EPITA - School of Engineering and Computer Science

Master of Computer Science in Computer Security 2022/24

Software and Database Security

Security Report for StockApp Desktop Application

6th June, 2023

Team Members

Saajan Maharjan

Santosh Kumar Yadav

Sahithi Rani Nacharaju

Nikhil Kumar Bachawal

Kanimozhi Iyyanar

Table of Contents

Title: Password Hardcoded in Executable File
The Common Weakness Enumeration (CWE) ID: https://cwe.mitre.org/data/definitions/798.html

CVSS v3.1 Vector AV:L/AC:H/PR:N/UI:N/S:U/C:H/I:L/A:N

Base Score: 5.7

Risk Level: Medium

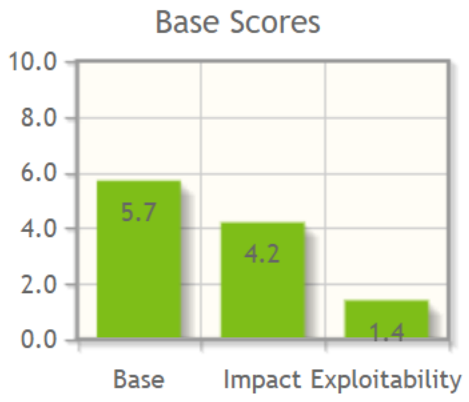
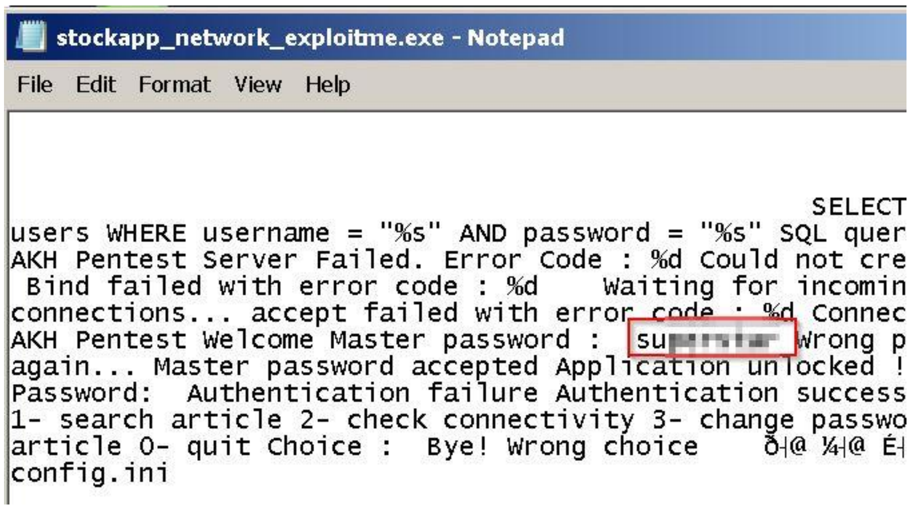


Fig. base score 5.7

Description: Hard-coding credentials in the source code or configuration files of an application or system poses a significant security risk. This practice makes sensitive information easily accessible to attackers, increasing the likelihood of theft or exploitation. Attackers can retrieve the credentials by gaining unauthorized access to the source code or configuration files, potentially leading to unauthorized access to other systems or sensitive data. Hard-coded credentials lack the necessary security controls and encryption, making them vulnerable to exploitation.

Exploitation:

- Locate "thick_client_local.exe" in the application folder
- Open the file with a text editor.
- Search for the keyword "password" in the file
- Run the "thick_client_local.exe" application and enter the password found in step 3 to unlock the application.


stockapp_network_exploitme.exe - Notepad
File Edit Format View Help

```
                                                   SELECT
users WHERE username = "%s" AND password = "%s" SQL quer
AKH Pentest Server Failed. Error Code : %d Could not cre
 Bind failed with error code : %d    Waiting for incomin
connections... accept failed with error code : %d Connec
AKH Pentest Welcome Master password :  su[redacted] Wrong p
again... Master password accepted Application unlocked !
Password:  Authentication failure Authentication success
1- search article 2- check connectivity 3- change passwo
article 0- quit Choice :  Bye! Wrong choice    ð┤@ ¼┤@ É┤
config.ini
```

Fig. password appear in notepad text editor

- The password stored in the executable can be used to access the application.

Remediation: To mitigate this risk, it is important to follow security best practices, such as utilizing secure storage mechanisms instead of hard-coding credentials. By adopting these practices, organizations can enhance the security of their applications and systems, protecting sensitive information and reducing the potential for unauthorized access.

Title: Password Shown While Typed
The Common Weakness Enumeration (CWE) ID: https://cwe.mitre.org/data/definitions/549.html

CVSS v3.1 Vector AV:L/AC:H/PR:H/UI:N/S:U/C:H/I:H/A:H

Base Score: 5.7

Risk Level: Medium

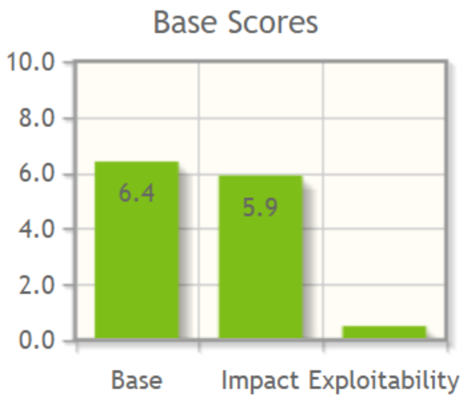
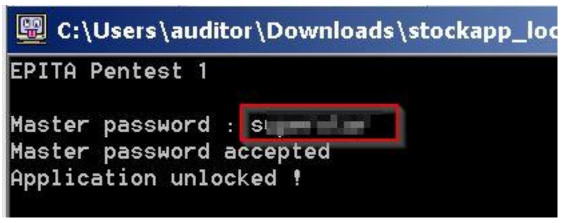


Fig: Base score 5.7

Description: The lack of password masking during input increases the risk of malicious attackers being able to monitor and record passwords. This vulnerability leaves user passwords exposed and susceptible to unauthorized access.

Exploitation:

- Open the Stock App Desktop Application
- Enter The Password
- Password can be viewed while typing



```
C:\Users\auditor\Downloads\stockapp_loc
EPITA Pentest 1

Master password : [redacted]
Master password accepted
Application unlocked !
```

Fig. Password showing while typing

Remediation:

- Enable password Masking by replacing the password characters with special characters such as asterisks
- Give an option for the user to temporarily unmask the password. So, users can reconfirm the entered password.

Title: Password exposed in Plain Text in config file
The Common Weakness Enumeration (CWE) ID: https://cwe.mitre.org/data/definitions/260.html

CVSS v3.1 Vector AV:L/AC:H/PR:N/UI:R/S:U/C:H/I:H/A:H

Base Score: 7.0

Risk Level: High

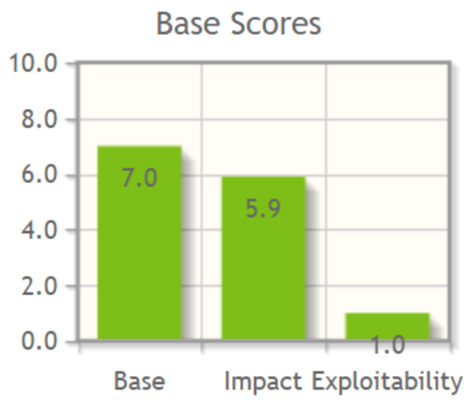
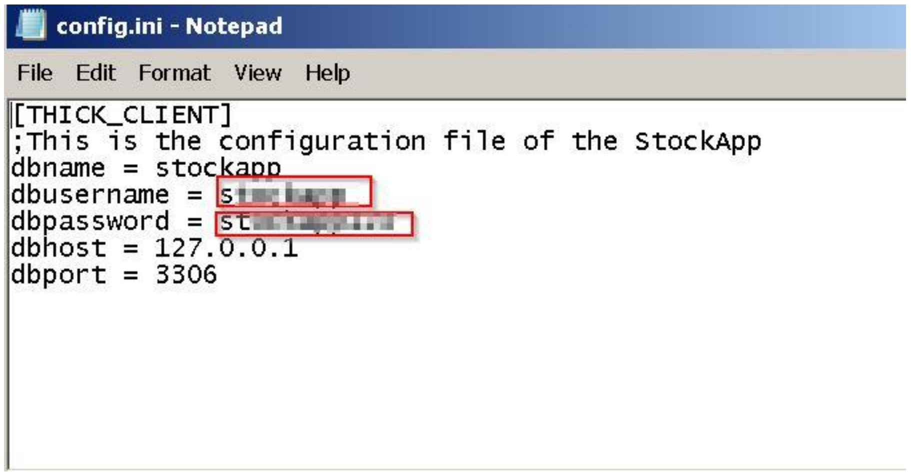


Fig. Base score 7.0

Description: Storing passwords in plain text within configuration files introduces a significant security vulnerability. This practice compromises password protection as anyone with access to the file can easily read and retrieve the passwords. Consequently, unauthorized individuals can gain access to the application and its associated data, posing a serious risk to the security and confidentiality of sensitive information.

Exploitation:

- Open the stockApp_ network_ exploitme
- Open the file <config.ini>



```
[THICK_CLIENT]
;This is the configuration file of the StockApp
dbname = stockapp
dbusername = [redacted]
dbpassword = [redacted]
dbhost = 127.0.0.1
dbport = 3306
```

Fig: Database username and password clearly shown in config.ini

- The file opens in the notepad which shows sensitive information like dB password

Remediation:

- To ensure the security of sensitive data, it is recommended to utilize secure storage solutions like Key Management Systems (KMS) or vaults for storing information such as passwords. These solutions offer robust protection by providing secure storage, encryption, and access control mechanisms. By leveraging KMS or vaults, organizations can enhance the confidentiality and integrity of sensitive data, reducing the risk of unauthorized access or exposure.
- When passwords need to be stored in configuration files, it is crucial to encrypt them using a strong encryption algorithm. By encrypting passwords, organizations can protect against unauthorized access, even if the configuration file is compromised. Encryption adds an additional layer of security, ensuring that the passwords remain unreadable and unusable to unauthorized individuals. This practice mitigates the risk of password exposure and helps maintain the confidentiality and integrity of sensitive information stored in configuration files.

Title: Password Aging Not Implemented
The Common Weakness Enumeration (CWE) ID [CWE - CWE-262: Not Using Password Aging (4.11) (mitre.org)]

CVSS v3.1 Vector AV:L/AC:H/PR:N/UI:N/S:U/C:H/I:H/A:L
Base Score: 6.9

Risk Level: Medium

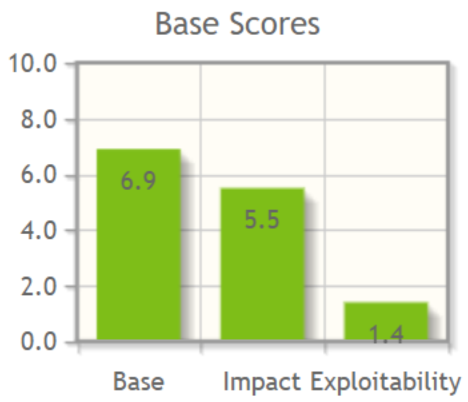


Fig. Base scores 6.9

Description: The absence of a mechanism to enforce regular password changes creates a vulnerability that gives attackers extended time to engage in password cracking before users are required to switch to a new password. This delay increases the likelihood of successful password attacks and compromises the security of user accounts and sensitive data.

Exploitation:

- If users are not required to change their passwords on a regular basis, attackers have more time to guess or crack passwords.
- Thus, If a user's password is compromised, an attacker may have access to the account
- for an extended period of time.
- Access control can become insufficient. For example, a user may leave an organization
- but still have access to resources because their password has not been changed.

Remediation:

- Access control can become insufficient. For example, a user may leave an organization but still have access to resources because their password has not been changed.
- Ensure that the user is notified several times leading up to the password expiration.
- Create mechanisms to prevent users from reusing passwords or creating similar passwords

Title: Single factor Authentication

Common Weakness Enumeration (CWE) ID: https://cwe.mitre.org/data/definitions/308.html

CVSS v3.1 Vector AV:L/AC:L/PR:H/UI:R/S:C/C:H/I:H/A:L
Base Score: 7.7

Risk Level: Medium

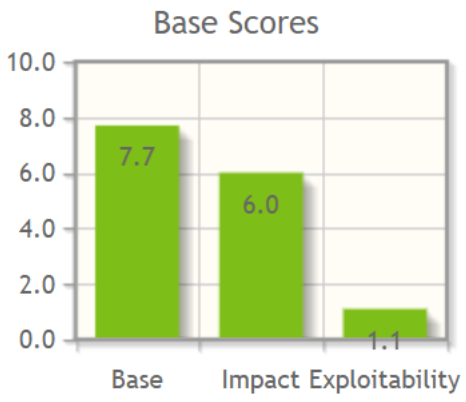
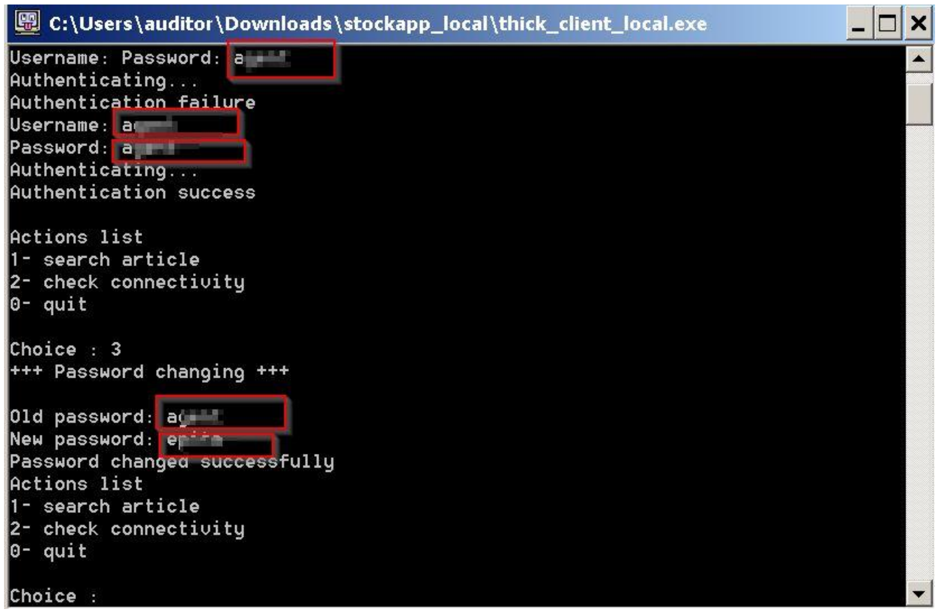


Fig. Base scores 7.7

Description: Utilizing single-factor authentication in an application introduces a vulnerability that unnecessarily heightens the risk of compromise. This increases the likelihood of unauthorized access by enabling attackers to employ various techniques such as brute force attacks, phishing, credential stuffing, password spraying, or keylogging. By leveraging these methods, attackers can exploit the absence of additional authentication factors, compromising the security of the application and potentially gaining unauthorized access to sensitive information.

Exploitation:

- Run the “thick_client_local.exe” application and enter the “Master” password to unlock the application.
- Login to the application with a valid username and password.
- The user needs only username and password authentication to log in to the application.



```
C:\Users\auditor\Downloads\stockapp_local\thick_client_local.exe
Username: Password: a[redacted]
Authenticating...
Authentication failure
Username: a[redacted]
Password: a[redacted]
Authenticating...
Authentication success

Actions list
1- search article
2- check connectivity
0- quit

Choice : 3
+++ Password changing +++

Old password: ag[redacted]
New password: ep[redacted]
Password changed successfully
Actions list
1- search article
2- check connectivity
0- quit

Choice :
```

Fig. use single factor authentication

Remediation:

It is important to implement robust password policies and requirements. This involves enforcing longer passwords with complexity criteria, regularly expiring passwords, implementing lockout policies, and leveraging two-factor authentication whenever feasible. As a MFA(Multi Factor Authentication) By enforcing these measures, organizations can enhance the strength of user passwords, reduce the risk of unauthorized access, and fortify overall system security.

Strong password policies and requirements act as a crucial defense against common attack methods, significantly mitigating the chances of successful breaches and unauthorized account access. Use long password using uppercase, lowercase, number special characters.

Using Google Authenticator, Microsoft Authenticator, Yubikey and secured mobile apps.

Implement secure password storage mechanisms, such as strong hashing algorithms and salting (e.g., bcrypt, SHA-256) combined with unique per-user salts. Avoid storing passwords in plaintext or using weak encryption.

Title: Use of Client Side Authentication
The Common Weakness Enumeration (CWE) ID: https://cwe.mitre.org/data/definitions/603.html

CVSS v3.1 Vector AV:L/AC:L/PR:L/UI:N/S:C/C:L/I:L/A:L

Base Score: 6.3

Risk Level: Medium

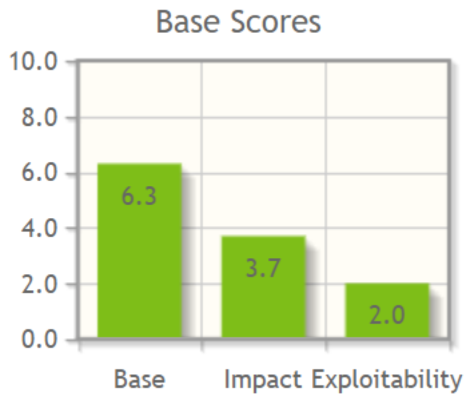


Fig. Base scores 6.3

Description: Using client-side authentication, where user authentication is performed on the client-side (e.g., in a web browser or mobile app) instead of the server-side, introduces a significant vulnerability. This approach is insecure because the client-side code can be easily manipulated by attackers, enabling them to bypass the authentication mechanism and gain unauthorized access to protected resources. By modifying the client-side code, attackers can circumvent the authentication process and potentially exploit sensitive data or functionalities. To ensure robust security, it is crucial to implement server-side authentication, where the authentication process occurs on the server, providing a more secure and reliable method for verifying user credentials and protecting sensitive resources.

Exploitation:

Remediation:

- To enhance security, it is essential to perform authentication and authorization checks on the server-side. This ensures that all sensitive operations are properly authenticated and authorized, mitigating the risk of unauthorized access. Utilizing industry-standard authentication protocols like OAuth or OpenID Connect, along with secure session management techniques such as SSL/TLS encryption and secure cookies, further strengthens the security of the system. By implementing these measures, organizations can safeguard sensitive data, protect against unauthorized access, and maintain a secure environment for users.

Title: Database user Over Privileged

The Common Weakness Enumeration (CWE) ID: https://cwe.mitre.org/data/definitions/269.html

CVSS 3.1 Base Score Metrics ():

CVSS v3.1 Vector CVSS v3.1 AV:N/AC:H/PR:N/UI:N/S:U/C:L/I:L/A:N •

Base Score: 7.2

Risk Level: High

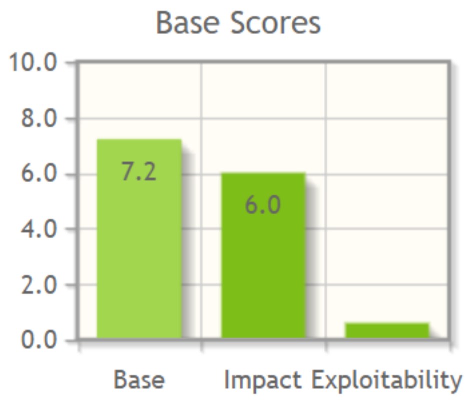
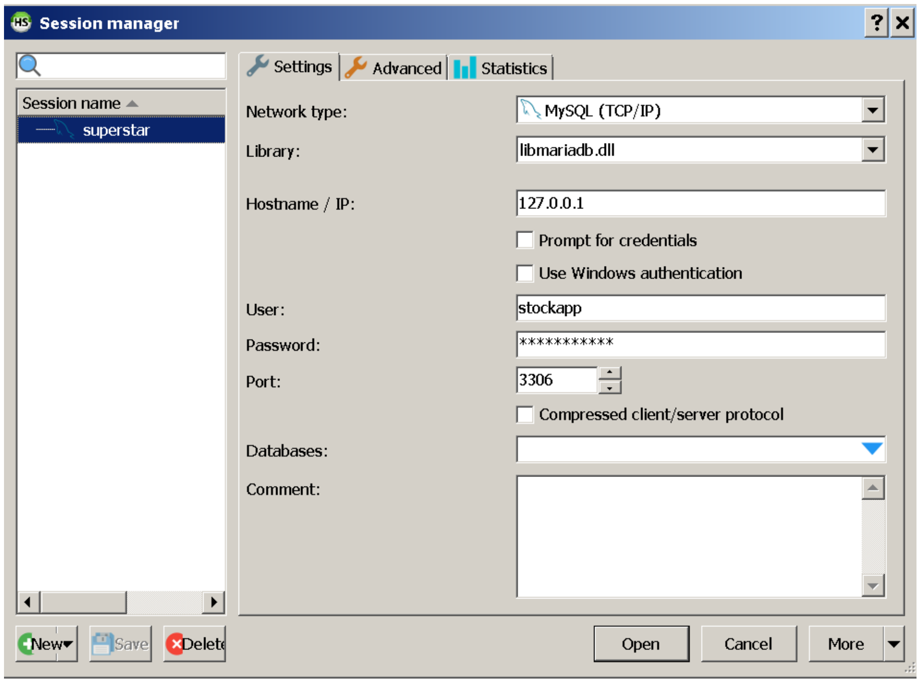


Fig. Base score 7.2

Description: Vulnerability that occurs which can access all the table from the superstar database list the data like commune which we can modify delete and add the data. The attacker may use the overprivileged user account to access sensitive data, such as customer information, financial records, or intellectual property. This can lead to data breaches and privacy violations. attacker may exploit the overprivileged user account to escalate their privileges further within the database system or gain administrative access to the underlying infrastructure. With elevated privileges, the attacker can modify, delete, or manipulate data within the database, causing data integrity issues, financial loss, or disruption of business operations.

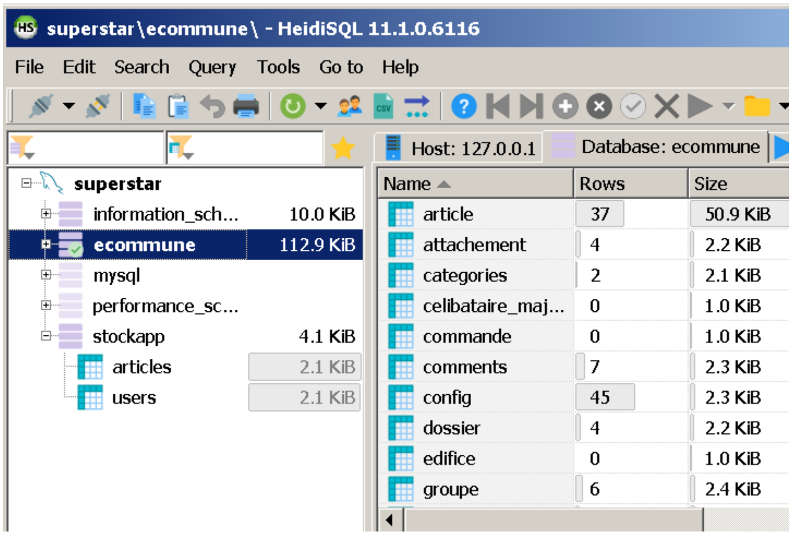
Exploitation:

- By using the DB management tool like HieldiSQL press the connect option
- Enter the DB username and password and that connects to the StockApp database tables.
- Where all the usernames and passwords are visible.



Network type: MySQL (TCP/IP)
Library: libmariadb.dll
Hostname / IP: 127.0.0.1
Prompt for credentials
Use Windows authentication
User: stockapp
Password: ************
Port: 3306
Compressed client/server protocol
Databases:
Comment:

Fig. Database connection using HieldiSQL of stockapp



superstar	
information_sch...	10.0 KiB
ecommune	112.9 KiB
mysql	
performance_sc...	
stockapp	4.1 KiB
articles	2.1 KiB
users	2.1 KiB

Name	Rows	Size
article	37	50.9 KiB
attachement	4	2.2 KiB
categories	2	2.1 KiB
celibataire_maj...	0	1.0 KiB
commande	0	1.0 KiB
comments	7	2.3 KiB
config	45	2.3 KiB
dossier	4	2.2 KiB
edifice	0	1.0 KiB
groupe	6	2.4 KiB

Fig. Database table of ecommune

Remediation:

- Implement secure authentication mechanisms, such as multi-factor authentication (MFA), to strengthen user authentication and prevent unauthorized access. Implement robust authorization controls to ensure that users can only access the data and perform actions appropriate for their roles.

- Conduct a thorough review of all user accounts in the database system and identify those with excessive privileges. Determine the minimum privileges required for each user based on their roles and responsibilities.
- Implement Role-based access control mechanisms to assign permissions and privileges based on predefined roles within the organization. This ensures that users are only granted access to the resources they need to fulfill their responsibilities.

Title: Plaintext Storage of a Password

The Common Weakness Enumeration (CWE) ID: https://cwe.mitre.org/data/definitions/256.html

CVSS 3.1 Base Score Metrics ():

CVSS v3.1 Vector CVSS v3.1 AV:N/AC:H/PR:N/UI:N/S:U/C:L/I:L/A:N •

Base Score: 7.2

Risk Level: High

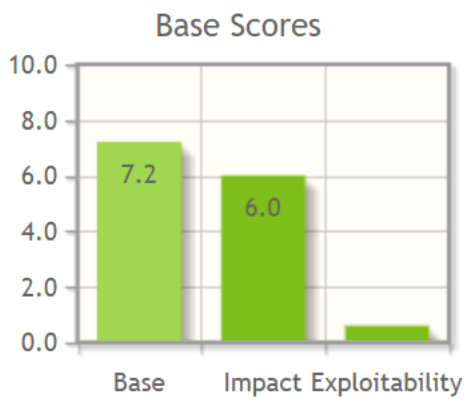
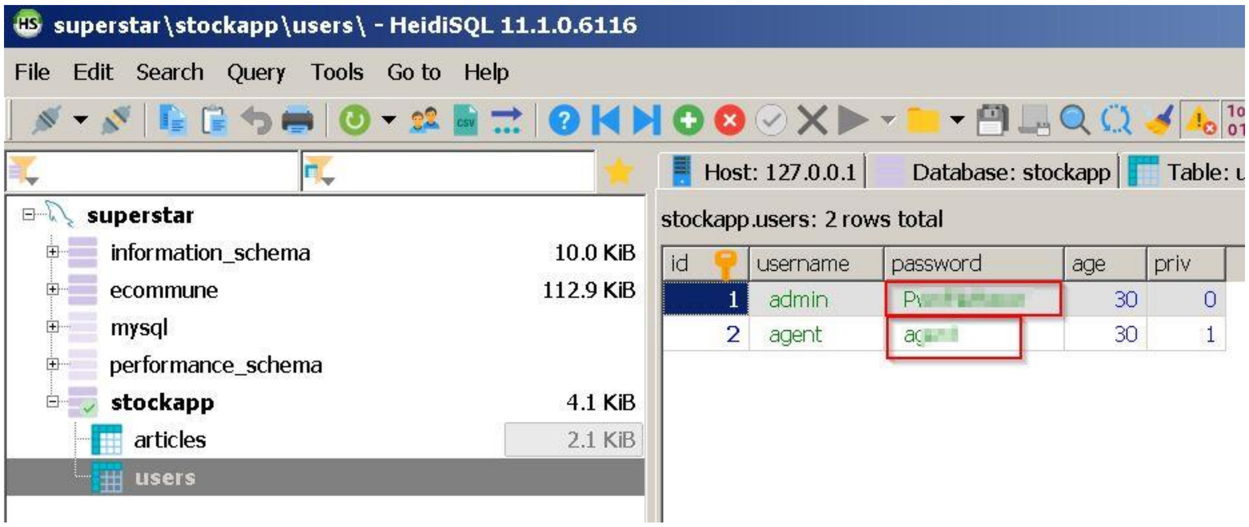


Fig. Base score 7.2

Description: Vulnerability that occurs when passwords are stored in plaintext format. Storing passwords in plaintext means that the passwords are not encrypted or hashed and can be read by anyone who has access to the storage location, such as a database or configuration file which leads the attacker to access the sensitive information.

Exploitation:

- By using the DB management tool like HieldiSQL press the connect option
- Enter the DB username and password and that connects to the StockApp database tables.
- Where all the usernames and passwords are visible



```
superstar
  information_schema    10.0 KiB
  ecommune             112.9 KiB
  mysql
  performance_schema
  stockapp               4.1 KiB
    articles             2.1 KiB
    users
```

stockapp.users: 2 rows total

id	username	password	age	priv
1	admin	Pv[redacted]	30	0
2	agent	ag[redacted]	30	1

Fig:1 Password stored as plain text

Remediation:

- Use strong encryption: Passwords should be encrypted using strong encryption algorithms such as AES or bcrypt. Encryption should be used both during storage and transmission of the password. Using a one-way hash function like bcrypt is recommended since it is not reversible, which makes it harder for an attacker to recover the original password.
- Use MFA, Strong Password like Uppercase, number, special character, use of Google Authenticator, Microsoft Authenticator, YubiKey and mobile app
- Use of hashing and salting: It enforcing strong password complexity requirements, implementing secure password storage practices, and regularly educating users about password hygiene, should also be considered as part of a comprehensive security strategy. A salt is a random string of data that is added to the password before it is hashed. This makes it harder for attackers to use precomputed hash tables to crack passwords since Each password has a unique salt value.

Title: Improper Privileges Separation

The Common Weakness Enumeration (CWE) ID: : https://cwe.mitre.org/data/definitions/272.html

CVSS 3.1 Base Score Metrics ():

CVSS v3.1 Vector CVSS v3.1 AV:N/AC:H/PR:N/UI:N/S:U/C:L/I:L/A:N •

Base Score: 7.5

Risk Level: High

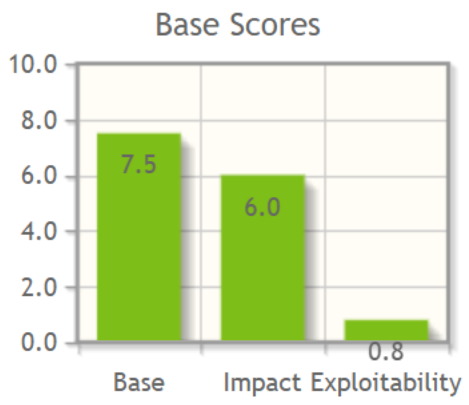
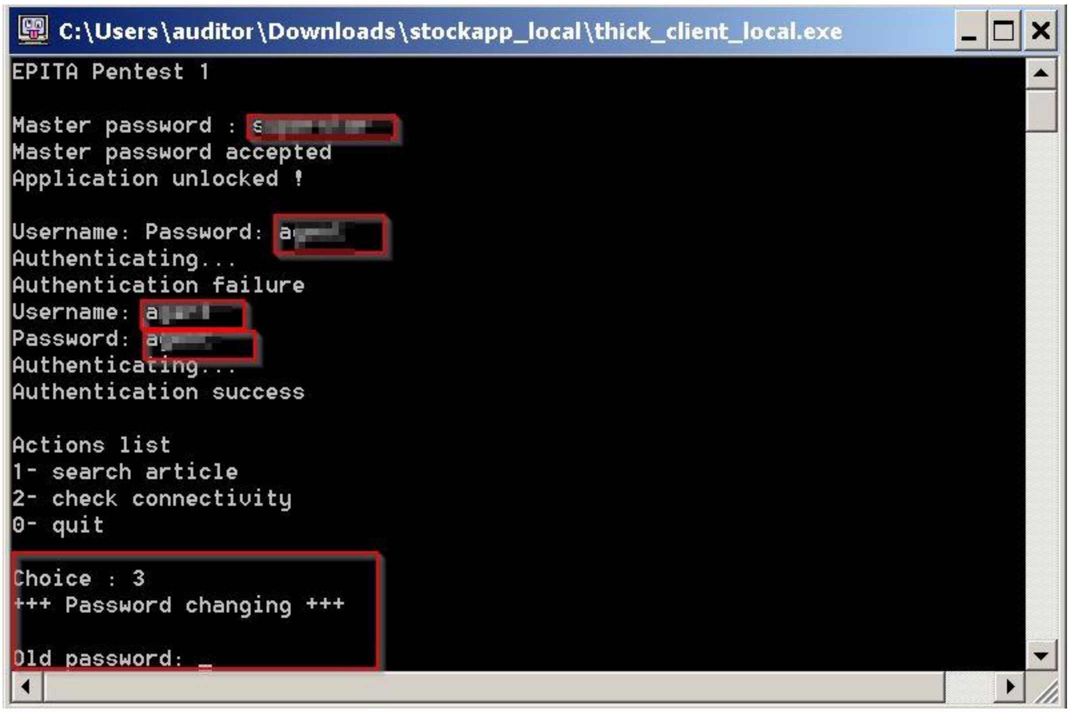


Fig. base score 7.5

Description: Vulnerability related to a least privilege violation. It occurs when an application or system grants more privileges to a user, program, or process than necessary to perform their tasks or functions. This can allow an attacker to gain elevated privileges, which can result in unauthorized access, data theft, or other malicious activities

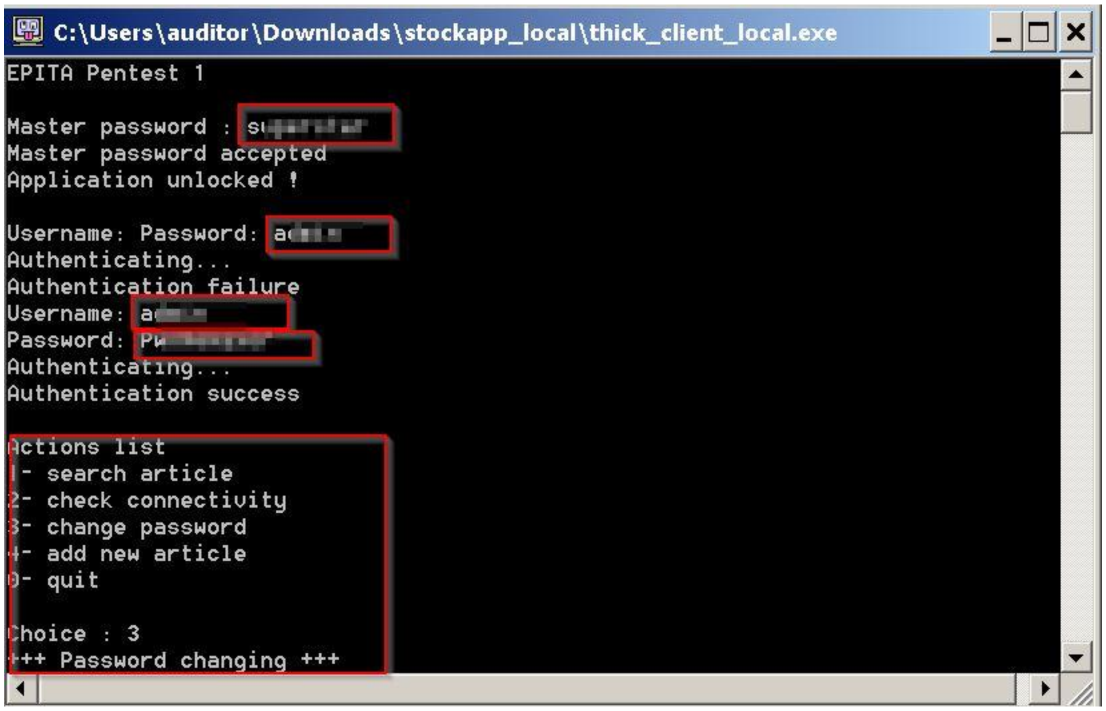
Exploitation:

- Open and Login to the application by entering master password
- And login as user with the agent credentials
- Enter the choice '3' which is not available in choice list
- Users can get access to change passwords which is restricted for Admin.
- And same way login with admin credentials it have same privileged for both admin and user to access change password.



```
C:\Users\auditor\Downloads\stockapp_local\thick_client_local.exe
EPITA Pentest 1

Master password : s[redacted]
Master password accepted
Application unlocked !

Username: Password: a[redacted]
Authenticating...
Authentication failure
Username: a[redacted]
Password: a[redacted]
Authenticating...
Authentication success

Actions list
1- search article
2- check connectivity
0- quit

Choice : 3
+++ Password changing +++

Old password: _
```

Fig. access change password from user account



```
C:\Users\auditor\Downloads\stockapp_local\thick_client_local.exe
EPITA Pentest 1

Master password : su[redacted]
Master password accepted
Application unlocked !

Username: Password: ad[redacted]
Authenticating...
Authentication failure
Username: ad[redacted]
Password: Pw[redacted]
Authenticating...
Authentication success

Actions list
1- search article
2- check connectivity
3- change password
4- add new article
0- quit

Choice : 3
+++ Password changing +++
```

Fig. access change password from admin account

Remediation:

- Implementing the principle of least privilege, by giving users, programs, or processes only the minimum necessary permissions to perform their tasks, granting permissions based on roles or responsibilities. Implementing access control mechanisms, such as file permissions, network segmentation, and firewalls, regularly reviewing and updating security policies to ensure they are not too permissive

Title: SQL Injection

The Common Weakness Enumeration (CWE) ID: https://cwe.mitre.org/data/definitions/89.html

CVSS v3.1 Vector AV:N/AC:H/PR:N/UI:N/S:U/C:H/I:L/A:L

Base Score: 7.0

Risk Level: High

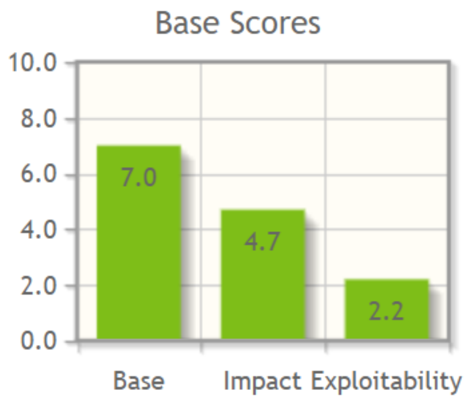


Fig. Base scores 7.0

Description: The product incorporates input from an external component to construct an SQL command. However, it neglects to remove or inadequately removes certain elements that have the potential to modify the intended SQL command. This oversight can have adverse effects on downstream components, potentially leading to unintended consequences or vulnerabilities in the system. Proper handling and sanitization of input data are crucial to ensure that the constructed SQL commands remain intact and free from any unintended alterations that could compromise the system's integrity and security.

Exploitation:

- Open the desktop app application and Enter the master password to access the app
- For the user admin, enter the phrase "“or”"=”" and the application get accessible.
- From the above it shows the hacker can easily login to the application by injecting sql command.

Remediation:

- Implementing robust input validation and sanitization routines helps to maintain the integrity of the application and protect against potential security breaches. It ensures that user input is treated as data rather than executable code, reducing the risk of unauthorized access or manipulation
- Input Sanitization

Title: Improper Neutralization of Special Elements used in an OS Command

The Common Weakness Enumeration (CWE) ID: https://cwe.mitre.org/data/definitions/78.html

CVSS v3.1 Vector AV:N/AC:L/PR:L/UI:N/S:C/C:H/I:H/A:H

Base Score: 9.9

Risk Level: High

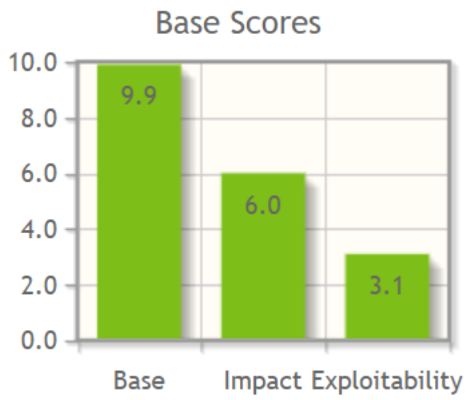
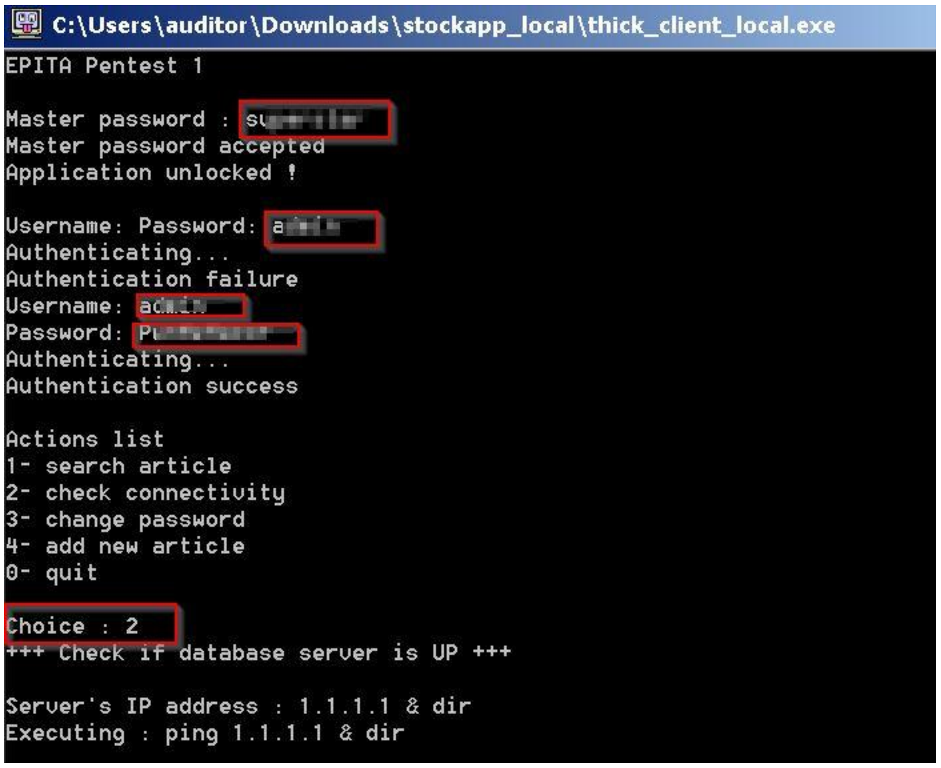


Fig. Base Scores 9.9

Description: Improper neutralization of special elements used in an OS command, also known as OS command injection, occurs when an application or system fails to adequately validate user input used to construct operating system commands. This vulnerability enables attackers to execute arbitrary commands on the system, potentially compromising its security and integrity.

Exploitation:

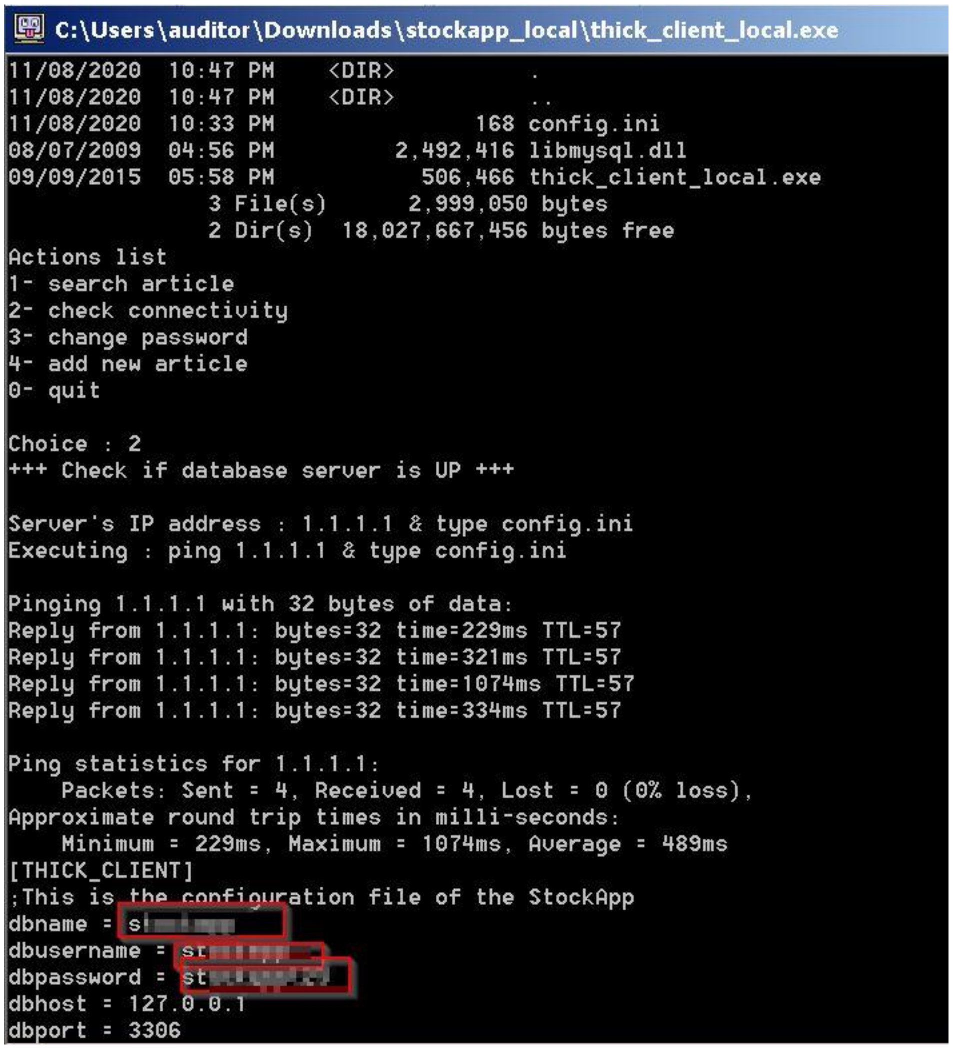
- Run the “thick_client_local.exe” application and enter the “Master” password to access the application.
- Login to the application and enter “2” when the application prompts for a choice in the action list



```
C:\Users\auditor\Downloads\stockapp_local\thick_client_local.exe
EPITA Pentest 1

Master password : su[redacted]
Master password accepted
Application unlocked !

Username: Password: a[redacted]
Authenticating...
Authentication failure
Username: ad[redacted]
Password: Pu[redacted]
Authenticating...
Authentication success

Actions list
1- search article
2- check connectivity
3- change password
4- add new article
0- quit

Choice : 2
+++ Check if database server is UP +++

Server's IP address : 1.1.1.1 & dir
Executing : ping 1.1.1.1 & dir
```

Fig. Select option 2 read the directory

- Enter the phrase “127.0.0.1 & dir” when the use is prompted for the server’s Ip address.



```
C:\Users\auditor\Downloads\stockapp_local\thick_client_local.exe
11/08/2020  10:47 PM    <DIR>          .
11/08/2020  10:47 PM    <DIR>          ..
11/08/2020  10:33 PM               168 config.ini
08/07/2009  04:56 PM         2,492,416 libmysql.dll
09/09/2015  05:58 PM           506,466 thick_client_local.exe
               3 File(s)      2,999,050 bytes
               2 Dir(s)  18,027,667,456 bytes free
Actions list
1- search article
2- check connectivity
3- change password
4- add new article
0- quit

Choice : 2
+++ Check if database server is UP +++

Server's IP address : 1.1.1.1 & type config.ini
Executing : ping 1.1.1.1 & type config.ini

Pinging 1.1.1.1 with 32 bytes of data:
Reply from 1.1.1.1: bytes=32 time=229ms TTL=57
Reply from 1.1.1.1: bytes=32 time=321ms TTL=57
Reply from 1.1.1.1: bytes=32 time=1074ms TTL=57
Reply from 1.1.1.1: bytes=32 time=334ms TTL=57

Ping statistics for 1.1.1.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 229ms, Maximum = 1074ms, Average = 489ms
[THICK_CLIENT]
;This is the configuration file of the StockApp
dbname = s[redacted]
dbusername = st[redacted]
dbpassword = st[redacted]
dbhost = 127.0.0.1
dbport = 3306
```

Fig. read the config.ini file

- The injected OS command gets executed successfully.

Remediation:

- All user input should be carefully validated to ensure it adheres to expected formats and does not contain any malicious characters or commands.
- By enforcing strict input validation, utilizing input sanitization mechanisms, and applying principle of least privilege, organizations can significantly reduce the likelihood of OS command injection attacks.

Title: Exposition of credentials in Memory Dump

The Common Weakness Enumeration (CWE) ID : https://cwe.mitre.org/data/definitions/244.html

CVSS v3.1 Vector AV:L/AC:H/PR:L/UI:R/S:C/C:H/I:H/A:H

Base Score: 7.5

Risk Level: High

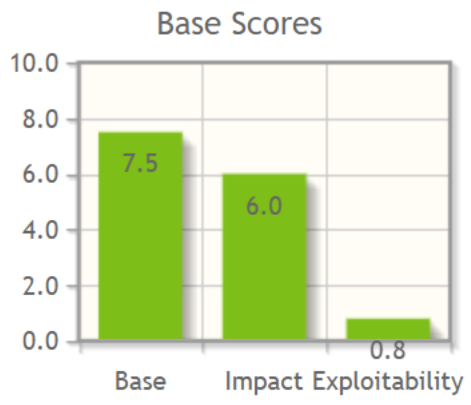


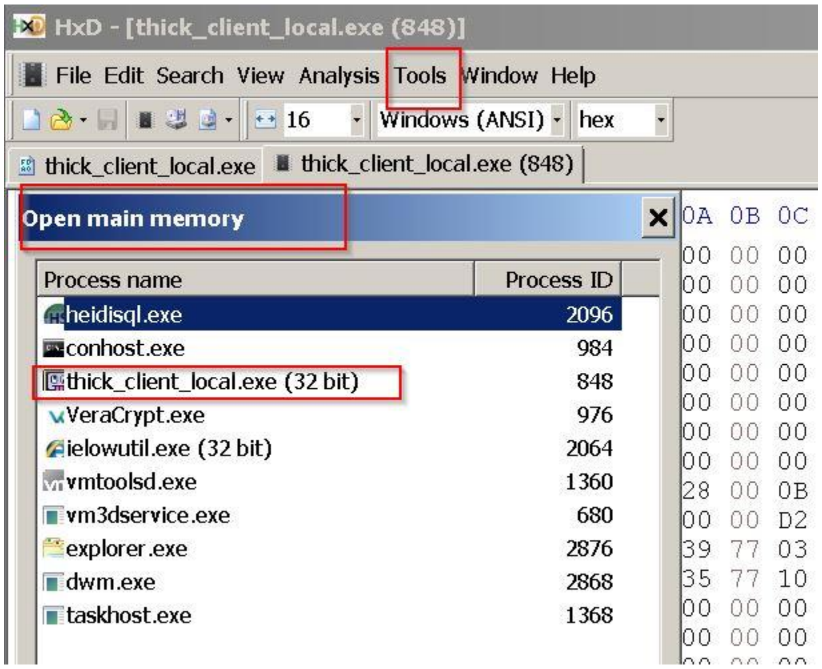
Fig Base score 7.5

Description: Secret data such as password was not removed from the memory even after sometime which is a vulnerability and attacker can read the sensitive data using memory dump or other method.

Exploitation:

- Start and open the desktop app and login using the user admin and password Pwnmehaxor

- Open HXD and go to tools – Select open main memory and form the list select the think_client_local.exe and click ok.

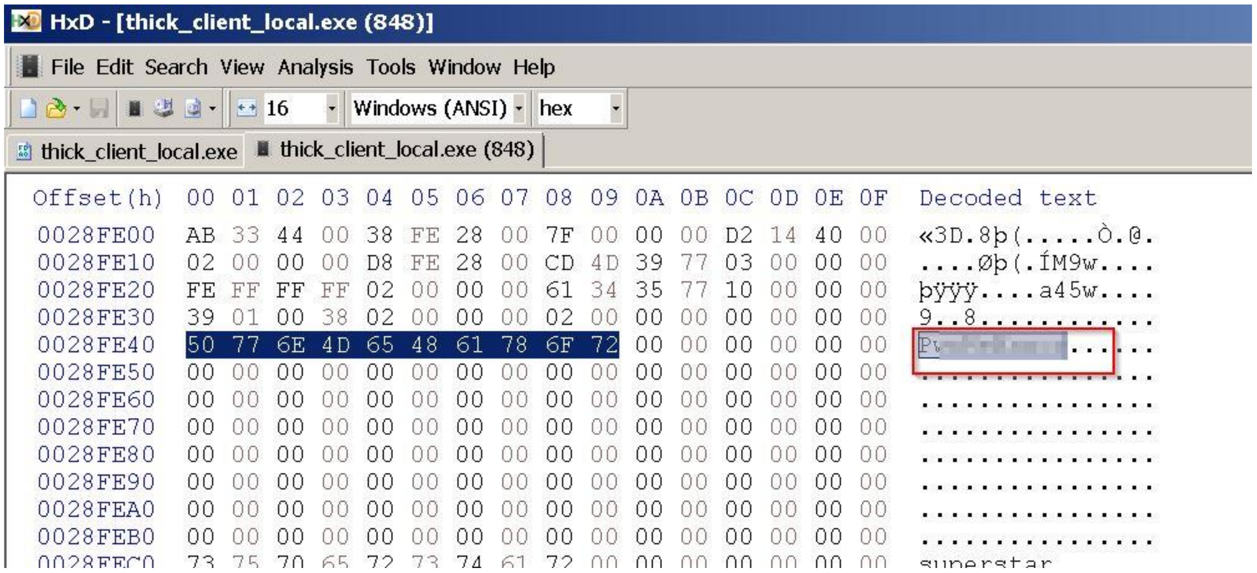

HxD - [thick_client_local.exe (848)]
File Edit Search View Analysis Tools Window Help
16
Windows (ANSI)
hex
thick_client_local.exe
thick_client_local.exe (848)
Open main memory

Process name	Process ID
heidisql.exe	2096
conhost.exe	984
thick_client_local.exe (32 bit)	848
VeraCrypt.exe	976
ielowutil.exe (32 bit)	2064
vmtoolsd.exe	1360
vm3dservice.exe	680
explorer.exe	2876
dwm.exe	2868
taskhost.exe	1368

```
0A 0B 0C
00 00 00
00 00 00
00 00 00
00 00 00
00 00 00
00 00 00
00 00 00
00 00 00
28 00 0B
00 00 D2
39 77 03
35 77 10
00 00 00
00 00 00
```

Fig.Hxd Tool

- On the right hand side search for string PwnmeHaxor


HxD - [thick_client_local.exe (848)]
File Edit Search View Analysis Tools Window Help
16
Windows (ANSI)
hex
thick_client_local.exe
thick_client_local.exe (848)

```
Offset(h) 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F  Decoded text
0028FE00  AB 33 44 00 38 FE 28 00 7F 00 00 00 D2 14 40 00  «3D.8þ(.....Ò.@.
0028FE10  02 00 00 00 D8 FE 28 00 CD 4D 39 77 03 00 00 00  ....Øþ(.ÍM9w....
0028FE20  FE FF FF FF 02 00 00 00 61 34 35 77 10 00 00 00  þÿÿÿ....a45w....
0028FE30  39 01 00 38 02 00 00 00 02 00 00 00 00 00 00 00  9..8............
0028FE40  50 77 6E 4D 65 48 61 78 6F 72 00 00 00 00 00 00  [redacted]......
0028FE50  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0028FE60  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0028FE70  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0028FE80  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0028FE90  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0028FEA0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0028FEB0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0028FEC0  73 75 70 65 72 73 74 61 72 00 00 00 00 00 00 00  superstar
```

Fig. Search for the password PwnMeHaxr

- Password of the admin still exist in the memory dump even after 10-15 minutes of login.

Remediation:

- As soon as the secret data is used, it should be taken out of the memory.

- Implement appropriate algorithms to wipe or zero size sensitive data. Overwriting the memory with random or null values can help prevent unauthorized access to sensitive information.

Title: Unprotected Transport of Credentials

The Common Weakness Enumeration (CWE) ID: https://cwe.mitre.org/data/definitions/523.html

CVSS 3.1 Base Score Metrics (): AV:L/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H

Base Score: 8.4

Risk Level: High

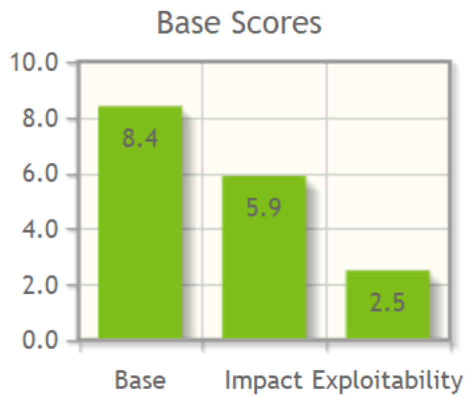


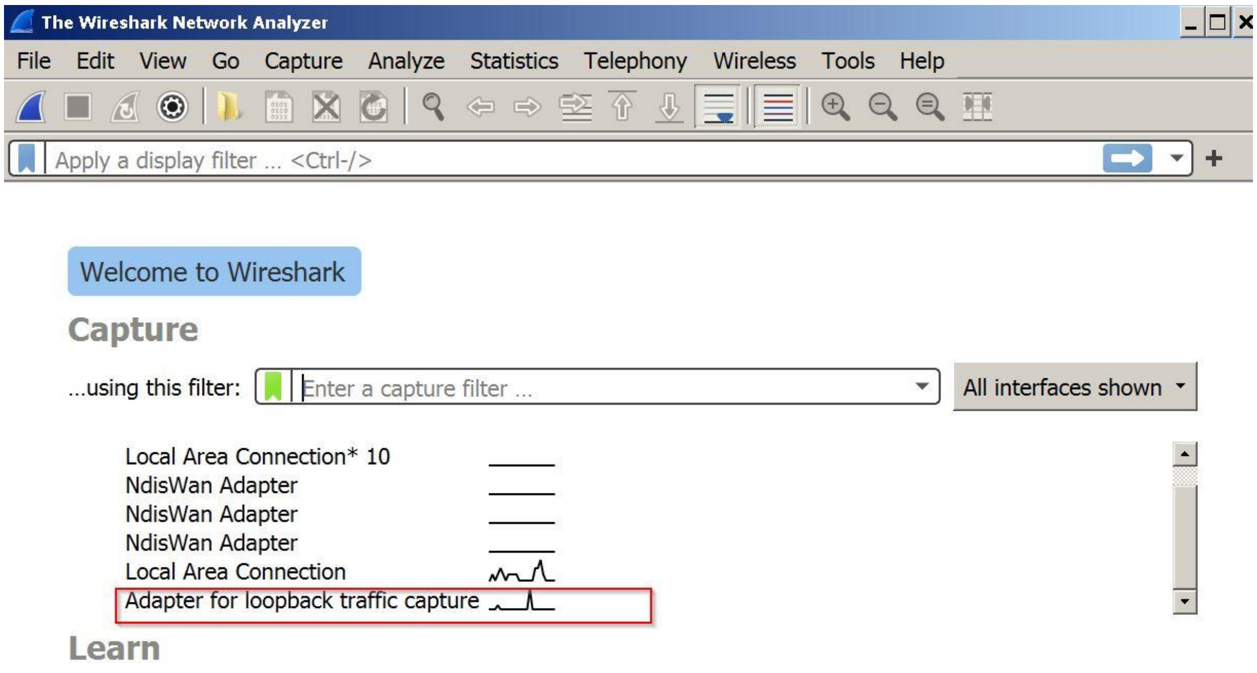
Fig. Base score 8.4

Description: Unprotected transport of credentials. It occurs when sensitive information, such as usernames and passwords, are transmitted over an insecure network or channel and improper certificate validations, making it vulnerable to interception by an attacker.

Exploitation:

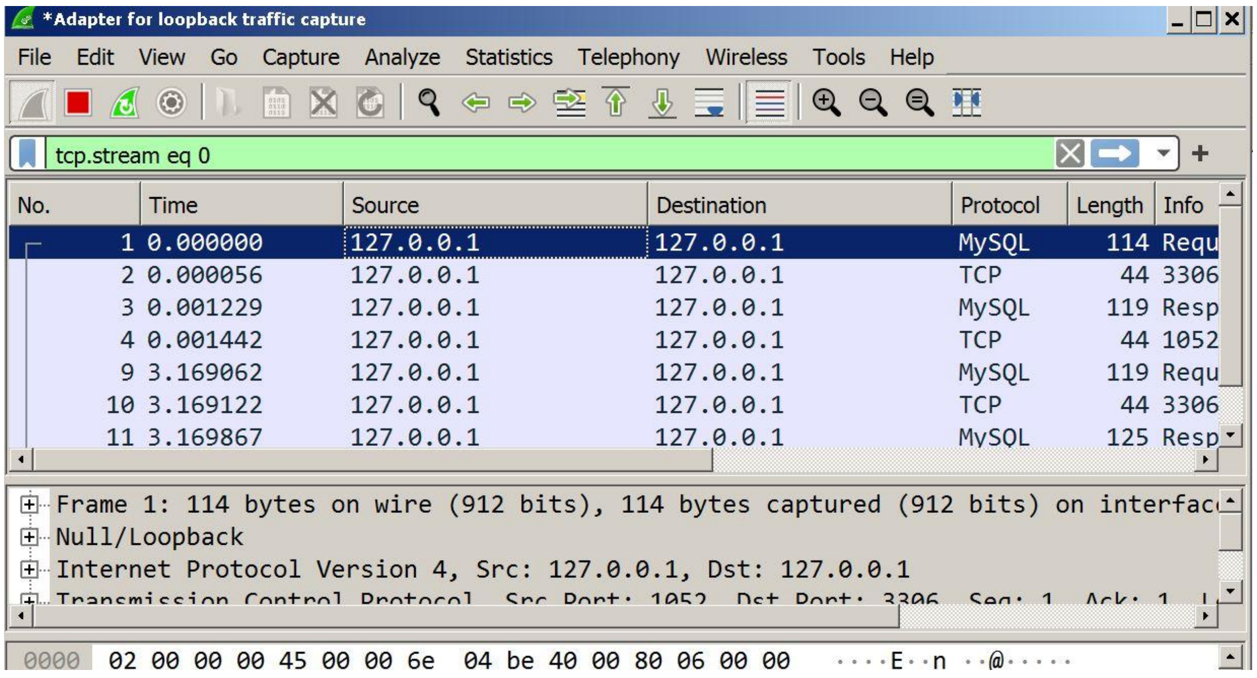
Open the app thick_client_local.exe application enter with credentials using username agent and password agent.

By using the wireshark tool and selecting the option Adapter for loopback traffic capture as config there is use localhost as host ip 127.0.0.1


The Wireshark Network Analyzer
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter … <Ctrl-/>
Welcome to Wireshark
Capture
…using this filter:
Enter a capture filter …
All interfaces shown
Learn

```
Local Area Connection* 10
NdisWan Adapter
NdisWan Adapter
NdisWan Adapter
Local Area Connection
Adapter for loopback traffic capture
```

Fig. selecting the capture option Adapter for loopback traffic capture


*Adapter for loopback traffic capture
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

tcp.stream eq 0

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	127.0.0.1	127.0.0.1	MySQL	114	Requ
2	0.000056	127.0.0.1	127.0.0.1	TCP	44	3306
3	0.001229	127.0.0.1	127.0.0.1	MySQL	119	Resp
4	0.001442	127.0.0.1	127.0.0.1	TCP	44	1052
9	3.169062	127.0.0.1	127.0.0.1	MySQL	119	Requ
10	3.169122	127.0.0.1	127.0.0.1	TCP	44	3306
11	3.169867	127.0.0.1	127.0.0.1	MySQL	125	Resp

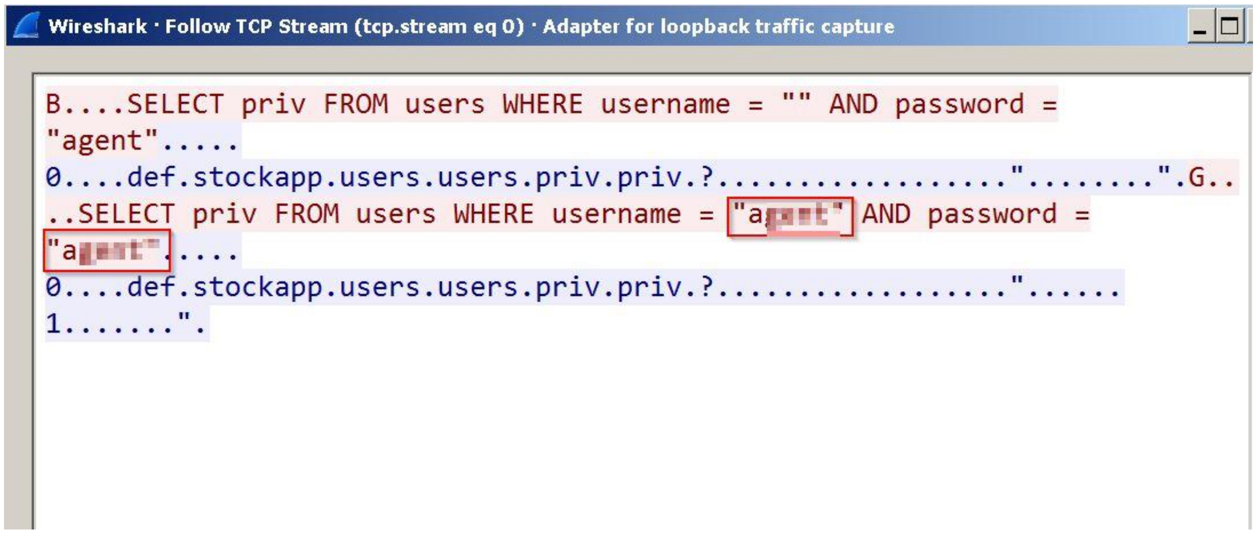
```
Frame 1: 114 bytes on wire (912 bits), 114 bytes captured (912 bits) on interfac
Null/Loopback
Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
Transmission Control Protocol, Src Port: 1052, Dst Port: 3306, Seq: 1, Ack: 1, L
```

```
0000  02 00 00 00 45 00 00 6e  04 be 40 00 80 06 00 00   ····E··n ··@·····
```

Fig. Wireshark too.

Select tcp.stream the right click Follow option and TCP stream

Open the Wireshark – Follow TCP stream windows so there appear username and password



Wireshark · Follow TCP Stream (tcp.stream eq 0) · Adapter for loopback traffic capture

```
B....SELECT priv FROM users WHERE username = "" AND password =
"agent".....
0....def.stockapp.users.users.priv.priv.?................."........".G..
..SELECT priv FROM users WHERE username = "a[redacted]" AND password =
"a[redacted]".....
0....def.stockapp.users.users.priv.priv.?................."......
1......."..
```

Fig. find out username and password

Remediation:

Essential to implement secure communication protocols, such as HTTPS, TLS, or SSH, to encrypt the transmission of sensitive information. Additionally, it is crucial to validate certificates and use strong encryption algorithms to prevent decryption by an attacker**.**