Le BonBon Croissant

Le BonBon Croissant Penetration Testing Report

January 7-8, 2022

[redacted]

Table of Contents

DOCUMENT CONTROL INFORMATION

Document Details	
Company:	Le BonBon Croissant
Document Title:	Le BonBon Croissant Penetration Test Report
Version:	1.0
Date Edited:	January 8, 2022
Authors:	[redacted]
Penetration-Testers:	[redacted]
Classification:	Confidential

Recipients	
Name:	Jim Joseph
Title:	Principal Security Engineer

EXECUTIVE SUMMARY

[redacted] performed a second penetration test on Le BonBon Croissant's warehouse network on January 7-8, 2022. The penetration test simulated an attack of an external/internal threat actor attempting to gain access to the Le BonBon Croissant warehouse network system. The purpose of the penetration test was to discover network strengths, vulnerabilities, and suggest remediations. The test also reevaluated the vulnerabilities found in the team's initial penetration test to help determine the effectiveness of Le BonBon Croissant's remediation efforts.

The consultants identified strengths including user input sanitization on API endpoints, up to date software, and usage of HTTPS on the network. These strengths significantly improve security across multiple points of the tested network. After revisiting findings from the initial test, the consultants found that removal of hosts from the network resolved 4 previous vulnerabilities.

[redacted] identified a total of 14 vulnerabilities within the scope of engagement, which are broken down by severity in the table below:

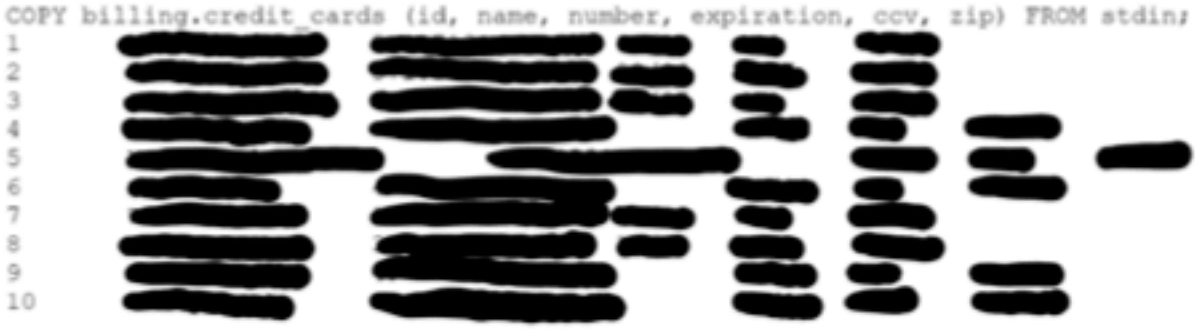
Critical	High	Medium	Low	Informational
2	3	3	3	3

The consultants identified a critical vulnerability in the system on a host with outdated ScadaBR software that allows for file upload and remote code execution. This system controls the heavy machinery of the Le BonBon Croissant Warehouse. With a successful exploitation, an attacker could change configurations and gain access to and control of the PLC devices. This could lead to system damage, physical damage to machinery, and potential loss of life resulting in expensive repairs and fines. Attackers choose to pursue this type of vulnerability to jeopardize a company's credibility, security, and stability.

Additionally, the consultants identified multiple vulnerabilities that led to the compromise of sensitive customer information such as credit card numbers. Aside from any potential reputational damage resulting from a breach of this network, Le BonBon Croissant could also experience significant financial losses through fines outlined in policies such as PCI DSS.

BUSINESS IMPACT

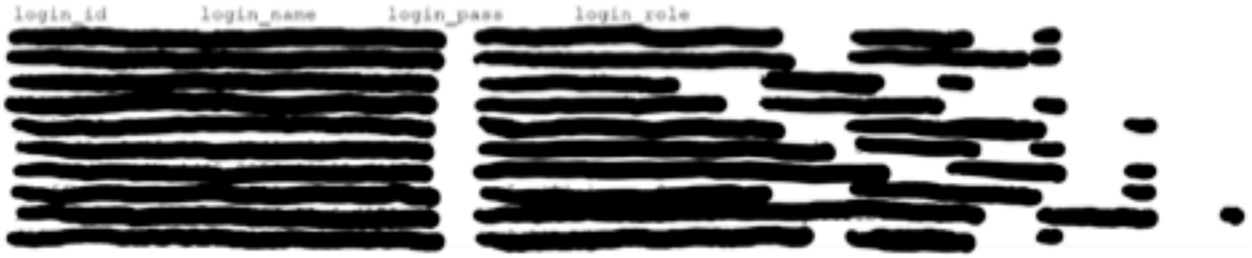
Upon conducting the consultant's test of the Le BonBon Croissant network, a few points of interest were discovered that may impact business operations. One notable finding was the storage of unencrypted and prohibited payment information in a vulnerable database that our team was able to breach during the penetration test. According to the Payment Card Industry Data Security Standard (PCI-DSS), primary account numbers (PAN) can be stored as long as they are rendered unreadable by using one of the methods outlined in section 3.4 of the PCI-DSS. The PCI-DSS also states that card verification values (CVV) cannot be stored, even in encrypted formats. Both unencrypted PAN and CVV information was found in Le BonBon Croissant's PostgreSQL database, which is non-compliant with PCI-DSS and can result in fines ranging from $5,000 to $100,000 per month.



```
COPY billing.credit_cards (id, name, number, expiration, ccv, zip) FROM stdin;
1	[redacted]
2	[redacted]
3	[redacted]
4	[redacted]
5	[redacted]
6	[redacted]
7	[redacted]
8	[redacted]
9	[redacted]
10	[redacted]
```

Image: Redacted unencrypted payment information from the PostgreSQL database.

Along with the unencrypted payment information found in the PostgreSQL database, our team found unsecure login credentials in Le BonBon Croissant's MariaDB database. After our team gained access to the login credentials stored in the vulnerable MariaDB database, it was discovered that the passwords were encoded using base64. Base64 is a well-known encoding technique that can easily be decoded using free and publicly available tools online. If information about Le BonBon Croissant's unsecure payment and credential information become publicly available, it could severely damage the reputation of the company. Customers are likely to avoid purchasing products from a company they cannot trust to protect their sensitive information, which will lead to a decrease in revenue.



```
login_id      login_name      login_pass      login_role
```

[redacted]

Image: Redacted login credential information with base64 encoded passwords from the MariaDB database.

Reference(s):

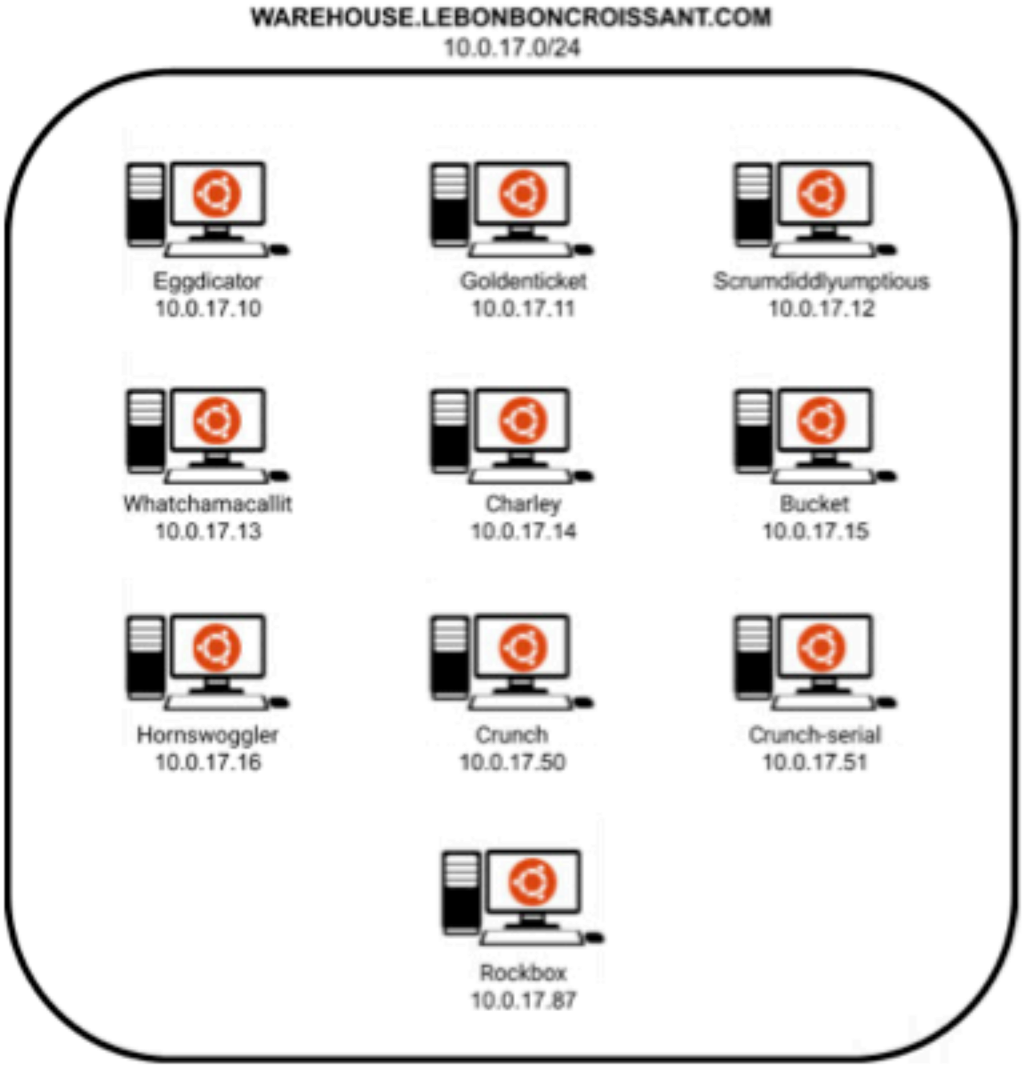
PCI Security Standards: https://www.pcisecuritystandards.org/document_library
PCI-DSS - Security Penalties: https://financial.ucsc.edu/pages/security_penalties.aspx
PCI-DSS Compliance Guide: https://www.pcicomplianceguide.org/faq/

TECHNICAL OVERVIEW

Scope

The scope of this penetration test included the IP range of 10.0.17.0/24 and any publicly available information including websites, social media, profiles, and GitHub repositories that pertain to Le BonBon Croissant. On the first day of testing the IP addresses 10.0.17.50 and 10.0.17.51 were excluded from the scope, but were included in testing on the second day with the approval of Le BonBon Croissant under careful consideration.

Network Topology



Key Strengths

[redacted] identified the following strengths within the Le BonBon Croissant warehouse network, and recommends continuing to maintain the existing level of security in the following areas:

I. Usage of input sanitization on API parameters

- The implementation of good input sanitization on API endpoints prevents injection attacks. This was observed during an in-depth testing of the FastAPI server located on 10.0.17.11.

II. Up-to-date software

- Most software on the Le BonBon Croissant warehouse network was running the latest version available. This helps prevent the exploitation of known vulnerabilities that have patches available.

III. Usage of HTTPS on API endpoints

- The usage of HTTPS on all API endpoints ensures the identity of the server and maintains secure connections.

Key Findings and Recommendations

I. Network Segmentation

- The implementation of network segmentation would separate systems that do not need to be connected.
- All systems within the network were directly accessible, so critical systems like the PLC controller and ScadaBR server could be placed behind a separate network than other systems.

II. Authentication

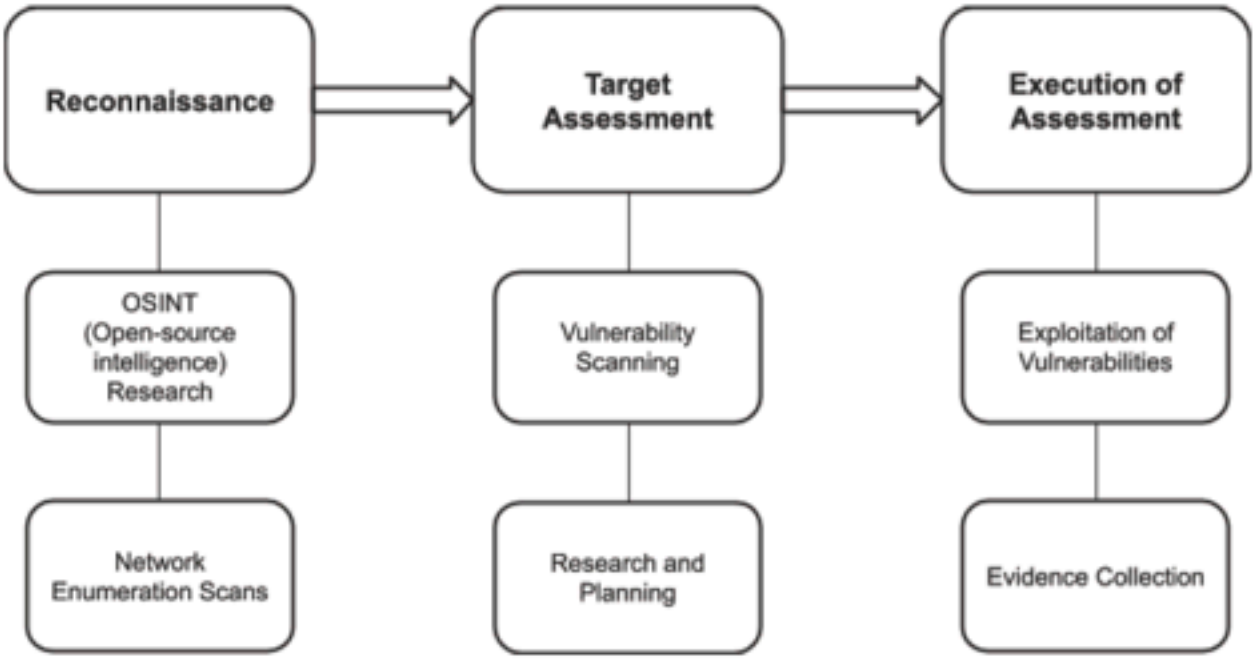
- Many API endpoints within the network utilized no authentication, which left the information that they provided accessible to anyone.
- The PostgreSQL server was accessible using default credentials. It is recommended that these credentials be changed to maintain the integrity of the database.

III. Principle of Least Privilege (PoLP)

- Authentication mechanisms should only allow functionality that is required of the user.
- This methodology can help prevent insider attacks.

TESTING METHODOLOGY

[redacted]s testing methodology consisted of three main phases - reconnaissance, target assessment, and execution of assessment. Reconnaissance involved performing OSINT (Open-source intelligence) research to gather publicly available information and passive and active network enumeration scans to gather information on available hosts and their topology. For target assessment, the consultants began to focus on individual services. Our team used tools such as Nmap, Dirb, and GoBuster to identify service versions and subpages of hosts running websites on the network. Manual vulnerability scans were conducted during this portion of the test. For execution of the assessment, the consultants used tools such as Burp Suite, ZAP, Metasploit, and custom web-shells to find and exploit vulnerabilities. The diagram below shows a visual representation of the testing methodology our team followed throughout our penetration test.



CLASSIFICATION DEFINITIONS

Risk Classifications

LEVEL	SCORE	DESCRIPTION
Critical	9-10	The vulnerability poses an immediate threat to the organization, and exploitation may permanently affect the organization. Remediation should be performed immediately.
High	7-8	The vulnerability poses an urgent threat to the organization. Recovery from impacts of the exploitation may be difficult. Remediation should be prioritized.
Medium	4-6	The exploitation of the vulnerability may result in notable disruption of business functionality. Remediation should be performed when feasible.
Low	1-3	The vulnerability poses a minimal threat to the organization. The presence of the vulnerability should be noted and remediated if possible.
Informational	0	These findings have no clear threat to the organization, but may cause business processes to function undesirably or reveal sensitive information about the company.

Exploitation Likelihood Classifications

LEVEL	DESCRIPTION
Likely	Exploitation methods are well-known and can be performed with minimal difficulty using publicly available tools.
Possible	Exploitation methods are well-known and may be performed using public tools with configuration changes. Understanding of the underlying system is required for successful exploitation.
Unlikely	Exploitation requires deep understanding of the underlying system or advanced technical skills. Precise conditions may be required for successful exploitation.

Business Impact Classifications

LEVEL	DESCRIPTION
Severe	Successful exploitation of the vulnerability may result in wide-spread disruption of critical business functions and significant financial damage.
Moderate	Successful exploitation of the vulnerability may cause significant disruptions to non-critical business functions.
Mild	Successful exploitation of the vulnerability may affect a few users, without causing much disruption to routine business functions.

Remediation Difficulty Classifications

LEVEL	DESCRIPTION
Hard	Remediation may require extensive reconfiguration of the underlying systems and disruption of normal business functions.
Medium	Remediation may require minor reconfigurations or additions that may be time-intensive or expensive.
Easy	Remediation may be accomplished within a short amount of time and with little difficulty.

Initial Penetration Test

This engagement was the second penetration test [redacted] has conducted for Le BonBon Croissant. The consultants re-evaluated all of the vulnerabilities found during the initial penetration test to provide detailed information about Le BonBon Croissant's remediation efforts. Out of the 11 previous findings, 7 findings have yet to be remediated and that status of 4 findings were unable to be assessed. The table below indicates the remediation status of each vulnerability from the initial engagement. The hosts that pertained to vulnerabilities classified with the status "Not Applicable" were unavailable during the retest of the network, so the consultants were unable to confirm whether or not the vulnerabilities have been remediated on the Le BonBon Croissant Network.

Vulnerability Name	Status
FastAPI Unauthenticated Modification of Querying	Not Remediated
Outdated OpenSSH	Not Remediated
Outdated SQL	Not Remediated
PLC Bridge Unauthenticated Access	Not Remediated
PostgreSQL Default Password	Not Remediated
PostgreSQL RCE	Not Remediated
ScadaBR	Not Remediated
GitLab CE User Enumeration	Host Removed
Outdated SMB	Host Removed
RealVNC Outdated Protocol	Host Removed
SMB Message Signing Disabled	Host Removed

Assessment Findings

Summary of Findings

Findings by Risk Level

Critical	High	Medium	Low	Informational
2	3	3	3	3

Table of Findings

NUMBER	FINDINGS	RISK LEVEL	RISK SCORE
1	ScadaBR Authenticated File Upload	Critical	10/10
2	PostgreSQL - Default Credentials	Critical	10/10
3	ScadaBR - Default Credentials	High	8/10
4	Unauthenticated Read/Write Access to API Dashboard and Endpoints	High	7/10
5	Unauthenticated API Access (Read/Write)	High	7/10
6	PostgreSQL - Arbitrary Code Execution Functionality	Medium	6/10
7	Jawbreaker Customer Portal Sequential User IDs	Medium	5/10
8	Base64 Encoded Passwords	Medium	4/10
9	Permit Root Login	Low	2/10
10	User Enumeration	Low	2/10
11	Outdated PostgreSQL	Low	2/10

12	SSH Password Authentication	Informational	0/10
13	Outdated FastAPI	Informational	0/10
14	Directory Listing Enabled	Informational	0/10

Critical Findings

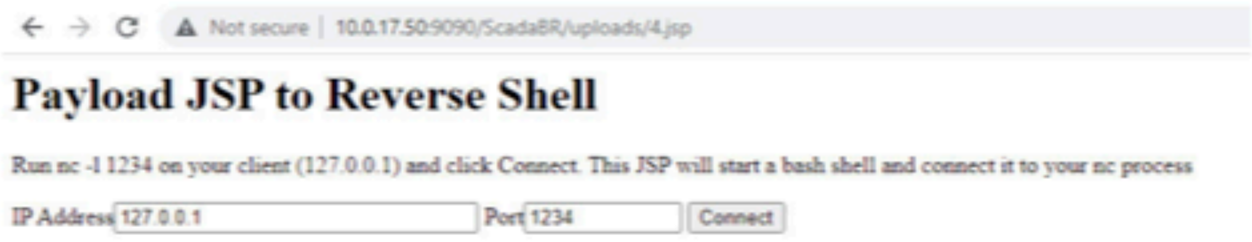
1. ScadaBR Authenticated File Upload (CVE-2021-26828)

Affected Host(s)	Host Name	Location	Service
10.0.17.50	crunch-serial	Port 9090	ScadaBR

Risk Score:	10/10
Exploitation Likelihood:	Possible
Business Impact:	Severe
Remediation Difficulty:	Medium

Description:
ScadaBR located at 'http://10.0.17.50:9090/ScadaBR/' is a web-management software that is used to manage Le BonBon Croissant's Programmable Logic Controllers or PLC. These devices measure and control physical machinery in the warehouse. The current version of ScadaBR allows for file upload and remote code execution.

Steps to reproduce:
Using the LinScada_RCE tool, a malicious user can upload a JSP file to the browsable 'uploads' directory on the web server at 'http://10.0.17.50:9090/ScadaBR/uploads/'.



Payload JSP to Reverse Shell

Run nc -l 1234 on your client (127.0.0.1) and click Connect. This JSP will start a bash shell and connect it to your nc process

IP Address 127.0.0.1 Port 1234

Remediation:
Update to the latest version of ScadaBR. Additionally, it is recommended to remove execution permissions from the 'uploads' directory.

Reference(s):

- Exploit-DB - ScadaBR Arbitrary File Upload
  https://www.exploit-db.com/exploits/49735
- NVD Database - CVE-2021-26828
  https://nvd.nist.gov/vuln/detail/CVE-2021-26828

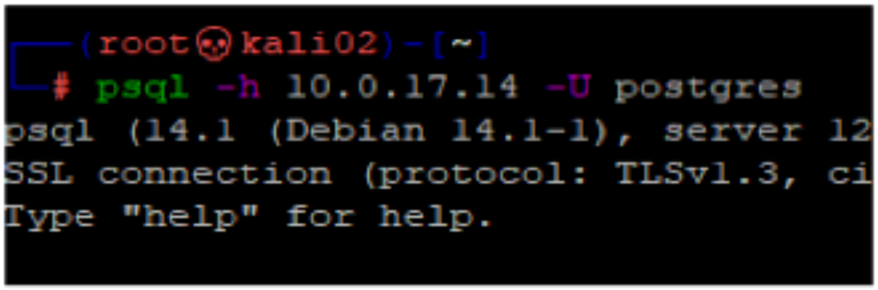
2. PostgreSQL - Default Credentials

Affected Host(s)	Host Name	Location	Service
10.0.17.14	charley	Port 5432	PostgreSQL

Risk Score:	10/10
Exploitation Likelihood:	Likely
Business Impact:	Severe
Remediation Difficulty:	Easy

Description:
The PostgreSQL database is currently using a default configuration which allows for remote access to the administrative user of the database with no password. A malicious user could potentially change billing and credit card data and view confidential client information.

Steps to reproduce:
Connect to the PostgreSQL database on 10.0.17.14 using:
'psql -h 10.0.17.14 -U postgres'.



```
┌──(root💀kali02)-[~]
└─# psql -h 10.0.17.14 -U postgres
psql (14.1 (Debian 14.1-1), server 12
SSL connection (protocol: TLSv1.3, ci
Type "help" for help.
```

Remediation:
To remediate this vulnerability, set a password for the user 'postgres' and if possible use a non-administrative and not default user.

Reference(s):

- PostgreSQL Documentation - Password Authentication
  https://www.postgresql.org/docs/13/auth-password.html

High Findings

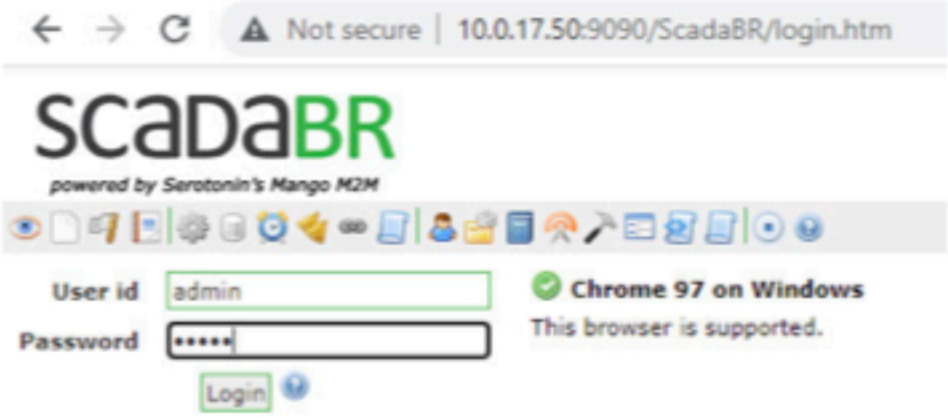
3. ScadaBR - Default Credentials

Affected Host(s)	Host Name	Location	Service
10.0.17.51	charley	Port 9090	ScadaBR

Risk Score:	8/10
Exploitation Likelihood:	Likely
Business Impact:	Severe
Remediation Difficulty:	Easy

Description:
A malicious user could potentially guess the default password pair of (admin:admin) and login to critical infrastructure.

Steps to reproduce:
Browse to http://10.0.17.50:9090/ScadaBR/login.htm and enter the username 'admin' and the password 'admin'.


Not secure | 10.0.17.50:9090/ScadaBR/login.htm
scadaBR
powered by Serotonin's Mango M2M
Login

User id: admin

Password: •••••

Chrome 97 on Windows

This browser is supported.

Remediation:
Using the user panel on the administrative page, update the admin password to a more secure credential pair.

Reference(s):

- CWE-798 Use of Hard-coded Credentials https://cwe.mitre.org/data/definitions/798.html

4. Unauthenticated Read/Write Access to API Dashboard and Endpoints

Affected Host(s)	Host Name	Location	Service
10.0.17.11	goldenticket	443	FastAPI

Risk Score:	7/10
Exploitation Likelihood:	Possible
Business Impact:	Moderate
Remediation Difficulty:	Medium

Description:
Users have unauthenticated access to multiple API endpoints through the FastAPI dashboard on https://10.0.17.11/docs/. These endpoints provide further information about accounts to these users. Individual user account information can be viewed on https://10.0.17.11/account/ and accounts can be verified on 10.0.17.11/check/ by providing a user ID. Accounts are able to be created with arbitrary amounts of money on the https://10.0.17.11/add/ endpoint. A listing of all accounts can be found on the https://10.0.17.11/accounts/ page.

Steps to reproduce:
These endpoints can be accessed through the FastAPI dashboard on 10.0.17.11/docs/. The account information can be viewed and verified with a valid user ID on https://10.0.17.11/account/ and https://10.0.17.11/check/ respectively. The accounts can be created on the /add/ section of the https://10.0.17.11/docs/ page by using the drop-down without authentication. All accounts can be viewed by navigating to the https://10.0.17.11/accounts/ page.

Remediation:
Remove public access to the API and require authentication to make any changes or view accounts.

Reference(s):

- CWE-306 Missing Authentication for Critical Function - https://cwe.mitre.org/data/definitions/306.html

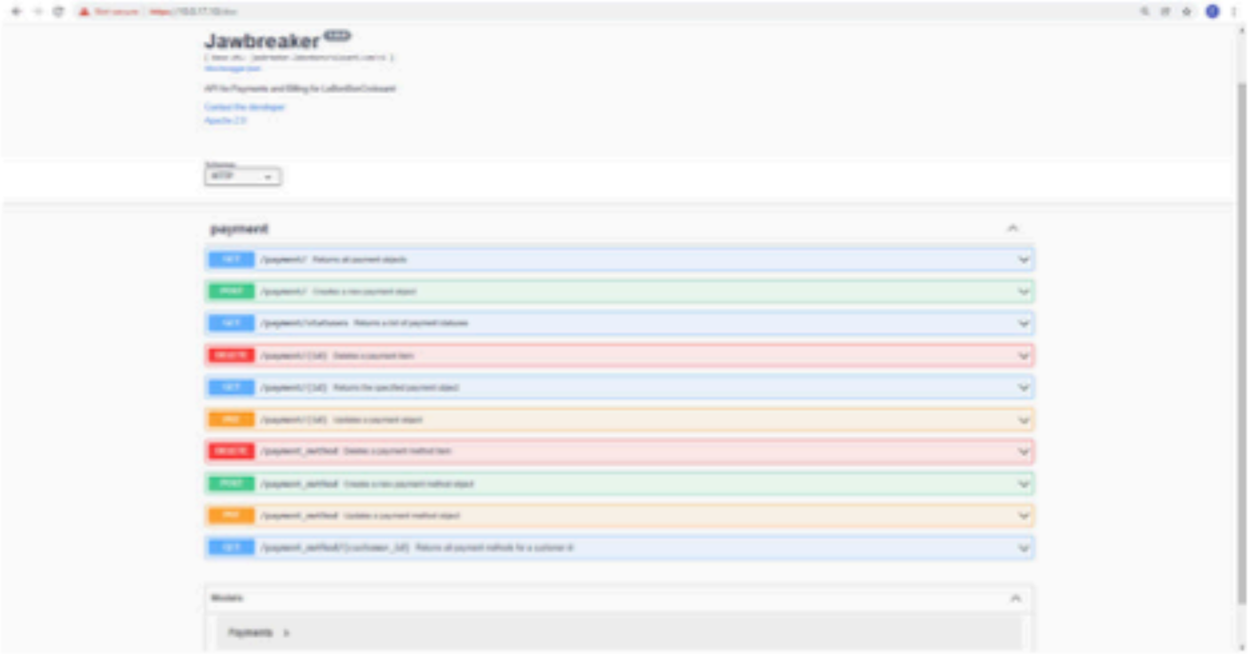
- FastAPI Security Documentation - https://fastapi.tiangolo.com/tutorial/security/first-steps/

5. Unauthenticated API Access (Read/Write)

Affected Host(s)	Host Name	Location	Service
10.0.17.10	eggdicator	Port 443	Jawbreaker API

Risk Score:	7/10
Exploitation Likelihood:	Possible
Business Impact:	Moderate
Remediation Difficulty:	Medium

Description:
A malicious actor can use the Jawbreaker custom portal to request or modify payment information from the Jawbreaker API.



Steps to reproduce:
Navigate to the web server (https://10.0.17.10/docs). Once there, query results using ID numbers in the range of existing IDs to see results.

Remediation:

The consultants suggest implementing a form of authentication to connect to and use these API functions. Using an API key, tokens, or cookies, a user or service should be allowed to query information from the API.

Reference(s):

- Stoplight.io - Best Practices to Authenticate API's
  https://blog.stoplight.io/api-keys-best-practices-to-authenticate-apis

Medium Findings

6. PostgreSQL - Code Execution Functionality

Affected Host(s)	Host Name	Location	Service
10.0.17.14	charley	Port 5432	PostgreSQL

Risk Score:	6/10
Exploitation Likelihood:	Possible
Business Impact:	Moderate
Remediation Difficulty:	Medium

Description:
The default configuration for PostgreSQL database enables arbitrary command execution on the host machine, allowing an attacker to execute commands on the database's host machine as the 'postgres' user or any user with permissions to use the PROGRAM command. Effectively, a user who either has access to the database with proper credentials or is able to successfully exploit an SQL injection attack may perform arbitrary code execution on the host machine.

Steps to reproduce:
Connect to the PostgreSQL database on 10.0.17.14 using 'psql -h 10.0.17.14 -U postgres'. Create a table called 'cmd_exec', then execute 'COPY *cmd_exec* FROM PROGRAM '*input_commands_here'* to capture the output of a specified input of *input_commands_here* into the table *cmd_exec.* By executing the 'SELECT * FROM cmd_exec' statement, a screen will appear presenting the contents of the *cmd_exec* table.

Remediation:
Configure the permissions for users in your PostgreSQL database to revoke permissions for the PROGRAM command.
For example, to revoke permissions to use the PROGRAM command in your existing databases and revoke permissions to use the CREATE command to prevent the creation of new databases:
` REVOKE PROGRAM ON example_database FROM example_user`

Reference(s):

- PostgreSQL Documentation - Command Execution
  https://www.postgresql.org/docs/9.1/libpq-exec.html
- Greenwolf Security - Arbitrary Command Execution in PostgreSQL
  https://medium.com/greenwolf-security/authenticated-arbitrary-command-execution-on-postgresql-9-3-latest-cd18945914d5
- DigitalOcean - How to Modify User Privileges in PostgreSQL
  https://docs.digitalocean.com/products/databases/postgresql/how-to/modify-user-privileges/

7. Jawbreaker Customer Portal Sequential User IDs

Affected Host(s)	Host Name	Location	Service
10.0.17.10	eggdicator	Port 443	FastAPI

Risk Score:	5/10
Exploitation Likelihood:	Possible
Business Impact:	Mild
Remediation Difficulty:	Easy

Description:
Customer IDs are generated sequentially, allowing a malicious actor to predict and enumerate user data from the data store.

Steps to reproduce:
Navigate to the web server (http://10.0.17.10:80). Once there, query results, within the range of current IDs, using ID numbers in sequential order to view results.

Remediation:
Generate user IDs non sequentially or by using a unique identifier.

Reference(s):

- Internet Engineering Task Force - RFC 4122
  https://datatracker.ietf.org/doc/html/rfc4122

8. Base64 Encoded Passwords

Affected Host(s)	Host Name	Location	Service
10.0.17.14	charley	wmci - logins	MySQL

Risk Score:	4/10
Exploitation Likelihood:	Possible
Business Impact:	Mild
Remediation Difficulty:	Medium

Description:
User passwords were stored using base64 encoding. Base64 is a reversible encoding algorithm, and in the event of a breach would allow an attacker to retrieve user passwords.

Steps to reproduce:
Using a Base64 encoding tool, decode the passwords to verify.

Remediation: Use strong hashing algorithms such as SHA-256 on stored user passwords, including salts.

Reference(s):

- OWASP - Password Storage Cheat Sheet
  https://cheatsheetseries.owasp.org/cheatsheets/Password_Storage_Cheat_Sheet.html

Low Findings

9. Permit Root Login

Affected Host(s)	Host Name	Location	Service
10.0.17.14	charley	Port 22	OpenSSH

Risk Score:	2/10
Exploitation Likelihood:	Unlikely
Business Impact:	Mild
Remediation Difficulty:	Medium

Description:
On the affected host, sshd config 'PermitRootLogin' is enabled. This would allow a malicious actor could attempt to brute force the password for root to SSH into the system.

Steps to reproduce:
Attempt to SSH into the host as root and guess at passwords.

Remediation:
Disable PermitRootLogin, see IBM article for information on how to remediate.

Reference(s):

- Baeldung - Why Should We Disable Root-Login Over SSH
  https://www.baeldung.com/linux/root-login-over-ssh-disable
- IBM - Enable or Disable Root Login for SSH
  https://www.ibm.com/docs/en/db2/11.5?topic=installation-enable-disable-remote-root-login

10. User Enumeration

Affected Host(s)	Host Name	Location	Service
10.0.17.11	goldenticket	Port 22	OpenSSH
10.0.17.12	scrumdiddlyumptious	Port 22	OpenSSH
10.0.17.13	whatchamacallit	Port 22	OpenSSH
10.0.17.15	bucket	Port 22	OpenSSH

Risk Score:	2/10
Exploitation Likelihood:	Possible
Business Impact:	Mild
Remediation Difficulty:	Medium

Description:
With the current version of OpenSSH on the affected hosts, it is possible for an attacker to guess possible usernames for the hosts and have them confirmed by a python script. With a list of confirmed usernames, an attacker could use them in a brute force attack on the possible passwords for each user that they confirmed.

Steps to reproduce:
Go to the exploit-db link referenced below. Download the python code on the webpage. The script requires modification to work correctly. It was made for python2 and needs to be updated to python3. You just need to add parentheses around the string part of the print statements in the file.(Ex: print “hello world” changes to print(“hello world”). After doing this, you can now run the code [python3 UserEnum.py target_ip username]. You can do this for as many times as you wish to test to see results for both existing and nonexistent users.

Remediation:
Updating the OpenSSH will fix this issue, as it is not present in the newest version.

Reference(s):

- NVD Database - CVE-2018-15473 https://nvd.nist.gov/vuln/detail/CVE-2018-15473,
- Exploit-DB - CVE-2018-15473 Exploit https://www.exploit-db.com/exploits/45939,

11. Outdated PostgreSQL

Affected Host(s)	Host Name	Location	Service
10.0.17.14	charley	Port 5432	Postgresql

Risk Score:	2/10
Exploitation Likelihood:	Unlikely
Business Impact:	Mild
Remediation Difficulty:	Medium

Description:
The affected host was running an outdated version of the PostgreSQL service. Upgrading to the latest version will ensure stability and support.

Steps to reproduce:
Check the version by running ‘psql –version’.

Remediation:
Update PostgreSQL on 10.0.17.14. You can find the current version of the software at https://www.postgresql.org/download/.

Reference(s):

- PostgreSQL Downloads - https://www.postgresql.org/download/

Informational Findings

12. SSH Password Authentication

Affected Host(s)	Host Name	Location	Service
10.0.17.10-16	charley	Port 22	OpenSSH

Risk Score:	0/10
Exploitation Likelihood:	Possible
Business Impact:	Mild
Remediation Difficulty:	Medium

Description:
SSH password authentication was enabled on the affected machine. This is bad because the user is required the password, which could be weak, reused, or even a password that has been in a data breach. The systems should use SSH keys for authentication. This is a more secure alternative to passwords and will protect against ssh password attacks.

Steps to reproduce:
SSH into the user and provide a password. 'ssh <user>@10.0.17.14'

Remediation:
Disable SSH password authentication and enable SSH key authentication.

Reference(s):

- Thorn Tech - Passwords VS SSH Keys
  https://www.thorntech.com/passwords-vs-ssh/

13. Outdated FastAPI

Affected Host(s)	Host Name	Location	Service
10.0.17.11	goldenticket	Port 443	FastAPI

Risk Score:	0/10
Exploitation Likelihood:	Unlikely
Business Impact:	Mild
Remediation Difficulty:	Medium

Description:
FastAPI is a web framework created to build APIs with Python 3.6+. Versions below 0.65.2 that use cookies for authentication in path operations that received JSON payloads sent by browsers were vulnerable to a Cross-Site Request Forgery (CSRF) attack. A request with header text/plain containing JSON data would be accepted and extracted. Header text/plain is exempt from CORS preflights, as they were considered simple requests. The browser will execute these forms immediately, and include the observing browser's cookies. The text content could be a JSON string that would be parsed and accepted by the FastAPI application.

Steps to reproduce:
This vulnerability could not be reproduced in the client's system, as it is considered a social engineering exploit. Meaning, there is a user action that must occur from this type of vulnerability, such as navigating to a link.

Remediation:
Upgrading the FastAPI to the latest version resolves the issue by only accepting JSON data if the header is application/json or other applicable JSON header.

Reference(s):

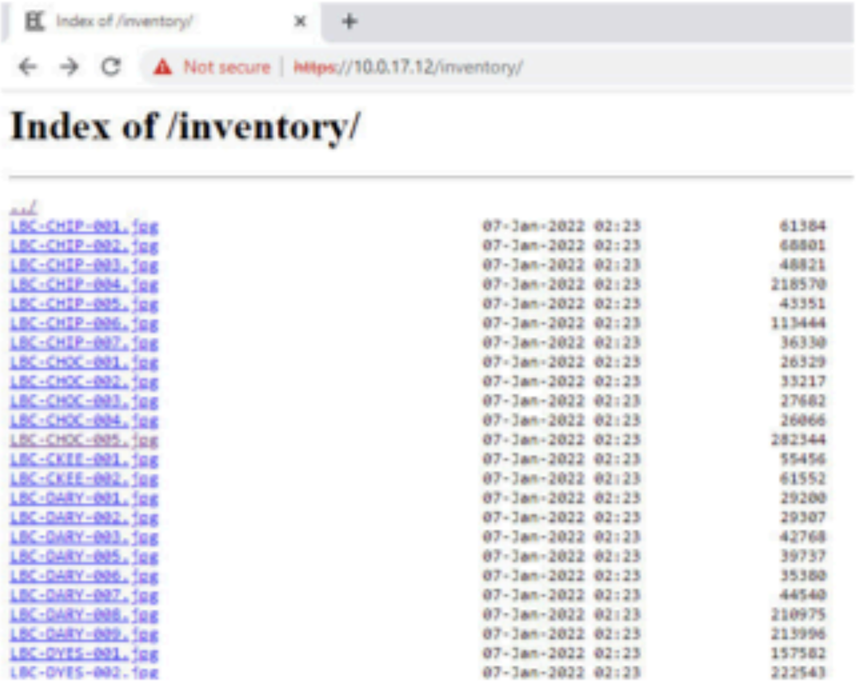
- https://nvd.nist.gov/vuln/detail/CVE-2021-32677
- https://github.com/tiangolo/fastapi/security/advisories/GHSA-8h2j-cgx8-6xv7

14. Directory Listing Enabled

Affected Host(s)	Host Name	Location	Service
10.0.17.12	scrumdiddlyumptious	Port 443	Nginx web server

Risk Score:	0/10
Exploitation Likelihood:	Likely
Business Impact:	Mild
Remediation Difficulty:	Medium

Description:
On the host 10.0.17.12, the inventory image directory's index is publicly viewable at https://10.0.17.12/inventory/ . The viewing of a directory's index is typically not intended when the contents of the directory are meant to be referenced by a webpage and not meant to be directly accessed by the end user.
Commonly found with the ability to view a directory's index is the capability for Path Traversal.



Index of /inventory/

```
../
LBC-CHIP-001.jpg                                   07-Jan-2022 02:23               61384
LBC-CHIP-002.jpg                                   07-Jan-2022 02:23               68801
LBC-CHIP-003.jpg                                   07-Jan-2022 02:23               48821
LBC-CHIP-004.jpg                                   07-Jan-2022 02:23              218570
LBC-CHIP-005.jpg                                   07-Jan-2022 02:23               43351
LBC-CHIP-006.jpg                                   07-Jan-2022 02:23              113444
LBC-CHIP-007.jpg                                   07-Jan-2022 02:23               36330
LBC-CHOC-001.jpg                                   07-Jan-2022 02:23               26329
LBC-CHOC-002.jpg                                   07-Jan-2022 02:23               33217
LBC-CHOC-003.jpg                                   07-Jan-2022 02:23               27682
LBC-CHOC-004.jpg                                   07-Jan-2022 02:23               26066
LBC-CHOC-005.jpg                                   07-Jan-2022 02:23              282344
LBC-CKEE-001.jpg                                   07-Jan-2022 02:23               55456
LBC-CKEE-002.jpg                                   07-Jan-2022 02:23               61552
LBC-DARY-001.jpg                                   07-Jan-2022 02:23               29200
LBC-DARY-002.jpg                                   07-Jan-2022 02:23               29307
LBC-DARY-003.jpg                                   07-Jan-2022 02:23               42768
LBC-DARY-005.jpg                                   07-Jan-2022 02:23               39737
LBC-DARY-006.jpg                                   07-Jan-2022 02:23               35380
LBC-DARY-007.jpg                                   07-Jan-2022 02:23               44540
LBC-DARY-008.jpg                                   07-Jan-2022 02:23              210975
LBC-DARY-009.jpg                                   07-Jan-2022 02:23              213996
LBC-DYES-001.jpg                                   07-Jan-2022 02:23              157582
LBC-DYES-002.jpg                                   07-Jan-2022 02:23              222543
```

Steps to reproduce:
The index can be accessed by refreshing the inventory page on 10.0.17.12. It can also be accessed by directly navigating to https://10.0.17.12/inventory/.

Remediation:
By modifying the nginx configuration, it is possible to disable directory listing for the website. The target parameter change is ‘autoindex on’ to ‘autoindex off’

Reference(s):

- Nginx Tutorial - Disable Directory Listing
  https://techexpert.tips/nginx/nginx-disable-directory-listing/

APPENDIX A: TOOLS USED

Tools Used	
Tool Name	**Tool Description**
Burp Suite	A Java based Web Penetration Testing framework that is an integrated platform for performing security testing of web applications.
Dirb	A Web Content Scanner that locates existing web objects by launching a dictionary based attack against a web server and analyzing the responses.
GoBuster	A tool used to brute-force URLs (directories and files) in websites and DNS subdomains.
Hashcat	A password cracking tool used for password recovery.
Hydra	A parallelized login cracker which supports numerous protocols to attack. It is used to brute-force username and password to different services such as ftp, ssh, telnet, MS-SQL, etc.
Nmap	A network scanning tool that uses IP packets to identify all the devices connected to a network and to provide information on the services and operating systems they are running.
Metasploit	A penetration testing framework that allows users to conveniently configure and enact exploitation of remote hosts.
http.server Python Module	A python module that starts an HTTP webserver and can be used to host malicious files to be downloaded to a victim server.
OWASP ZAP	A dynamic application security testing (DAST) tool for finding vulnerabilities in web applications.