

Network Security Assignment

Mircea-Radu-Şerban Dobrău
P147987

M7X00671
Dr. Carlene Campbell

Abstract

An essential skill of a Network Security professional is the ability and knowledge of dealing with penetration testing scenarios in real life. Some of the important skills a professional penetration tester must have involve: keeping up-to-date with trends in the Information Technology field through continuous learning and training; keeping up with OS and software vulnerabilities, as well as the patches and methods required towards fixing them; having a knowledge of the true costs of security breaches and the true cost of fixing them. This report presents an attempt of demonstrating adequate penetration testing knowledge. It is written to the effect of meeting the requirements of the M7X00671 module content and learning outcome.

Contents

1	Introduction	3
2	Contingency plan	3
3	Choice of penetration testing framework	4
4	Choice of penetration testing tools	4
5	Reconnaissance	5
6	Data compromise: SQL injection	6
7	Session hijacking: session cookie without the “HTTPOnly” flag	7
8	Session hijacking: session cookie without the “secure” flag	7
9	Arbitrary command execution: Yoast SEO	8
10	Click-jacking: X-Frame-Options header is not defined	8
11	XSS: X-XSS-Protection header is not defined	9
12	Arbitrary command execution: X-Content-Type-Options	10
13	Man-in-the-middle: no HSTS support	10
14	Information disclosure: file disclosure	11
15	Information disclosure: “ramwptest”	11
16	Information disclosure: web-page full-path disclosure	12
17	Timetable for execution of solution	12
18	Cost of Solution	13
19	Conclusion	13
	References	14

1 Introduction

The assignment requirements start with the assumption that the writer of the report has been assigned to conduct a real penetration test on the tevelcyber.com domain. As such, the tasks provided towards the accomplishment of this objective are the following:

- To conduct a “black box” penetration test on the tevelcyber.com domain.
- To use the OWASP Top 10 framework as a starting point of reference for the penetration testing process, as well as to freely use and/or research any other types of vulnerabilities and attacks that may prove useful during the penetration testing process.
- To write a technical report, whose scope is to detail the findings of the penetration testing process.

The report is to comprise of details of the vulnerabilities found, as well as an exploit methodology, Proof of Concept (PoC) and information required towards the reproduction and validation of each.

Furthermore, Tevel Cyber Corps Private Limited provides a set of rules and guidelines, which are briefly outlined below:

- Do not use found vulnerabilities with the scope of pivoting (for example, towards finding other vulnerabilities).
- Do not engage in any form of malicious behaviour that would either:
 - degrade the operation of the service (for example, a Denial of Service attack),
 - result in sending of unsolicited or duplicative messages, or
 - result in access and/or tampering of highly confidential data.

2 Contingency plan

Several risks [1, p. 3] may arise during the penetration testing process, which may be caused by the penetration testing process itself or through someone else’s behaviour: outages, system or application faults, and/or accidental destruction of information. Furthermore, information leaks through the use of questionable resources or methods during the penetration test process are also possible.

For this reason, a couple of steps will be taken towards minimizing the risk of tampering with the company’s data and/or accidentally dealing damage onto the system/s:

- Tools which claim to have a chance of leaving any form of negative impact (for example, data-loss and information leak) will not be used.
- If privileged access is to be gained anywhere, no tampering or other illegal behaviour will be engaged into.
- If one or more exploits prove to be successful, they will not be used to pivot towards finding new vulnerabilities and/or any form of information.

On the other hand, on the administrator's side, it is recommended to perform back-ups of all important data and configuration settings found on the web server, as well as any other essential resource which could be compromised during the penetration test.

3 Choice of penetration testing framework

For this penetration test process, part of the ISSAF [2, pp. 76-82] framework has been selected as it provides a strong introduction to the penetration testing process. The ISSAF framework is divided into 3 phases:

Phase I – Planning and Preparation

Phase II – Assessment

Phase III – Reporting, Clean-up, and Destroy Artifacts

Furthermore, the “OWASP Top 10” [3, pp. 382-414] framework will be used towards finding web application vulnerabilities. The OWASP Top 10 framework provides a sorted list of the most common web application vulnerabilities:

A1 – SQL Injection

A2 – Broken Authentication and Session Management

A3 – Cross-Site Scripting (XSS)

A4 – Insecure Direct Object References

A5 – Security Misconfiguration

A6 – Sensitive Data Exposure

A7 – Missing Function Level Access Control

A8 – Cross Site Request Forgery (CSRF)

A9 – Using Known Vulnerable Components

A10 – Unvalidated Redirects and Forwards

Finally, the report structure, as provided by the assignment requirements, will be used towards the analysis and discussion of found vulnerabilities.

The following chapters will detail and discuss the findings of the penetration testing process.

4 Choice of penetration testing tools

The OS used during the penetration testing process is a custom-built, 64 bit, XFCE-based Kali Linux distribution image installed locally onto an SSD drive. The Linux kernel version is 4.19.13.

As regards to the tools used during the penetration testing process, a list is presented below:

Info gathering – dig, dmitry, dnsmap, dnsrecon, fierce, smtp-enum, google hacking, host, nmap, automater, theharvester, urlcrazy, p0f, 0trace, in-trace, tracert, smtp-user-enum, swaks, ssldump, sslscan, sslyze, tlssled, whois

Vulnerability analysis – hydra, nikto

Web application analysis – blindelephant, plecost, wpscan, dirb, hydra, wfuzz, metasploit, OWASP ZAP, skipfish, whatweb, xsser

If further analysis of the tools’ output is desired, the results are to be found in the attachment provided with the report. It is recommended to use “less -R” in case an output file’s contents seem garbled.

5 Reconnaissance

Domain – tevelcyber.com

Registrant name – Tevel Cyber Corps

Registrant organization – Tevel Cyber Corps Private Limited

Registrant country – India

Registrant phone – +91.9841756250

Registrant email – harish@tevelcyber.com

URL – https://www.tevelcyber.com

Web platform – WordPress 4.9.9

IP address – 192.185.39.52

Name servers – ns8201.hostgator.com, ns8202.hostgator.com

E-mail addresses – padma.r@tevelcyber.com, ram@tevelcyber.com, karthick.s@tevelcyber.com, harish@tevelcyber.com

Mail servers – alt1.aspmx.l.google.com, alt4.aspmx.l.google.com, aspmx.l.google.com, alt2.aspmx.l.google.com, alt3.aspmx.l.google.com

Open ports – ftp, rsftp, domain, http, pop3, imap, https, smtps, submission, imaps, pop3s, EtherNetIP-1, mysql, http-proxy, https-alt

Filtered ports – ssh, smtp, msrpc, netbios-san, microsoft-ds, h323q931

OS details – DD-WRT v23 (Linux 2.4.36), MikroTik RouterOS 5.25 (Linux 2.6.35), Netgear ReadyNAS Duo NAS device (RAIDiator 4.1.4).

A “nmap” OS scan suggests that the device type of the system is “WAP [router]storage-misc”.

Furthermore, a scan with the “-sV” flag reveals a CPE which identifies the system as running RedHat Enterprise Linux 6.

6 Data compromise: SQL injection

Attack tools involved – OWASP ZAP, wfuzz

Protocols involved in the attack – SQL, HTTP

Description – OWASP ZAP suggests several pages of the tevelcyber.com webpage to be vulnerable under two forms of SQL injection:

- query' OR '1'='1'–
- robots.txt AND 8436=8436– PieL

SQL is the language used for communicating with a database. It can be used to read, write, delete, or edit database entries (SELECT, INSERT, DELETE, UPDATE). To this purpose users issue “SQL statements”.

However, flaws in either the SQL interpreter or the way user input is validated can give attackers a method of compromising the database through what are called “SQL injection” attacks.

For example, if a malicious user enters crafted input into the login page of a website, he/she could virtually enter any SQL statement, thus having the ability to read, edit, or delete the entire database. [4, pp. 178-179]

How the attack was launched – To further probe for the existence of SQL injection vulnerabilities, the “wfuzz” application was used along with a wordlist comprising of SQL injection payloads.

Success or failure – Failure. None of the payloads returned any significant result.

Solution – OWASP ZAP suggests the following:

- Do not trust client side input, even if there is client side validation in place.
- In general, type check all data on the server side.
- If the application uses JDBC, use PreparedStatement or CallableStatement, with parameters passed by '?'
- If the application uses ASP, use ADO Command Objects with strong type checking and parameterized queries.
- If database Stored Procedures can be used, use them.
- Do **not** concatenate strings into queries in the stored procedure, or use 'exec', 'exec immediate', or equivalent functionality!
- Do not create dynamic SQL queries using simple string concatenation.
- Escape all data received from the client.
- Apply a 'whitelist' of allowed characters, or a 'blacklist' of disallowed characters in user input.
- Apply the principle of least privilege by using the least privileged database user possible.

- In particular, avoid using the 'sa' or 'db-owner' database users. This does not eliminate SQL injection, but minimizes its impact.
- Grant the minimum database access that is necessary for the application.

7 Session hijacking: session cookie without the “HTTPOnly” flag

Attack tools involved – OWASP ZAP

Protocols involved in the attack – HTTPS, HTTP

Description – OWASP ZAP reports the “PHPSESSID” session cookie to be lacking the “HTTPOnly” flag.

The “HTTPOnly” flag prevents JavaScript code from accessing cookie data, whilst still allowing a browser to automatically attach cookies to requests. [5, p. 35] As such, it prevents a cookie from being accessed via non-HTTP calls such as JavaScript methods from third-party or custom applications. [6, p. 114]

Without the “HTTPOnly” flag, important cookies such as session cookies can be read and tampered with using JavaScript, for example, during an XSS and CSRF attack.

How the attack was launched – See below.

Success or failure – Failure. This vulnerability can only be used profitably if the malicious user happens to be the owner/administrator of one of the third-party scripts used on the web-page.

Solution – Ensure that all cookies or all cookies with sensitive information have the “HTTPOnly” flag set.

8 Session hijacking: session cookie without the “secure” flag

Attack tools involved – OWASP ZAP

Protocols involved in the attack – HTTPS, HTTP

Description – OWASP ZAP reports that the session cookie “PHPSESSID” lacks the “secure” flag.

The “secure” cookie flag tells a browser to not submit cookies anywhere as the result of un-encrypted HTTP connections. As a result, cookies without the “secure” flag on are much more liable to session hijacking or man-in-the-middle attacks if the client chooses to transmit sensitive information (for example, session cookies) using the encryption-less HTTP protocol. [7, p. 31]

How the attack was launched – See below.

Success or failure – Failure. This vulnerability can be used profitably only if the malicious user succeeds in placing himself between a privileged user (for example, the web-page’s administrator) and the web server.

Solution – Ensure that all cookies or all cookies with sensitive information have the “secure” flag set.

9 Arbitrary command execution: Yoast SEO

Attack tools involved – wpscan

Protocols involved in the attack – HTTP, HTTPS

Description – The WordPress scanning tool “wpscan” reports that it has detected an older version of the Yoast SEO plugin: 7.8, with the latest version being 9.3. Furthermore, versions lower than 9.2 have been reported as having an “authenticated race condition” vulnerability.

This vulnerability allows an owner of a WordPress account with the role of SEO Manager to arbitrarily execute commands on the server’s underlying OS via a .zip file import action, thus potentially compromising the entire server.

How the attack was launched – A proof of concept of the vulnerability can be found at the following link:
<https://www.youtube.com/watch?v=nL141dcDGCY>

Success or failure – Failure, as the penetration tester does not own any SEO Manager-role WordPress account.

Solution – Update the existing Yoast SEO plugin to version 9.2 or higher.

10 Click-jacking: X-Frame-Options header is not defined

Attack tools involved – nikto

Protocols involved in the attack – HTTP, HTTPS

Description – Nikto reports the tevelcyber.com domain as being vulnerable to click-jacking attacks.

Clickjacking [8, p. 141] is defined as a technique through which the user is maliciously deceived towards clicking invisible elements or links found on a web-page.

This function is made possible with the help of the “iframe” HTML element, which inserts a second HTML document inside an already existing HTML document, as well as CSS styling so as to make the malicious HTML element invisible to the end-user.

A standard method of dealing with click-jacking is through the use of the HTTP header “X-Frame-Options“, which explicitly instructs the client

browser on how to deal with “iframe” elements inside web-pages. This header has not been found inside of Tevel Cyber HTTP responses, thus leaving users to a risk of click-jacking if iframe embedding is possible.

How the attack was launched – See below.

Success or failure – Failure, for the same reasons stated with regards to the “HTTPOnly” flag. Only an owner of a third-party script could silently add invisible links on the page using JavaScript code.

Otherwise, it seems that this vulnerability cannot be exploited, as there is no way of injecting other third-party scripts into the web-page.

Solution – Define the “X-Frame-Options” header [8, p. 156], with the optional value being set to “DENY”. The “DENY” will instruct the browser to refuse to load any iframes found in web-pages retrieved from tevelcyber.com.

Alternatively, if the use of iframes is necessary or desired (for example, towards embedding documents from the same domain), set the optional value to “SAMEORIGIN”.

11 XSS: X-XSS-Protection header is not defined

Attack tools involved – nikto

Protocols involved in the attack – HTTP, HTTPS

Description – Nikto reports the tevelcyber.com domain to be lacking a definition of the “X-XSS-Protection” header.

The “X-XSS-Protection” response header can be found in the IE, Chrome and Safari web browsers. This header, once set, helps these browsers stop pages from loading once they have been detected as containing reflected cross-site scripting (XSS) attacks.

A reflected cross-site scripting attack is defined [9, p. 128] as a malicious attack that makes use of the fact that vulnerable web software “reflects” or re-executes code input by the user. Thus, pages with malicious parameters or JavaScript code can be crafted with the purpose of fooling users into running malicious code on their browser.

As the X-XSS-Protection header is not defined on the Tevel Cyber website, a malicious user could craft special links with a malicious purpose (for example, collecting the user’s session cookie) and use it in a phishing attack.

How the attack was launched – N/A

Success or failure – N/A

Solution – Define the “X-XSS-Protection” header [10, p. 256] with either of the following values:

1 Enables the cross-site scripting filter.

1; mode = block Blocks the loading of the web-page as a whole if a cross-site scripting attack is detected.

12 Arbitrary command execution: X-Content-Type-Options

Attack tools involved – nikto

Protocols involved in the attack – HTTP, HTTPS

Description – The “X-Content-Type-Options” response header [11, p. 14], combined with its “nosniff” directive, has as its purpose of telling the client’s browser to not try to guess the type of content received, and, subsequently, to not automatically perform any specific action that is mapped to that specific file-type. Instead, the content is interpreted by the browser in accordance to the content’s “content-type” header.

Without the implementation of this header and directive, the browser may be tricked into wrongly interpreting, and, subsequently, acting upon a retrieved file’s type. For example, a seemingly-safe malicious JPEG image could get interpreted as malicious JavaScript code.

How the attack was launched – N/A

Success or failure – N/A

Solution – Define the “X-Content-Type-Options” with the “nosniff” directive.

13 Man-in-the-middle: no HSTS support

Attack tools involved – nikto

Protocols involved in the attack – HTTP, HTTPS

Description – Nikto reports the tevelcyber.com domain to be lacking in HSTS support.

“HTTP Strict Transport Security” (HSTS) [12, pp. 145-147] is a security mechanism which enforces users accessing a web-page to transmit data through the more secure HTTPS protocol.

With HSTS, standard requests to a HTTP web-page are first redirected towards their HTTPS equivalent. A secure connection is then established. Finally, the client web browser now receives the HTTP “Strict-Transport-Security” header, which instructs the client that the web server is an HSTS host, as well as for how long the browser should regard it as such with the help of the “max-age” value.

This mechanism is enabled by defining the “Strict-Transport-Security” response header. Next, the “max-age” value specifies for how long since the client’s browser has received the initial HSTS response header should it regard the host as an HSTS host. Finally, the “includeSubDomains” directive tells the browser that HSTS should be applied to all of the host’s subdomains.

How the attack was launched – N/A

Success or failure – N/A

Solution – Enforce an HSTS policy across the domain and all subdomains by defining a “Strict-Transport-Security” header.

The response header can be defined as follows: “Strict-Transport-Security: max-age=10000000; IncludeSubDomains”.

The “max-age” directive value in this case is of 10000000 seconds, which roughly corresponds to 4 months.

The “IncludeSubDomains” directive enforces that the HSTS policy be applied to all of the host’s subdomains.

14 Information disclosure: file disclosure

Attack tools involved – OWASP ZAP

Protocols involved in the attack – N/A

Description – OWASP ZAP reports that numerous file back-ups were disclosed by the web server, containing “old” and “bak” suffix. Some of those files include “wp-admin”, “robots.txt”, “admin-ajax.php” and “sitemap”. The total number of instances reported is a significant number of 35.

This information can be very useful to a malicious user who desires to break into the system and learn more sensitive information about the web server.

How the attack was launched – N/A

Success or failure – N/A

Solution – OWASP ZAP recommends to “not edit files in-site on the web server, and ensure that un-necessary files (including hidden files) are removed from the web server.”

15 Information disclosure: “ramwptest”

Attack tools involved – wpscan

Protocols involved in the attack – N/A

Description – The “ramwptest” WordPress administrative account has been found through the use of wpscan.

This presents a significant issue as this eases a malicious user’s attempt to enumerate valid WordPress administrative accounts, as well as towards the attempt of brute-forcing administrative access to the web-page through the WordPress administrative log-in page, “wp-login.php”.

Furthermore, after further investigation, it has been found out that the username discloses the account’s owner. This could aid towards a social engineering attack and/or towards more useful information which could help a malicious user construct a more successful password word-list for further password brute-forcing attacks.

How the attack was launched – N/A

Success or failure – N/A

Solution – It is recommended, first of all, to change the username to one that does not reveal any sensitive information, such as its owner. Furthermore, a very strong password is recommended.

16 Information disclosure: web-page full-path disclosure

Attack tools involved – wpscan

Protocols involved in the attack – N/A

Description – A visit to the “tevelcyber.com/wp-settings.php” web-page outputs an error which discloses potentially useful information: it reveals that the Tevel Cyber web-page is located inside a user’s “home” folder with the name of “kalakkad”.

This information could be useful to an attacker who wishes to enumerate valid Linux user accounts for further exploitation and/or password brute-forcing attempts.

How the attack was launched – N/A

Success or failure – N/A

Solution – It is recommended to disable the output of any sensitive information whenever a web-page visit is erroneous. As a general practice it is highly recommended [13] to disable any sort of debugging-related or informational output from programs across the entire web application stack once a web application is in production status.

17 Timetable for execution of solution

Below is presented a rough timeline of the penetration process which includes the length of the penetration testing process, as well as the possible length of time required towards the implementation of each solution:

Process	Time
Penetration testing	45h
SQL injection solution	8h to 16h
Secure flags solution	1h
WordPress plugin update solution	10m
Header customization solution	3h
HSTS solution	1h
File disclosure solution	30m
Web-page full-path disclosure solution	30m
Total time of solution implementation	14h:10m to 22h:10m
Total	59h:10m to 67h:10m

18 Cost of Solution

All of the above solutions do not require the acquisition of new non-free software or new hardware. As such, the cost required towards the implementation of the solution rests solely on the amount of work the penetration tester carried during the penetration testing process, as well as the work the penetration tester may carry towards the implementation of a solution for each found vulnerability.

The proposed hourly rate for the work undertaken is 10£. As such:

- The proposed cost of the penetration testing process equals to 450£.
- The proposed cost of the solutions' implementation equals from roughly 140£ to 220£, depending mostly on how much time it will take for the SQL injection-related solution/s to be implemented.
- In total, the proposed cost for the entire process equals from roughly 590£ to 670£.

19 Conclusion

It can be asserted, judging from the information retrieved up to this point, that the tevelcyber.com web server is fairly robust. On the other hand, however, it can also be said that it presents three significant attack vectors:

- The administrative WordPress account “ramwptest”, which discloses its owner, thus potentially opening up the Tevel Cyber company to social engineering attacks in an attempt to gain access to the account.
- Using the same method, a malicious hacker, once placed inside the company's or IT staff's boundaries, could potentially intercept the “PHPSESSID” session cookie and hijack it for malicious purposes, such as compromising an administrative account.
- Finally, the “wp-settings.php” web-page revealing a Linux user account, which helps a malicious hacker in narrowing his scope with regards to brute-force attacks or the exploitation of the Linux system.

However, it is important to note that the bulk of this report stems from analysis provided by the tools “nikto”, “OWASP ZAP” and “wpscan”, which are aimed primarily towards web-page vulnerability scanning. It is not certain whether or not they are 100% accurate. As such, false positives and false negatives may occur.

Details of the penetration testing process are to be found in the attachment to this report.

To conclude, a computer system is never secure. [14, p. 544] As such, it is vitally important to perform regular back-ups and implement security mechanisms which ensure that the minimum amount of damage is possible in the case of a security compromise.

References

- [1] J. Tiller, *CISO's Guide to Penetration Testing: A Framework to Plan, Manage, and Maximize Benefits*. CRC Press, 2016. [Online]. Available: <https://books.google.ro/books?id=dLrMBQAAQBAJ>
- [2] T. Wilhelm, *Professional Penetration Testing: Creating and Learning in a Hacking Lab*. Elsevier Science, 2013. [Online]. Available: https://books.google.ro/books?id=K_-Di3khZHAC
- [3] M. Posadas, *Mastering C# and .NET Framework*. Packt Publishing, 2016. [Online]. Available: <https://books.google.ro/books?id=VMzcDgAAQBAJ>
- [4] D. Gibson, *SSCP Systems Security Certified Practitioner All-in-One Exam Guide, Second Edition*. McGraw-Hill Education, 2015. [Online]. Available: <https://books.google.ro/books?id=10adCgAAQBAJ>
- [5] J. Manico and A. Detlefsen, *Iron-Clad Java: Building Secure Web Applications*, ser. Oracle Press. McGraw-Hill Education, 2014. [Online]. Available: https://books.google.ro/books?id=iY_zAwAAQBAJ
- [6] P. Goodey, *Salesforce CRM: The Definitive Admin Handbook*. Packt Publishing, 2013. [Online]. Available: <https://books.google.ro/books?id=lOoNr4eZ0ooC>
- [7] R. Lepofsky, *The Manager's Guide to Web Application Security: A Concise Guide to the Weaker Side of the Web*, ser. Expert's voice in security. Apress, 2014. [Online]. Available: <https://books.google.ro/books?id=0VQnCcAAQBAJ>
- [8] H. Wu and L. Zhao, *Web Security: A White-Hat Perspective*. CRC Press, 2015. [Online]. Available: <https://books.google.ro/books?id=TQEZCwAAQBAJ>
- [9] P. Hope and B. Walther, *Web Security Testing Cookbook: Systematic Techniques to Find Problems Fast*. O'Reilly Media, 2008. [Online]. Available: https://books.google.ro/books?id=VmrSJ3V-s_MC
- [10] M. Heiderich, E. Nava, G. Heyes, and D. Lindsay, *Web Application Obfuscation: '-/WAFs..Evasion..Filters//alert(/Obfuscation/)-'*. Elsevier Science, 2010. [Online]. Available: <https://books.google.ro/books?id=Znxa3zrJWJsC>
- [11] W. Alcorn, C. Frichot, and M. Orru, *The Browser Hacker's Handbook*. Wiley, 2014. [Online]. Available: <https://books.google.ro/books?id=1tjjAgAAQBAJ>
- [12] I. Ristic, *Bulletproof SSL and TLS: Understanding and Deploying SSL/TLS and PKI to Secure Servers and Web Applications*, ser. Computers / Security. Feisty Duck, 2013. [Online]. Available: <https://books.google.ro/books?id=fQOLBAAAQBAJ>
- [13] B. Hoffman and B. Sullivan, *Ajax Security*. Pearson Education, 2007. [Online]. Available: https://books.google.ro/books?id=AIm_LV5a528C

- [14] J. Vacca, *Computer and Information Security Handbook*. Elsevier Science, 2012. [Online]. Available: <https://books.google.ro/books?id=zb916YOr16wC>