeps to discover IP addresses on the network, obtaining useful information from company employees, rummaging through company's dumpsters to find receipts of telecommunications services, to thieving, lying of people, interference in phone calls and the network. The search for information is limited by the willingness and ethical behavior agreed between the client and the penetration test team [6].

Enumeration
This is the process where information is gained directly from the target system, apps, and networks with the help of tools and techniques in order to build the company's environmental picture. Network enumeration creates a view of the network configuration being tested, while the host enumeration identifies the services available on various devices like firewalls, routers, and web servers, and reveals their functions along with the opening ports that can be used to infiltrate the system. This process identifies and lists the potential weaknesses [6].

Exploitation
The exploitation phase uses different automated tools, techniques and manual steps specifically executed to compromise the system through identified vulnerabilities or

other open channels. The ultimate goal of this process is to provide administrative access to the system [6].

A formal method of penetration testing built around these 3 steps, usually involves more sub-activities, as shown in Figure 3.

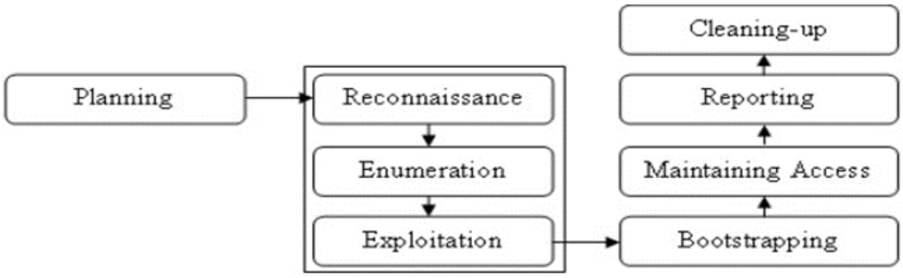


Figure 3 – Formal methodology of penetration testing

Planning is the first step that affects the test result. *Bootstrapping* is the activity that starts the process again to see if there are new weaknesses. The most important step is *Reporting*, where a good report should consist of clear descriptions of problems together with their potential impacts, security level assessment, along with the number of practical recommendations for customers to reduce or eliminate in general such weaknesses.

Penetration testing techniques

Black box testing

In this technique, the tester does not have any prior knowledge of network architecture or network testing systems. Usually black box testing is performed from the external network into the internal network. The tester should use his expertise and skills to carry out this test.

Grey box testing

In this technique, the tester has some partial knowledge of the test network. The tester doesn’t have the full knowledge of the network architecture, but he knows some basic information about network testing and system configuration. Gray box testing is the combination of two other techniques. This can be done from the internal or external network.

White box testing

The tester has full knowledge of network configuration and configuration of the network system. Usually this test is performed from the internal network. White box testing requires a deep understanding of the network or system and gives the best results.

Web Application Penetration Test

Web Application Penetration Test is the process that tests a website or a web application using manual or automatic penetration tests to identify any vulnerabilities, security weaknesses, or threats. Testing involves the use or implementation of any known attack in the application. The tester fabricates attacks and environment from the hacker's perspective, as using SQL Injection testings.

Web applications technicaly expose themselves to attackers due to their very nature of being publicly accessed and processing data elements from within HTTP requests[10].

The key goal of web penetration testing is to identify security vulnerabilities throughout the web application and its components (source code, database, and back-end network). It also helps prioritize identified weaknesses, and ways to avoid them. So, the biggest weakness that exists, will show up at the beginning of the report.

Web Vulnerabilities

A web vulnerability is a weakness or misconfiguration in a website or web application code that allows the attacker to gain control over the site and possibly the host server. Most vulnerabilities are exploited through automated tools such as vulnerability scanners or botnets. Cybercriminals create specialized tools for search on the internet of certain platforms, looking for common and published weaknesses. Once found, these vulnerabilities are then exploited to steal data, distribute malicious content, or inject damage and spam content on the weak site.

Case Study – Web Application Penetration Testing

Process of Web Penetration Testing

After we have received the program and we are registered, in this case with the Acunetix program we begin the process of testing the target webpage. On the left side of the menu, we choose **Targets** that contains the list of pages tested earlier and the possibility to add new targets. We click the **Add Target** button and initially we write the URL of the webpage, then choose the type of Scan, whose options are shown in Figure 4.

We continue with the type of Report we want to see in the end (Affected Items, Developer, Executive Summary, Quick). Click the **Create Scan** button and so we have started the scanning or testing of the specified page.

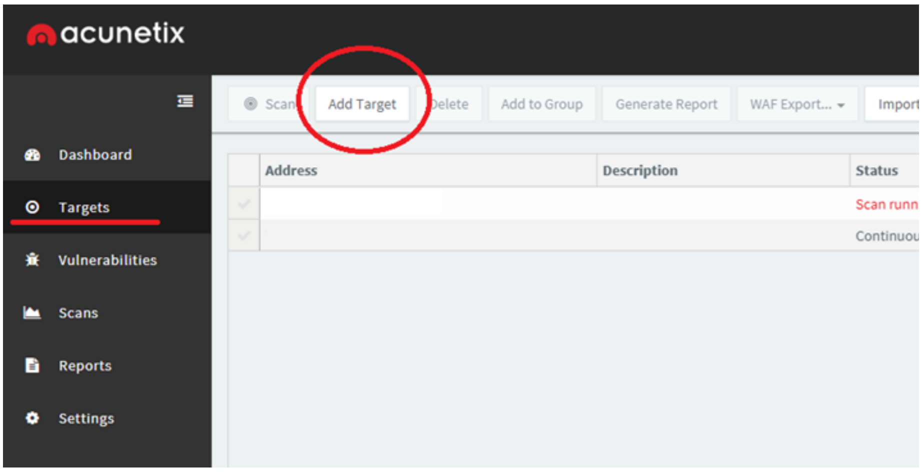
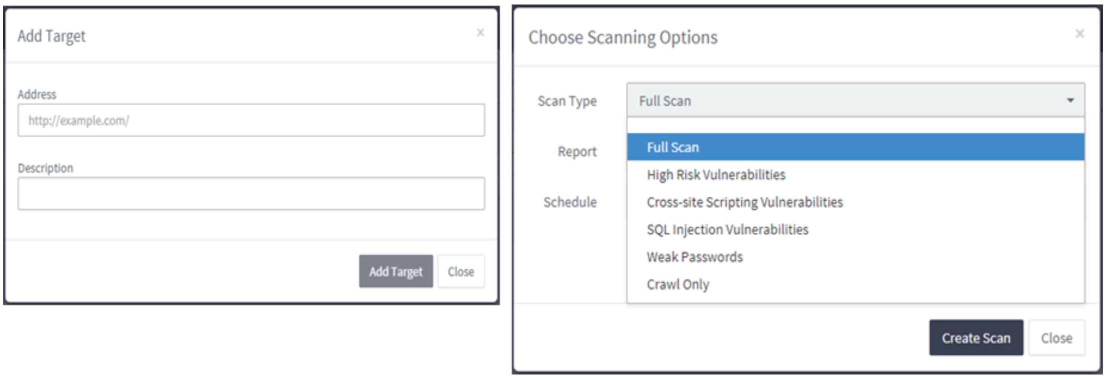
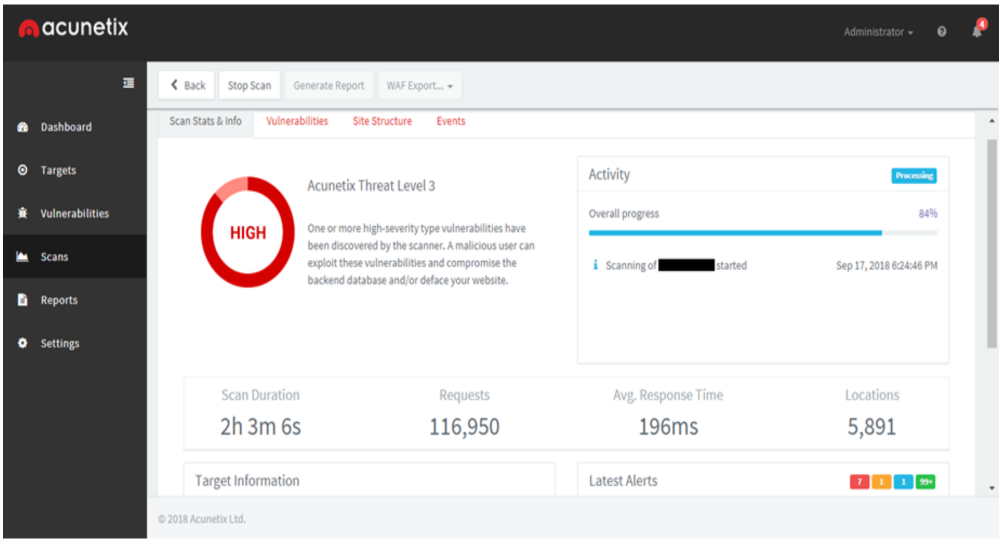


Figure 4 – Acunetix Interface 1



Scan Type: Full Scan

Figure 5 – Acunetix Interface 2



Acunetix Threat Level 3

One or more high-severity type vulnerabilities have been discovered by the scanner. A malicious user can exploit these vulnerabilities and compromise the backend database and/or deface your website.

Activity

Overall progress 84%

Scanning of [redacted] started Sep 17, 2018 6:24:46 PM

Scan Duration	Requests	Avg. Response Time	Locations
2h 3m 6s	116,950	196ms	5,891

Figure 6 – Acunetix Interface 3

The penetration testing process does not have a fixed duration. Depending on the content of the site it may take several minutes or several hours. During the process, you can view the requests sent to the server and the average time of the responses being received.

The Vulnerabilities and Site Structure buttons show the real-time list of weaknesses as soon as they are discovered, as well as the structure or form they have on the website.

The Events button contains additional information about the time of when scanning started.Target Information provides data such as the name of the website, the type of server that hosts the site, the operating system, and whether the webpage is responsive.

Latest alerts shows the number of weaknesses for High, Medium, Low and Informational levels.

Already in deep scan, Figure 6 shows that testing has sent over 100,000 requests and has found 7 High vulnerabilities, 1 Medium, 1 Low, and over 200 Informational weaknesses, where the latter do not pose any significant risk.

Once the scanning is complete, click the Generate Report button and the program will prepare the report which we can download in PDF format. Even if during the process we have clicked the Stop Scan button, we still have the opportunity to generate a report that will show all the weaknesses found until that moment. However, this method is not recommended because we may lose valuable information.

Summary of the tested Targets

With the permission of their owners, for this case study several websites were tested for vulnerabilities through NetSparker and Acunetix programs. By finding these weaknesses, we will be aware of whether the site is at risk from attackers, for access to sensitive information or blocking it.

The report consists of weaknesses found in these websites, the potential risk they contain, and suggestions to reduce and eliminate those problems in order to prevent potential attacks. Finally, the report will be submitted to site owners and developers to continue the process of improving vulnerabilities and security holes in those websites, thus protecting the risk of hacking. Because of security and confidentiality between the tester and the client, the names of the tested webpages will not be shown but will be marked with numbers:

1. Target 1
2. Target 2
3. Target 3

Vulnerabilities

Target 1:

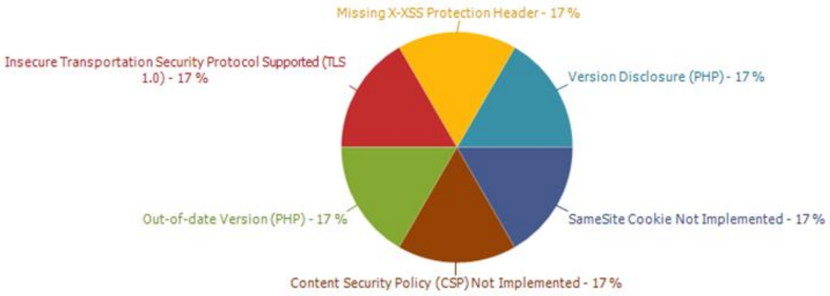


Figure 7 – Target 1 test results

Vulnerability: Cookie Not Marked as Secure – High Risk

Description:

This information can help an attacker gain a greater understanding of the systems in use and potentially develop further attacks targeted at the specific version of PHP.

Impact:

This cookie will be transmitted over a HTTP connection, therefore if this cookie is important (*such as a session cookie*), an attacker might intercept it and hijack a victim's session. If the attacker can carry out a man-in-the-middle attack, he/she can force the victim to make an HTTP request to steal the cookie.

Remedy:

1. See the remedy for solution.
2. Mark all cookies used within the application as secure. If the cookie is not related to authentication or does not carry any personal information, you do not have to mark it as secure.
3. Mark all cookies used within the application as secure.

Vulnerability: Version Disclosure (PHP) – Low Risk

Description:
We have identified a cookie not marked as secure, and transmitted over HTTPS. This means the cookie could potentially be stolen by an attacker who can successfully intercept and decrypt the traffic, or following a successful man-in-the-middle attack.
This cookie will be transmitted over a HTTP connection, therefore if this cookie is important (such as a session cookie), an attacker might intercept it and hijack a victim's session. If the attacker can carry out a man-in-the-middle attack, he/she can force the victim to make an HTTP request to steal the cookie.
Impact:
An attacker might use the disclosed information to harvest specific security vulnerabilities for the version identified.
Remedy:
Configure your web server to prevent information leakage from the SERVER header of its HTTP response.

Vulnerability: Insecure Transportation Security Protocol Supported – Low Risk

Description:
Netsparker detected that insecure transportation security protocol (TLS 1.0) is supported by your web server.
TLS 1.0 has several flaws. An attacker can cause connection failures and they can trigger the use of TLS 1.0 to exploit vulnerabilities like BEAST (Browser Exploit against SSL/TLS).
Websites using TLS 1.0 will be considered non-compliant by PCI after 30 June 2018.
Impact:
Attackers can perform man-in-the-middle attacks and observe the encryption traffic between your website and its visitors.
Remedy:
Configure your web server to disallow using weak ciphers. You need to restart the web server to enable changes.

- For Apache, adjust the SSLProtocol directive provided by the mod_ssl module. This directive can be set either at the server level or in a virtual host configuration.

```
SSLProtocol +TLSv1.1 +TLSv1.2
```

- For Nginx, locate any use of the directive ssl_protocols in the nginx.conf file and remove TLSv1.

```
ssl_protocols TLSv1.1 TLSv1.2;
```

- For Microsoft IIS, you should make some changes on the system registry.

1. Click on Start and then Run, type regedt32 or regedit, and then click OK.
2. In Registry Editor, locate the following registry key or create if it does not exist:

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Protocols\TLS 1.0\
```

3. Locate a key named Server or create if it doesn't exist.
4. Under the Server key, locate a DWORD value named Enabled or create if it doesn't exist and set its value to "0".

Target 2:

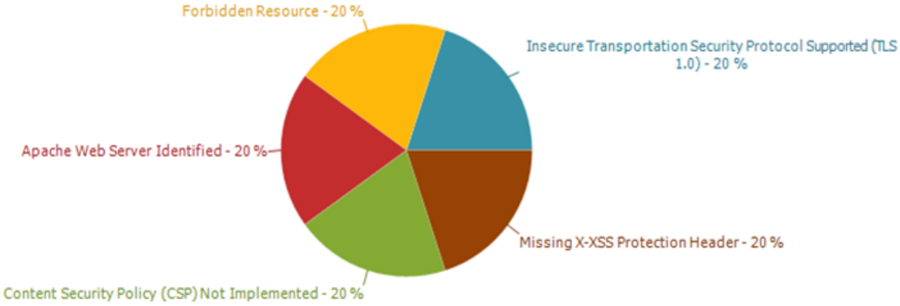


Figure 8 –Target 2 test results

Vulnerability: Insecure Transportation Security Protocol Supported – Low Risk

Description:
Netsparker detected that insecure transportation security protocol (TLS 1.0) is supported by your web server.
TLS 1.0 has several flaws. An attacker can cause connection failures and they can trigger the use of TLS 1.0 to exploit vulnerabilities like BEAST (Browser Exploit Against SSL/TLS).
Websites using TLS 1.0 will be considered non-compliant by PCI after 30 June 2018.
Impact:
Attackers can perform man-in-the-middle attacks and observe the encryption traffic between your website and its visitors.
Remedy:
Configure your web server to disallow using weak ciphers. You need to restart the web server to enable changes.

- For Apache, adjust the SSLProtocol directive provided by the mod_ssl module. This directive can be set either at the server level or in a virtual host configuration.

```
SSLProtocol +TLSv1.1 +TLSv1.2
```

- For Nginx, locate any use of the directive ssl_protocols in the nginx.conf file and remove TLSv1.

```
Ssl_protocols TLSv1.1 TLSv1.2;
```

- For Microsoft IIS, you should make some changes on the system registry.

1. Click on Start and then Run, type regedt32 or regedit, and then click OK.
2. In Registry Editor, locate the following registry key or create if it does not exist:

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Protocols\TLS 1.0\

3. Locate a key named Server or create if it doesn't exist.

4. Under the Server key, locate a DWORD value named Enabled or create if it doesn't exist and set its value to "0".

Target 3:

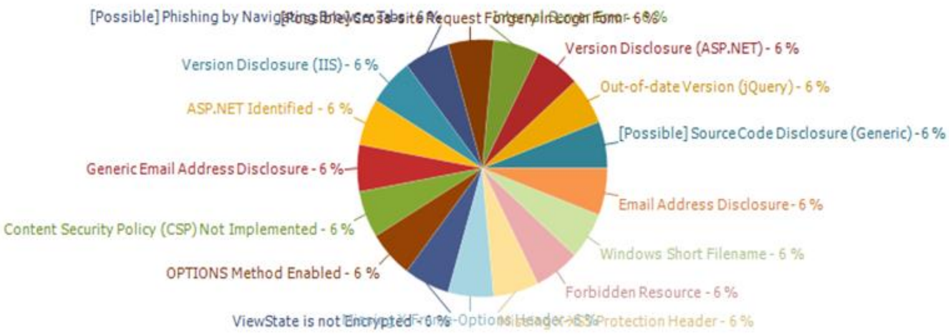


Figure 9 – Target 3 test results

Vulnerability: Out-Of-Date Version (jQuery) – Medium Risk

Description:

Netsparker identified the target web site is using jQuery and detected that it is out of date.

Impact:

Since this is an old version of the software, it may be vulnerable to attacks.

Remedy:

Please upgrade your installation of jQuery to the latest stable version.

Exploit

https://bugs.jquery.com/ticket/11290

Vulnerability: [Possible] Source Code Disclosure – Medium Risk

Description:

Netsparker identified a possible source code disclosure (Generic).

An attacker can obtain server-side source code of the web application, which can contain sensitive data - such as database connection strings, usernames and passwords - along with the technical and business logic of the application.

Impact:

Depending on the source code, database connection strings, username and passwords, the internal workings and business logic of the application might be revealed. With such information, an attacker can mount the following types of attacks:

- Access the database or other data resources. Depending on the privileges of the account obtained from the source code, it may be possible to read, update or delete arbitrary data from the database.
- Gain access to password protected administrative mechanisms such as dashboards, management consoles and admin panels, hence gaining full control of the application.
- Develop further attacks by investigating the source code for input validation errors and logic vulnerabilities.

Remedy:

1. Confirm exactly what aspects of the source code are actually disclosed; due to the limitations of these types of vulnerability, it might not be possible to confirm this in all instances. Confirm this is not an intended functionality.
2. If it is a file required by the application, change its permissions to prevent public users from accessing it. If it is not, then remove it from the web server.
3. Ensure that the server has all the current security patches applied.
4. Remove all temporary and backup files from the web server.

Required Skills for Successful Exploitation

This is dependent on the information obtained from the source code. Uncovering these forms of vulnerabilities does not require high levels of skills.

However, a highly skilled attacker could leverage this form of vulnerability to obtain account information from databases or administrative panels, ultimately leading to the control of the application or even the host the application resides on.

Vulnerability: Internal Server Error – Low Risk

Description:

Netsparker identified an internal server error.

The server responded with an HTTP status 500, indicating there is a server-side error. Reasons may vary, and the behavior should be analyzed carefully. If Netsparker is able to find a security issue in the same resource, it will report this as a separate vulnerability.

Impact:

The impact may vary depending on the condition. Generally this indicates poor coding practices, not enough error checking, sanitization and whitelisting.

However, there might be a bigger issue, such as SQL injection. If that's the case, Netsparker will check for other possible issues and report them separately.

Remedy:

Analyze this issue and review the application code in order to handle unexpected errors; this should be a generic practice, which does not disclose further information upon an error.

All errors should be handled server-side only.

Vulnerability: Version Disclosure (ASP.NET) – Low Risk

Description:

Netsparker identified a version disclosure (ASP.NET) in target web server's HTTP response.

This information can help an attacker gain a greater understanding of the systems in use and potentially develop further attacks targeted at the specific version of ASP.NET.

Impact:

An attacker might use the disclosed information to harvest specific security vulnerabilities for the version identified.

Remedy:

Apply the following changes to your web.config file to prevent information leakage by using custom error pages and removing X-AspNet-Version from HTTP responses.

```
<System.Web>
        <httpRuntime enableVersionHeader="false" />
        <customErrors mode="On" defaultRedirect="~/error/GeneralError.aspx">
                <error statusCode="403" redirect="~/error/Forbidden.aspx" />
                <error statusCode="404" redirect="~/error/PageNotFound.aspx" />
                <error statusCode="500" redirect="~/error/InternalError.aspx" />
        </customErrors>
</System.Web>
```

Vulnerability: ViewState is not Encrypted – Low Risk

Description:

Netsparker detected that ViewState encryption is disabled.

Impact:

An attacker can study the application's state management logic for possible vulnerabilities; if your application stores application-critical information in the ViewState, it will also be revealed.

Remedy:

ASP.NET provides encryption for ViewState parameters.

For page based protection, place the following directive at the top of affected page.

```
<%@Page ViewStateEncryptionMode="Always" %>
```

You can also set this option for the whole application by using web.config files.

Apply the following configuration for your application's web.config file.

```
<System.Web>
        <pages viewStateEncryptionMode="Always">
</System.Web>
```

Vulnerability: Missing X-Frame-Options Header – Low Risk

Description: Netsparker detected a missing X-Frame-Options header which means that this website could be at risk of a clickjacking attack.

The X-Frame-Options HTTP header field indicates a policy that specifies whether the browser should render the transmitted resource within a frame or an iframe. Servers can declare this policy in the header of their HTTP responses to prevent clickjacking attacks, which ensures that their content is not embedded into other pages or frames.

Impact:

Clickjacking is when an attacker uses multiple transparent or opaque layers to trick a user into clicking on a button or link on another page when they were intending to click on the top-level page. Thus, the attacker is "hijacking" clicks meant for their page and routing them to other another page, most likely owned by another application, domain, or both.

Using a similar technique, keystrokes can also be hijacked. With a carefully crafted combination of stylesheets, iframes, and text boxes, a user can be led to believe they are typing in the password to their email or bank account but are instead typing into an invisible frame controlled by the attacker.

Remedy:

- Sending the proper X-Frame-Options in HTTP response headers that instruct the browser to not allow framing from other domains.
- Employing defensive code in the UI to ensure that the current frame is the most top level window.

Vulnerability: Windows Short Filename – Low Risk

Description:

Netsparker identified a Windows short file/folder name disclosure.

The vulnerability is caused by the tilde character (~) with the old DOS 8.3 name convention in an HTTP request. It allows a remote attacker to disclose file and folder names that is not supposed to be accessible.

Impact:

Attackers could find important files that are normally not accessible from the outside and gain intelligence about the application infrastructure. This may cause the leakage of files containing sensitive information such as credentials, configuration files and maintenance scripts.

Remedy:

In order to disable short names creation, add a registry key named

NtfsDisable8dot3NameCreation

to HKLM\SYSTEM\CurrentControlSet\Control\FileSystem and set its value to "1".

Vulnerability: [Possible] Cross Site Request Forgery in Login Form – Low Risk

Description:

Netsparker identified a possible Cross-Site Request Forgery in login form.

In a login CSRF attack, the attacker forges a login request to an honest site using the attacker's user name and password at that site. If the forgery succeeds, the honest server responds with a Set-Cookie header that instructs the browser to mutate its state by storing a session cookie, logging the user into the honest site as the attacker. This session cookie is used to bind subsequent requests to the user's session and hence to the attacker's authentication credentials. The attacker can later log into the site with his legitimate credentials and view private information like activity history that has been saved in the account.

Impact:

Attackers could find important files that are normally not accessible from the outside and gain intelligence about the application infrastructure. This may cause the leakage of files containing sensitive information such as credentials, configuration files and maintenance scripts.

Remedy:

In order to disable short names creation, add a registry key named
NtfsDisable8dot3NameCreation
to HKLM\SYSTEM\CurrentControlSet\Control\FileSystem and set its value to "1".

Vulnerability:[Possible] Phishing by Navigating Browser Tabs – Low Risk

Description:

Opened windows through normal hrefs with target="_blank" can modify window.opener.location and replace the parent webpage with something else, even on a different origin.

While this doesn't allow script execution, it does allow phishing attacks that silently replace the parent tab.

Impact:

If the links lack of rel="noopener noreferrer" attribute, third party site can change the URL of source tab using window.opener.location.assign and trick the user as if he is still in a trusted page and lead him to enter his secret information or credentials to this malicious copy.

Remedy:

To prevent pages from abusing window.opener, use *rel=noopener*. This ensures window.opener is null in Chrome 49 and Opera 36. For older browsers and in Firefox, you could use *rel=noreferrer* which also disables the Referer HTTP header.

<a href="..." target="_blank" rel="noopener noreferrer">...</a>

Conclusions

Securing a computer, system, network or an app is the responsibility of everyone who is part of its configuration and those who use it. Starting with system architects, developers, testers, and product quality assurance team, and the project management team [11]. So, not only Cyber Security team have to deal with the risk that hackers bring. Organizations and companies should provide to teach their employees the basic rules of online protection, and the steps to be followed to prevent possible attack. Continuous system or application testing and tightening of protection mechanisms with updates saves the company's damage to unimaginable values. The timely improvement of technological equipment or their replacement with new equipment is a very important step in maintaining safety in general.

Outside the companies, all of us owning personal technology devices such as smartphones, computers, laptops, or any other smart device (TV, fridge, speaker) are also at risk of being intercepted and attacked. Nowadays it is known that there are many applications that have access to our data as soon as they are installed on the device. They can be used to record the movements and everything we do when we have a smartphone with us. The most commonly stolen information are the phone numbers and the location of the device.

Although a simple citizen may not have the privacy and data that risk large sums of money or confidential documents, privacy and personal confidentiality are the rights of everyone.

Internet protection begins with the awareness and recognition of the equipment or programs we face every day. Following are some recommendations on how to be safer in the cyber space.

References

[1] Study A Penetration Testing Model, Federal Office for Information Security(BSI), Retrieved October 14, 2018, from https://www.bsi.bund.de/SharedDocs/Downloads/EN/BSI/Publications/Studies/Penetration/penetration_pdf.pdf;jsessionid=859F1AF817C8C95991A5B91C4AB5ADD6.1_cid360?__blob=publicationFile&v=1

[2] McGraw, G. (2006). Software Security: Building Security In, Adison Wesley Professional.

[3] Mohanty, D. "Demystifying Penetration Testing HackingSpirits,"

[4] An Overview of Penetration Testing, Aileen G. Bacudio, Xiaohong Yuan, Bei-Tseng Bill Chu, Monique Jones, International Journal of Network Security & Its Applications (IJNSA), Vol.3, No.6, November 2011

[5] Engebretson, P. (2011). The Basics of Hacking and Penetration Testing: Ethical Hacking and Penetration Testing Made Easy. Retrieved from: http://AUT.eblib.com.au/patron/FullRecord.aspx?p=730200

[6] Ch. Phong "A Study of Penetration Tools and Approaches" MA Thesis, 2014

[7] iVolution Security Technologies, "Benefits of Penetration Testing," http://www.ivolutionsecurity.com/pen_testing/benefits.php

[8] J.N. Goel "Vulnerability Assessment & Penetration Testing as a Cyber Defence Technology" Conference Paper, 2015

[9] Saindane, M. "Penetration Testing – A Systematic Approach," http://www.infosecwriters.com/text_resources/pdf/PenTest_MSaindane.pdf, Accessed on October 14, 2018.

[10] Melbourne, J., & Jorm, D. (2010c). Penetration Testing for Web Applications (Part Three). Retrieved October 14, 2018, from https://www.symantec.com/connect/articles/penetration-testing-web- applications-part-three

[11] SANS Inst. "Security Testing of web applications: Best Practices and Tools" GIAC paper, 2004