Fortify Tech
Security Assessment Findings Report

Nicholas Marco Weinandra
5027221042
Ethical Hacking A

Business Confidential

Date: May 8th, 2024
Prakticum 2 Ethical Hacking
Version 1.0

Table of Contents

Confidentiality Statement

This document is the exclusive property of Fortify Tech. This document contains proprietary and confidential information. Duplication, redistribution, or use, in whole or in part, in any form, requires consent of Fortify Tech.

Fortify Tech may share this document with auditors under non-disclosure agreements to demonstrate penetration test requirement compliance.

Disclaimer

A penetration test is considered a snapshot in time. The findings and recommendations reflect the information gathered during the assessment and not any changes or modifications made outside of that period.

Time-limited engagements do not allow for a full evaluation of all security controls. Fortify Tech prioritized the assessment to identify the weakest security controls an attacker would exploit. Fortify Tech recommends conducting similar assessments on an annual basis by internal or third-party assessors to ensure the continued success of the controls.

Contact Information

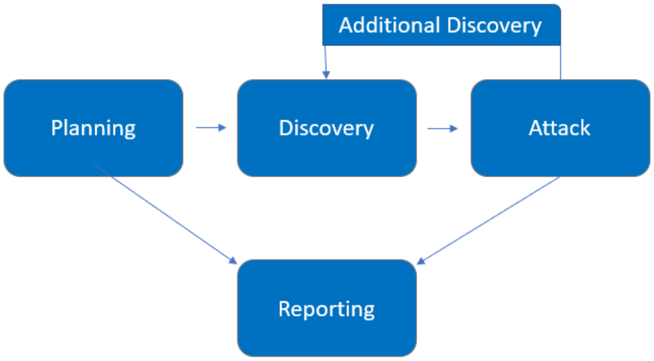
Name	Title	Contact Information
Author		
Nicholas Marco Weinandra	Lead Penetration Tester	nicoweinandra@gmail.com

Assessment Overview

From May 5 2024 – May 8 2024, DC engaged TCMS to evaluate the security posture of its infrastructure compared to current industry best practices that included an external penetration test. All testing performed is based on the NIST *SP 800-115 Technical Guide to Information Security Testing and Assessment, OWASP Testing Guide (v4), and customized testing frameworks*.

Phases of penetration testing activities include the following:

- Planning – Customer goals are gathered and rules of engagement obtained.
- Discovery – Perform scanning and enumeration to identify potential vulnerabilities, weak areas, and exploits.
- Attack – Confirm potential vulnerabilities through exploitation and perform additional discovery upon new access.
- Reporting – Document all found vulnerabilities and exploits, failed attempts, and company strengths and weaknesses.



Assessment Components

External Penetration Test

An external penetration test emulates the role of an attacker attempting to gain access to an internal network without internal resources or inside knowledge. A TCMS engineer attempts to gather sensitive information through open-source intelligence (OSINT), including employee information, historical breached passwords, and more that can be leveraged against external systems to gain internal network access. The engineer also performs scanning and enumeration to identify potential vulnerabilities in hopes of exploitation.

Finding Severity Ratings

The following table defines levels of severity and corresponding CVSS score range that are used throughout the document to assess vulnerability and risk impact.

Severity	CVSS V3 Score Range	Definition
Critical	9.0-10.0	Exploitation is straightforward and usually results in system-level compromise. It is advised to form a plan of action and patch immediately.
High	7.0-8.9	Exploitation is more difficult but could cause elevated privileges and potentially a loss of data or downtime. It is advised to form a plan of action and patch as soon as possible.
Moderate	4.0-6.9	Vulnerabilities exist but are not exploitable or require extra steps such as social engineering. It is advised to form a plan of action and patch after high-priority issues have been resolved.
Low	0.1-3.9	Vulnerabilities are non-exploitable but would reduce an organization's attack surface. It is advised to form a plan of action and patch during the next maintenance window.
Informational	N/A	No vulnerability exists. Additional information is provided regarding items noticed during testing, strong controls, and additional documentation.

Scope

Assessment	Details
External Penetration Test	10.15.42.36, 10.15.42.7

Executive Summary

Nicholas evaluated Fortify Tech's security between the scopes provided through an external network penetration test from May 5 2024 to May 8 2024. By performing a series of scanning and enumeration, Nicholas found high lever vulnerabilities that allowed attackers to send files through the FTP server. It is highly recommended that Fortify address these vulnerabilities as soon as possible according to the table below.

Furthermore, the login page of the Wordpress 10.15.42.7/wp-login.php is highly vulnerable to bruteforce attack due to the lack of security preventions. Given enough time and resources, an attacker could easily try to gain access to the admin role by trying a vast amount of possible passwords. It is recommended that Fortify Tech to take preventive measures listed in the recommendation column below.

Aditionally, through scanning the IP address, I found that both the IP addresses—10.15.42.7 and 10.15.42.36 are vulnerable to Terappin. The SSH transport protocol with certain OpenSSH extensions, found in OpenSSH before 9.6 and other products, allows remote attackers to bypass integrity checks such that some packets are omitted (from the extension negotiation message), and a client and server may consequently end up with a connection for which some security features have been downgraded or disabled, aka a Terrapin attack. I highly recommend Fortify to patch this security breach immediately, as attackers could easily access sensitive information located in the backup.sql file.

Attack Summary

The following table describes how I gained internal network access, step by step:

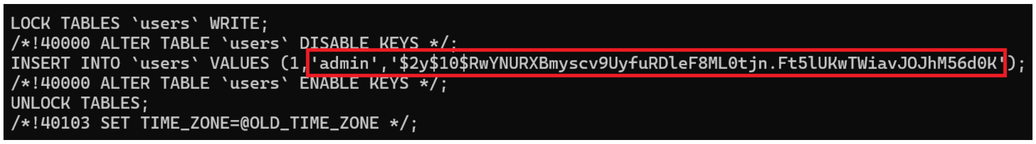
Step	Action	Recommendation
1	**Gained access to FTP server.** Here, I found a SQL database containing a username and password(hashed) that may or may not be sensitive information.	Disable anonymous access to the FTP server. Ensure that the server is configured to require authentication (username and passowrd) for accessing files.
2	**Performed bruteforce on website login system** (10.15.42.7/wp-login.php and 10.15.42.36:8888). Although I failed to gain access to admin due to time limitation, this is still a critical vulnerability. Attackers with enough time and resources might successfully gain access to a higher role.	It is highly recommended to use a plugin that limits the number of login attempts from a single IP address within a specific time frame. This helps prevent bruteforce attacks.

3	Found **CVE-2023-48795 - Vulnerable to Terappin**. The SSH transport protocol with certain OpenSSH extensions, found in OpenSSH before 9.6 and other products, allows remote attackers to bypass integrity checks such that some packets are omitted (from the extension negotiation message), and a client and server may consequently end up with a connection for which some security features have been downgraded or disabled, aka a Terrapin attack.	Updated client and server provide kex pseudo-algorithms indicating usage of the updated version of the protocol which is protected from the attack. If "kex-strict-c-v00@openssh.com" is provided by clients and "kex-strict-s-v00@openssh.com" is in the server's reply, you use the latest version and are safe, otherwise you may want to disable vulnerable ciphers via crypto policy. Furthermore, please go to https://access.redhat.com/security/cve/cve-2023-48795 for a complete mitigation.

Security Strengths

Hashed Password found in FTP

During the assessment, I found that the password in the backup.sql file is hashed using bycrypt. By hashing passwords, even if the database is compromised, attackers cannot directly retrieve the original passwords. Instead, they would need to crack the hashed passwords, which is significantly more difficult than simply obtaining plaintext passwords.



```
LOCK TABLES `users` WRITE;
/*!40000 ALTER TABLE `users` DISABLE KEYS */;
INSERT INTO `users` VALUES (1,'admin','$2y$10$RwYNURXBmyscv9UyfuRDleF8ML0tjn.Ft5lUKwTWiavJOJhM56d0K');
/*!40000 ALTER TABLE `users` ENABLE KEYS */;
UNLOCK TABLES;
/*!40103 SET TIME_ZONE=@OLD_TIME_ZONE */;
```

Security Weaknesses

FTP Server Detection

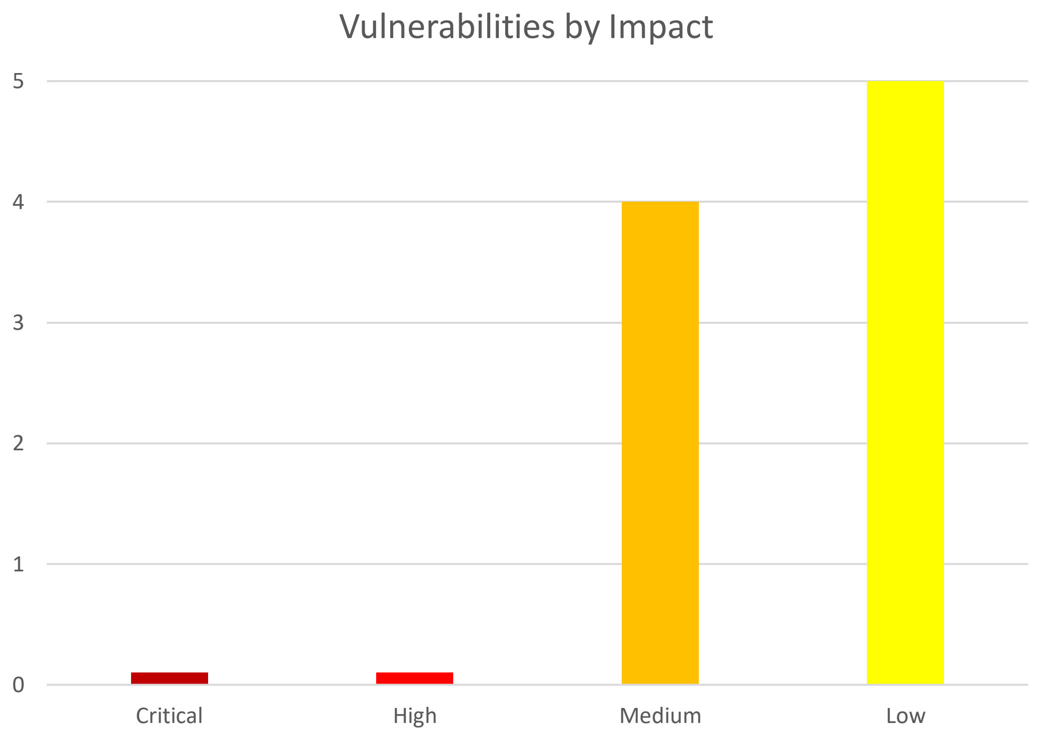
An FTP server is listening on a remote port. It is possible to obtain the banner of the remote FTP server by connecting to a remote port.

Vulnerable to Terappin

The SSH transport protocol with certain OpenSSH extensions, found in OpenSSH before 9.6 and other products, allows remote attackers to bypass integrity checks such that some packets are omitted (from the extension negotiation message), and a client and server may consequently end up with a connection for which some security features have been downgraded or disabled, aka a Terrapin attack.

Vulnerabilities by Impact

The following chart illustrates the vulnerabilities found by impact:



External Penetration Test Findings

- Through scanning by using OWASP-ZAP, Nuclei, and Nessus, there are several security vulnerabilities found on the server.

1. Content Security Policy (CSP) Header Not Set (Medium)

Description:	Content Security Policy (CSP) is an added layer of security that helps to detect and mitigate certain types of attacks, including Cross Site Scripting (XSS) and data injection attacks. These attacks are used for everything from data theft to site defacement or distribution of malware. CSP provides a set of standard HTTP headers that allow website owners to declare approved sources of content that browsers should be allowed to load on that page — covered types are JavaScript, CSS, HTML frames, fonts, images and embeddable objects such as Java applets, ActiveX, audio and video files.
Impact:	Medium
System:	- 10.15.42.7 - 10.15.42.36:8888
References:	- OWASP_2021_A05 - OWASP_2017_A06 - CWE-693

2. Missing Anti-Clickjacking Header (Medium)

Description:	The response does not include either Content-Security-Policy with 'frame-ancestors' directive or X-Frame-Options to protect against 'ClickJacking' attacks.
Impact:	Medium
System:	- 10.15.42.7 - 10.15.42.36:8888
References:	- CWE-1021

3. CVE-2023-48795 - Vulnerable to Terappin (Medium)

Description:	The SSH transport protocol with certain OpenSSH extensions, found in OpenSSH before 9.6 and other products, allows remote attackers to bypass integrity checks such that some packets are omitted (from the extension negotiation message), and a client and server may consequently end up with a connection for which some security features have been downgraded or disabled, aka a Terrapin attack. This occurs because the SSH Binary Packet Protocol (BPP), implemented by these extensions, mishandles the handshake phase and mishandles use of sequence numbers.

Impact:	Medium
System:	- 10.15.42.7 - 10.15.42.36:22
References:	- CVE-2023-48795

4. FTP Server Detection (Medium)

Description:	An FTP server is listening on a remote port.
Impact:	Medium
System:	- 10.15.42.36
References:	- FTP Server Detection

5. Absence of Anti-CSRF Tokens (Low)

Description:	No Anti-CSRF tokens were found in a HTML submission form. A cross-site request forgery is an attack that involves forcing a victim to send an HTTP request to a target destination without their knowledge or intent in order to perform an action as the victim. The underlying cause is application functionality using predictable URL/form actions in a repeatable way. The nature of the attack is that CSRF exploits the trust that a web site has for a user. By contrast, cross-site scripting (XSS) exploits the trust that a user has for a web site. Like XSS, CSRF attacks are not necessarily cross-site, but they can be. Cross-site request forgery is also known as CSRF, XSRF, one-click attack, session riding, confused deputy, and sea surf. CSRF attacks are effective in a number of situations, including: * The victim has an active session on the target site. * The victim is authenticated via HTTP auth on the target site. * The victim is on the same local network as the target site. CSRF has primarily been used to perform an action against a target site using the victim's privileges, but recent techniques have been discovered to disclose information by gaining access to the response. The risk of information disclosure is dramatically increased when the target site is vulnerable to XSS, because XSS can be used as a platform for CSRF, allowing the attack to operate within the bounds of the same-origin policy.
Impact:	Low
System:	- 10.15.42.7 - 10.15.42.36:8888
References:	- CWE-352

6. Server Leaks Version Information via "Server" HTTP Response Header Field (Low)

Description:	The web/application server is leaking version information via the "Server" HTTP response header. Access to such information may facilitate attackers identifying other vulnerabilities your web/application server is subject to.
Impact:	Low
System:	- 10.15.42.7 - 10.15.42.36:8888
References:	- CWE-200

7. Cookie No HttpOnly Flag (Low)

Description:	A cookie has been set without the HttpOnly flag, which means that the cookie can be accessed by JavaScript. If a malicious script can be run on this page then the cookie will be accessible and can be transmitted to another site. If this is a session cookie then session hijacking may be possible.
Impact:	Low
System:	10.15.42.7
References:	- CVE-1004

8. Cookie without SameSite Attribute (Low)

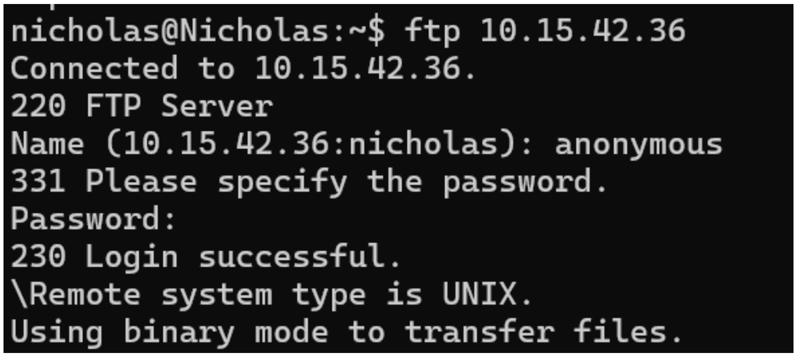
Description:	A cookie has been set without the SameSite attribute, which means that the cookie can be sent as a result of a 'cross-site' request. The SameSite attribute is an effective counter measure to cross-site request forgery, cross-site script inclusion, and timing attacks.
Impact:	Low
System:	10.15.42.7
References:	- CVE-1275

9. Server Leaks Information via "X-Powered-By" HTTP Response Header Field(s) (Low)

Description:	The web/application server is leaking information via one or more "X-Powered-By" HTTP response headers. Access to such information may facilitate attackers identifying other frameworks/components your web application is reliant upon and the vulnerabilities such components may be subject to.
Impact:	Low
System:	10.15.42.7
References:	- CWE-200

Exploitation Proof of Concept

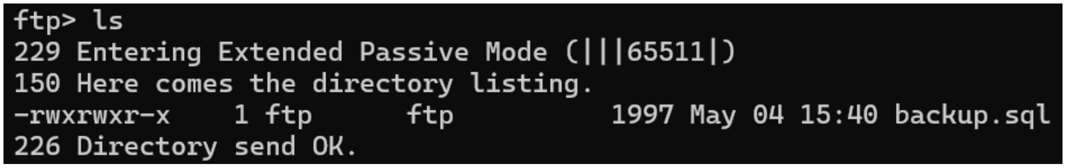
- I successfully gained access to the FTP server of 10.15.42.36 using the command ftp 10.15.42.36. Gaining unauthorized access to the FTP server, can lead to various security risks such as data theft, data manipulation, or unauthorized file uploads/downloads.



```
nicholas@Nicholas:~$ ftp 10.15.42.36
Connected to 10.15.42.36.
220 FTP Server
Name (10.15.42.36:nicholas): anonymous
331 Please specify the password.
Password:
230 Login successful.
\Remote system type is UNIX.
Using binary mode to transfer files.
```

Figure 1: Connecting to the FTP Server 10.15.42.36

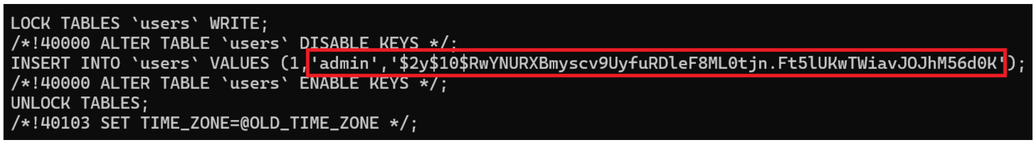
- Here, I found a directory containing a .sql file.



```
ftp> ls
229 Entering Extended Passive Mode (|||65511|)
150 Here comes the directory listing.
-rwxrwxr-x    1 ftp      ftp          1997 May 04 15:40 backup.sql
226 Directory send OK.
```

Figure 2: Listing the directories

- Next, I opened the backup.sql file and found a username and password. Although the password is hashed, it is important to ensure that only authorized users have access to the database.



```
LOCK TABLES `users` WRITE;
/*!40000 ALTER TABLE `users` DISABLE KEYS */;
INSERT INTO `users` VALUES (1,'admin','$2y$10$RwYNURXBmyscv9UyfuRDleF8ML0tjn.Ft5lUKwTWiavJOJhM56d0K');
/*!40000 ALTER TABLE `users` ENABLE KEYS */;
UNLOCK TABLES;
/*!40103 SET TIME_ZONE=@OLD_TIME_ZONE */;
```

Figure 3: Username and hashed password found in 'users' tables

Additional Reports and Scans (Informational)

To access additional information about this report, please refrain to my Github under the Additionals folder.
Github Link

Last Page