FortifyTech
Security Assessment Findings Report

Business Confidential

Date: May 8th, 2024
Project: DC-001
Version 1.0

Contents

Confidentiality Statement

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

FortifyTech 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. CyberShield prioritized the assessment to identify the weakest security controls an attacker would exploit. CyberShield 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
FortifyTech		
John Smith	Global Information Security Manager	Email: jsmith@democorp.com
CyberShield		
Marcelinus A	Lead Penetration Tester	Email: marcelalvin11@gmail.com

Assessment Overview

From May 5nd, 2024 to May 8th, 2024, FortifyTech engaged CyberShield to evaluate the security posture of its infrastructure compared to current industry best practices that included an internal network penetration test.

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..
- Reporting – Document all found vulnerabilities and exploits, failed attempts, and company strengths and weaknesses.

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.

Risk Factors

Risk is measured by two factors: Likelihood and Impact:

Likelihood

Likelihood measures the potential of a vulnerability being exploited. Ratings are given based on the difficulty of the attack, the available tools, attacker skill level, and client environment.

Impact

Impact measures the potential vulnerability's effect on operations, including confidentiality, integrity, and availability of client systems and/or data, reputational harm, and financial loss.

Scope

Assessment	Details
Internal Penetration Test	• 10.15.42.36 used as login purposes • 10.15.42.7 used as a landing page with wordpress as it's framework

Scope Exclusions

.

FortifyTech did not forbid any specified attacks form during testing

Client Allowances

FortifyTech did not provide CyberShield any forms of allowances.

Executive Summary

TCMS evaluated Demo Corp's internal security posture through penetration testing from February 22nd, 2021 to March 5th, 2021. The following sections provide a high-level overview of vulnerabilities discovered, successful and unsuccessful attempts, and strengths and weaknesses.

Scoping and Time Limitations

Scoping during the engagement did not permit denial of service or social engineering across all testing components.

Time limitations were in place for testing. Internal network penetration testing was permitted for three (3) business days.

Testing Summary

Marcel use various tools to recon target ip address. Nmap to search for accessible ports and services being used by FortifyTech, Gobuster to search all possible endpoints from scope that had been given to Marcel. And lastly the usage of both Nuclei and OWASP-ZAP to find vulnerabilities from targeted ip address.

Vulnerability Summary & Report Card

The following tables illustrate the vulnerabilities found by impact and recommended remediations:

Penetration Test Findings

0	0	4	5	0
Critical	High	Moderate	Low	Informational

Finding	Severity	Recommendation
Penetration Test		
Absence of Anti-CSRF Tokens	Moderate	**Implement Anti-CSRF Tokens**
Content Security Policy (CSP) Header Not Set	Moderate	**Set Content Security Policy (CSP) Header**
Missing Anti-clickjacking Header	Moderate	**Include Anti-clickjacking Header**
Terrapin Attack (CVE-2023-48795)	Moderate	**Update your SSH client and server to the latest versions**
Server Leaks Information via "X-Powered-By" HTTP Response Header Fields(s)	Low	**Remove Server Information from Headers**
Server Leaks Information via "Server" HTTP Response Header Fields	Low	**Enable X-Content-Type-Options Header**
X-Content-Type-Options Header Missing	Low	Apply the appropriate Microsoft patches to remediate the issue.
Cookie No HttpOnly Flag	Low	**Set HttpOnly Flag for Cookies**
Cookie without SameSite Attribute	Low	**Include SameSite Attribute for Cookies**

Technical Findings

FTP Server Detection

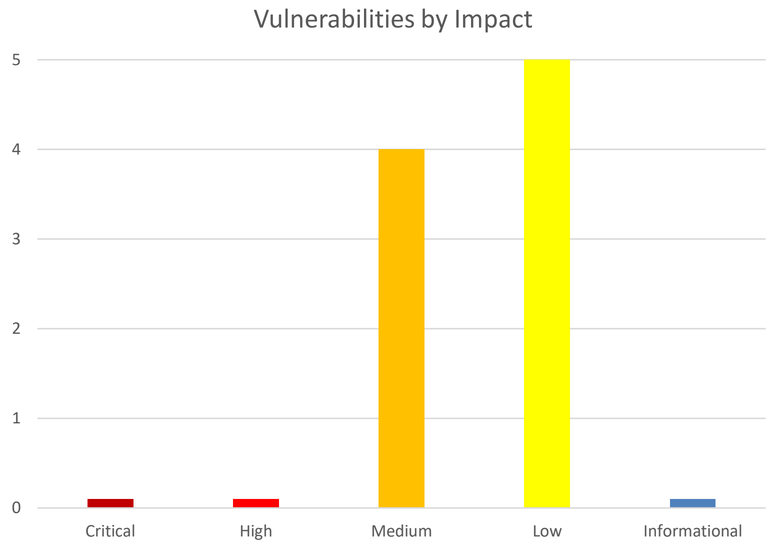
An FTP server is listening on a remote port. It is possible to obtain the banner of the remote FTP server by connecting to a remote port.

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, there are several security vulnerabilities found on the server.

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

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

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

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

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

Commented [A1]:

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

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

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

Commented [A2]:

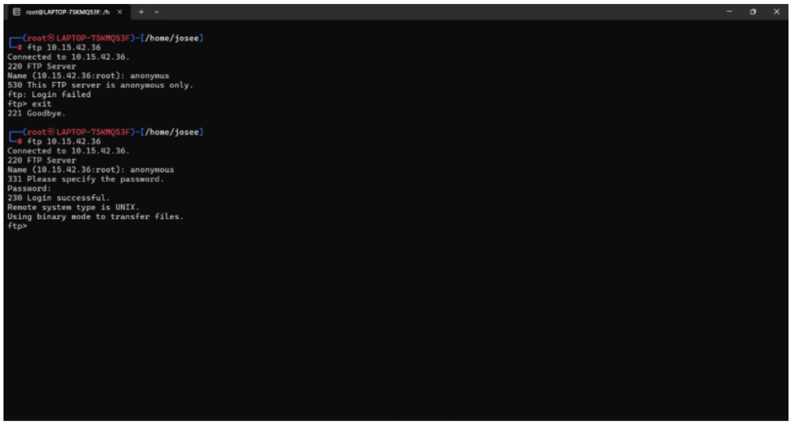
Server Leaks Information via "X-Powered-By" HTTP Response Header Field(s) (Low)

• 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

Commented [A3]:

Exploitation Proof of Concept

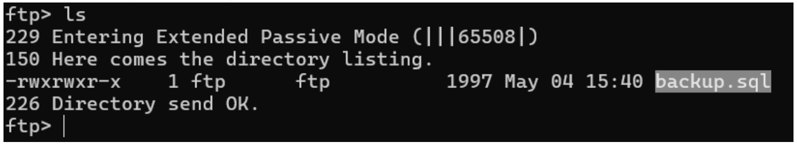
- I successfully gained access to the FTP server of 10.15.42.36 using the command ftp 10.15.42.36. Gaining unauthorized access to the FTP server, can lead to various security risks such as data theft, data manipulation, or unauthorized file uploads/downloads.



```
┌──(root㉿LAPTOP-75KMQ53F)-[/home/josee]
└─# ftp 10.15.42.36
Connected to 10.15.42.36.
220 FTP Server
Name (10.15.42.36:root): anonymus
530 This FTP server is anonymous only.
ftp: Login failed
ftp> exit
221 Goodbye.

┌──(root㉿LAPTOP-75KMQ53F)-[/home/josee]
└─# ftp 10.15.42.36
Connected to 10.15.42.36.
220 FTP Server
Name (10.15.42.36:root): anonymous
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp>
```

Figure 1: Connecting to the FTP Server 10.15.42.36

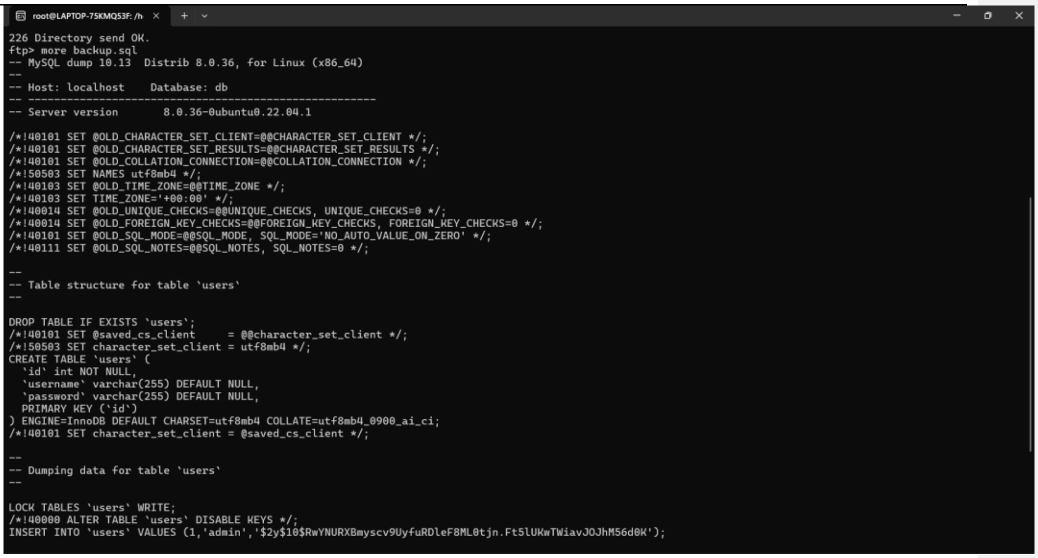
- Here, I found a directory containing a .sql file named backup.sql



```
ftp> ls
229 Entering Extended Passive Mode (|||65508|)
150 Here comes the directory listing.
-rwxrwxr-x    1 ftp      ftp          1997 May 04 15:40 backup.sql
226 Directory send OK.
ftp> |
```

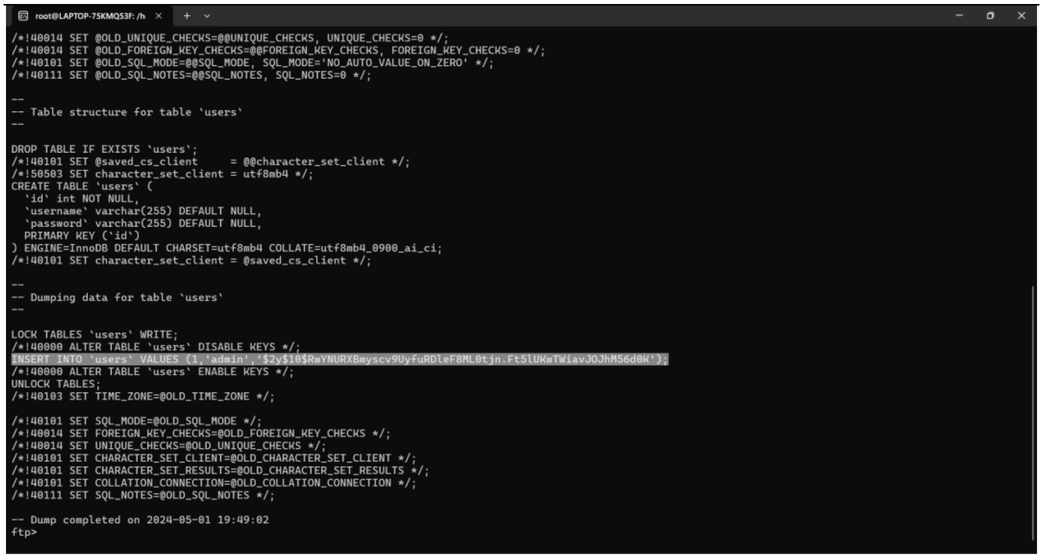
Figure 2: Listing the directories insides

- 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.



```
226 Directory send OK.
ftp> more backup.sql
-- MySQL dump 10.13  Distrib 8.0.36, for Linux (x86_64)
--
-- Host: localhost    Database: db
-- ------------------------------------------------------
-- Server version       8.0.36-0ubuntu0.22.04.1

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!50503 SET NAMES utf8mb4 */;
/*!40103 SET @OLD_TIME_ZONE=@@TIME_ZONE */;
/*!40103 SET TIME_ZONE='+00:00' */;
/*!40014 SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0 */;
/*!40014 SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0 */;
/*!40101 SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='NO_AUTO_VALUE_ON_ZERO' */;
/*!40111 SET @OLD_SQL_NOTES=@@SQL_NOTES, SQL_NOTES=0 */;

--
-- Table structure for table `users`
--

DROP TABLE IF EXISTS `users`;
/*!40101 SET @saved_cs_client     = @@character_set_client */;
/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE `users` (
  `id` int NOT NULL,
  `username` varchar(255) DEFAULT NULL,
  `password` varchar(255) DEFAULT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `users`
--

LOCK TABLES `users` WRITE;
/*!40000 ALTER TABLE `users` DISABLE KEYS */;
INSERT INTO `users` VALUES (1,'admin','$2y$10$RwYNURXBmyscv9UyfuRDleF8ML0tjn.Ft5lUKwTWiavJOJhM56d0K');
```

Figure 3.1: Structure of “users” table



```
/*!40014 SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0 */;
/*!40014 SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0 */;
/*!40101 SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='NO_AUTO_VALUE_ON_ZERO' */;
/*!40111 SET @OLD_SQL_NOTES=@@SQL_NOTES, SQL_NOTES=0 */;

--
-- Table structure for table `users`
--

DROP TABLE IF EXISTS `users`;
/*!40101 SET @saved_cs_client     = @@character_set_client */;
/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE `users` (
  `id` int NOT NULL,
  `username` varchar(255) DEFAULT NULL,
  `password` varchar(255) DEFAULT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `users`
--

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 */;

/*!40101 SET SQL_MODE=@OLD_SQL_MODE */;
/*!40014 SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS */;
/*!40014 SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS */;
/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
/*!40111 SET SQL_NOTES=@OLD_SQL_NOTES */;

-- Dump completed on 2024-05-01 19:49:02
ftp>
```

Figure 3.2: Username and hashed password found in ‘users’ tables

Additional Scans and Reports

TCMS provides all clients with all report information gathered during testing. This includes Nessus files and full vulnerability scans in detailed formats. These reports contain raw vulnerability scans and additional vulnerabilities not exploited by TCM Security.

The reports identify hygiene issues needing attention but are less likely to lead to a breach, i.e. defense-in-depth opportunities. For more information, please see the documents in your shared drive folder labeled “Additional Scans and Reports”.

Last Page