**Security Audit Report**

**Prepared for**

**esiea\_lourd.exe**

**application**

Prepared by

**Bashetty Arun Kumar**

**Master of Computer Security**

**22-December-2018**

**TABLE OF CONTENTS:**

EXECUTIVE SUMMARY……………………………………………………………………**3**

Useful Steps to do Audit…………………………………………………**4**

Secrets stored in plaintext…………………………………………**5**

Secrets/passwords shown on the screen………………**8**

Weak passwords accepted……………………………………………………**10**

Bad user profile segregation………………………………………**13**

secrets stored in plaintext within the configuration file of the application………………**15**

passwords stored in plaintext within

the database……………………………………………………………………………………**17**

Technical information disclosure………………………………**19**

SQL Injection…………………………………………………………………………………**21**

system command injection……………………………………………………**23**

**EXECUTIVE SUMMARY**

This report represents the **Vulnerability Analysis Report** for the application esiea\_lourd.exe as required by the Software Security administrator. We described clearly about the risk associated with the different vulnerabilities those we found during the security evaluation.

Totally we found around 9 vulnerabilities during evaluation and in that some of them are more critical and we need to take a quick and immediate actions to secure the application and data base.

We also enclose our recommendations to correct suggesting issues at each vulnerability.

**Tools used for Evaluation:**

HxD

IDA Pro

DbVisualizer 9.2.10

Command Prompt.

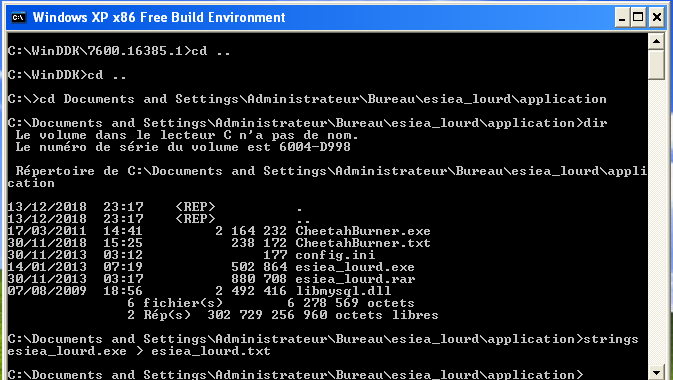
**Useful Steps to do Audit:**

Before entering vulnerabilities , we must open the application from your system where you downloaded in your system.

Using command Prompt, I gave below path:

**“C:\Documents and Settings\Administrateur\Bureau\esiea\_lourd\application”**

We can find **esiea\_lourd.exe** file and using strings command you can extract into text files and save the file using .txt extension.



**NOTE:**

This esiea\_lourd.txt file will help you to find the vulnerabilities inside the application.

**Vulnerability 1:**

**Secrets stored in plaintext within the application:**

**CRITICALITY INDEX:**

|  |  |
| --- | --- |
| AREA | INDEX |
| Risk | **MEDIUM** |
| Exploitation | **HIGH** |
| Correction | **EASY** |

**DESCRIPTION:**

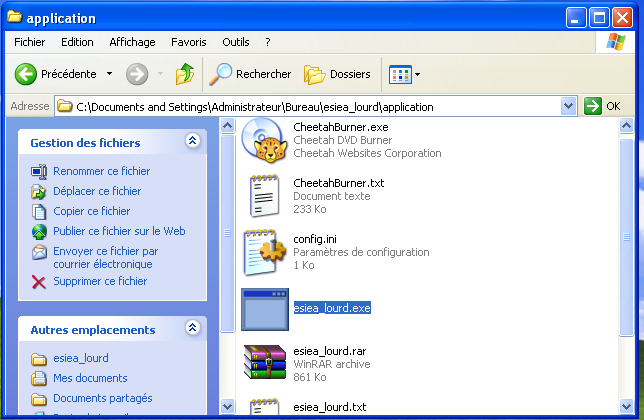
Currently we are working on **esiea\_lourd.exe** file and we can check that password is directly available publicly, we can use some tools like **HxD** and common **notepad** to get the username and password details.

**EXPLOITATION:**

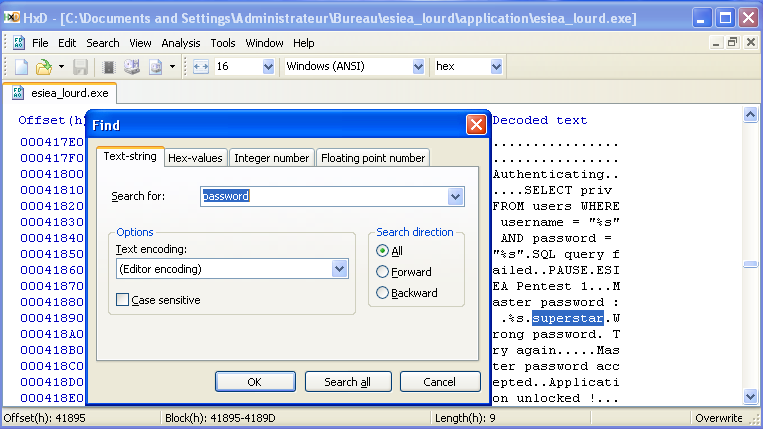
Most important and Delicate Information is leaked from application to user.

**Steps to do first vulnerability:**

**1st one using HxD,**



Import the esiea\_lourd.exe file in HxD and use below following steps,

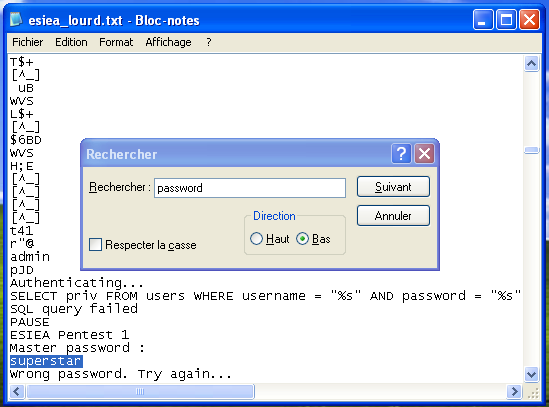


=>First, you must extract the **esiea\_lourd.rar file** , we will get a config and application files,

=>Import the esiea\_loaurd.exe file in Hxd application, use **CNTRL+F** to find the text-string and type **password** and enter. You can see the password to use this application. i.e., superstar

=>One more method to check the plain text using notepad , as I already explained in 1st slide, open the esiea\_lourd.exe file

=>use **CNTRL+F** to Rechercher the file and type **password**, it will display Master Password: **superstar.**



**RECOMMENDATION:**

For this Vulnerability, I can say that **Encrypting the passwords** is much better,

If user, find the encrypted password using this tool, it is very hard to decrypt without knowing the key.

**Vulnerability 2:**

**Secrets/passwords shown on the screen.**

**CRITICALITY INDEX:**

|  |  |
| --- | --- |
| AREA | INDEX |
| Risk | **MEDIUM** |
| Exploitation | **MEDIUM** |
| Correction | **EASY** |

**DESCRIPTION:**

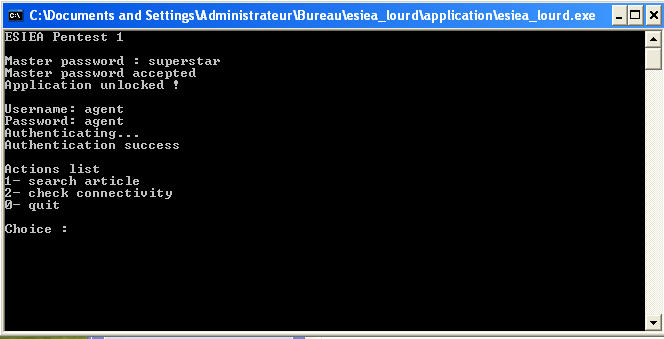
Passwords are a royal pain in our digital lives. If you create a simple one, it isn’t safe, if you create a hard one , you will never remember it.

In this vulnerability we found the username and password of ESIEA Pentest 1,those are displayed directly.

I used that username: **agent** and password: **agent** to login into application and I am succeeded.

**EXPLOITATION:**

User can access the secret data by giving easy user name and password



**RECOMMENDATION:**

1.The best recommendation for this vulnerability is use some **special characters** in your password.

2.for each mail id you can give a unique password.

3.if you replace password with “\*” also not a good option, better to avoid it, because number of characters are counted.

4. Try to put a big password , as a human we do not have fish memory, so use some applications like **KeePass 2,** to store passwords.

**Vulnerability 3**

**weak passwords accepted:**

**CRITICALITY INDEX:**

|  |  |
| --- | --- |
| AREA | INDEX |
| Risk | **HIGH** |
| Exploitation | **HIGH** |
| Correction | **MEDIUM** |

**DESCRIPTION:**

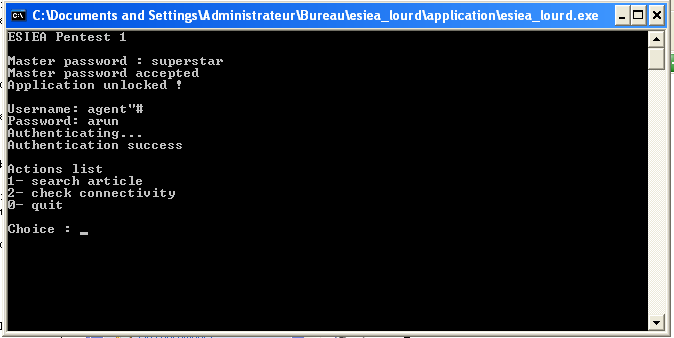
For this Vulnerability we used to test on this application and [www.e-commune.org](http://www.e-commune.org), finally we find that authentication is successful.

In this vulnerability we found that with of using password also we login successfully into data base.

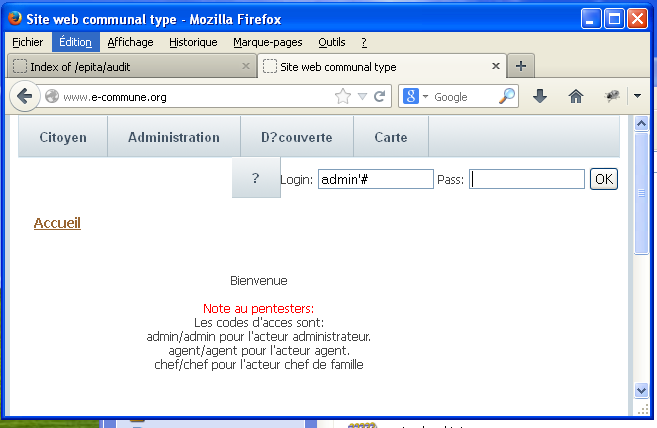
**Example**: username is agent and Password if you don’t know, don’t worry, you can easily access the account by giving the username as agent’# and password you can leave empty or give any name, you can enter into user data base.

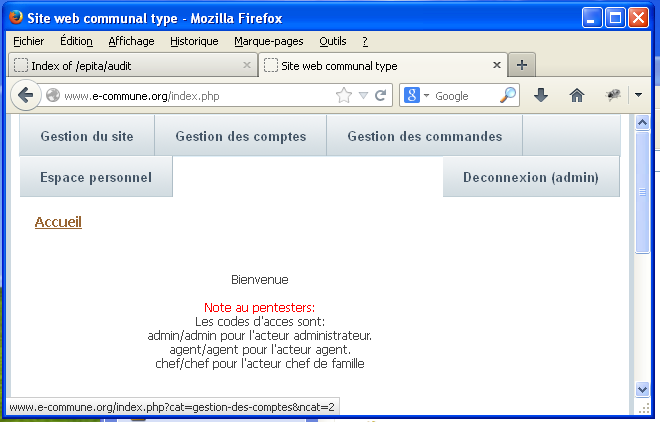
**EXPLOITATION:**

Useful information can be leaked without giving the password.



In above Screenshot, we can find that I gave username as agent”# and given fake password, but it shows authentication success.





In web site,[www.e-commune.org](http://www.e-commune.org) , we gave Login as admin’# and password I leaved as empty, I clicked on “OK”, Directly, I accessed into web site.

**RECOMMENDATION:**

This attack is called **Scripting attack.**

When the programmer writing the program for the application, programmer can exclude the scripting commands.

When the user enters the scripting commands in username, programmer can set an error message to user.

**Vulnerability 4**

**Bad user profile segregation:**

**CRITICALITY INDEX:**

|  |  |
| --- | --- |
| AREA | INDEX |
| Risk | **MEDIUM** |
| Exploitation | **MEDIUM** |
| Correction | **EASY** |

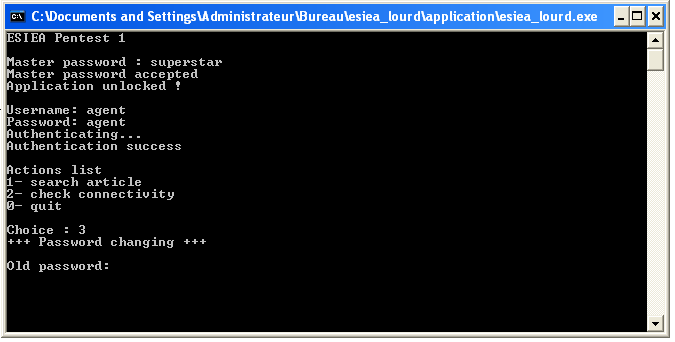
**DESCRIPTION:**

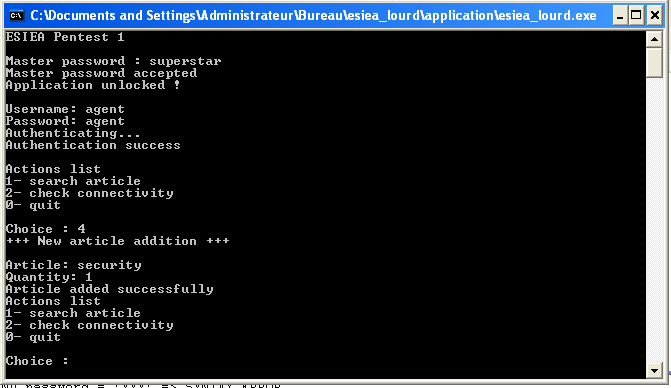
After enter username and password you can see that authentication is success and it will be showing action list to perform option.

In option list we can see three option to perform action, but the main problem is if you give extra option 3 and press enter it will ask the password changing. Option 4 also give extra information.

**EXPLOITATION:**

Hacker can create a new password and it would be dangerous for company.





**RECOMMENDATION:**

For this vulnerability I can say that they should only be accessible for the admin

If admin want more option, admin can add by using his password.

**Vulnerability 5:**

**secrets stored in plaintext within the configuration file of the application:**

**CRITICALITY INDEX:**

|  |  |
| --- | --- |
| AREA | INDEX |
| Risk | **MEDIUM** |
| Exploitation | **HIGH** |
| Correction | **MEDIUM** |

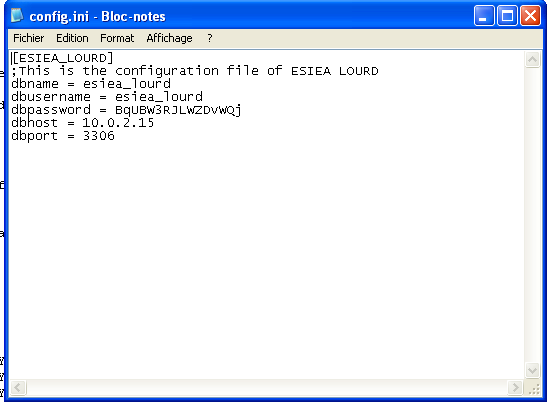
**DESCRIPTION:**

When you extracted the esiea\_lourd.zip application, you can see the one esiea\_lourd.exe file and one more configuration file.

Inside configuration file, we can find the data base name, db username, db password, host and port number

**EXPLOITATION:**

User can access the data base using local host and port number very easily and he can modify or retrieve the useful data.



**RECOMMENDATION:**

The most common recommendation is do not insert db username and db password details inside configuration files.

Admin can store this data inside his data base and he can manage the data base structure.

**Vulnerability 6:**

**passwords stored in plaintext within the database:**

**CRITICALITY INDEX:**

|  |  |
| --- | --- |
| AREA | INDEX |
| Risk | **MEDIUM** |
| Exploitation | **HIGH** |
| Correction | **EASY** |

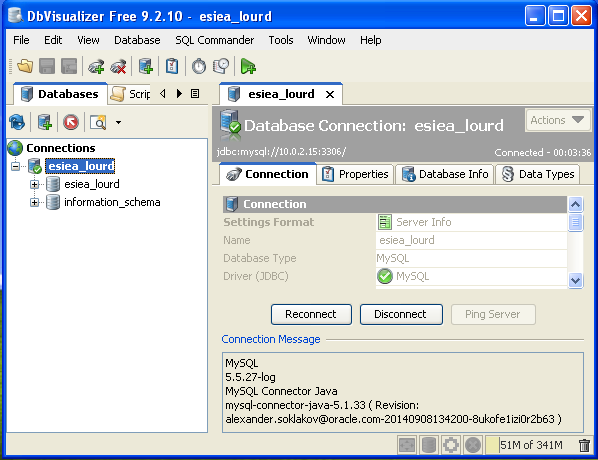
**DESCRIPTION:**

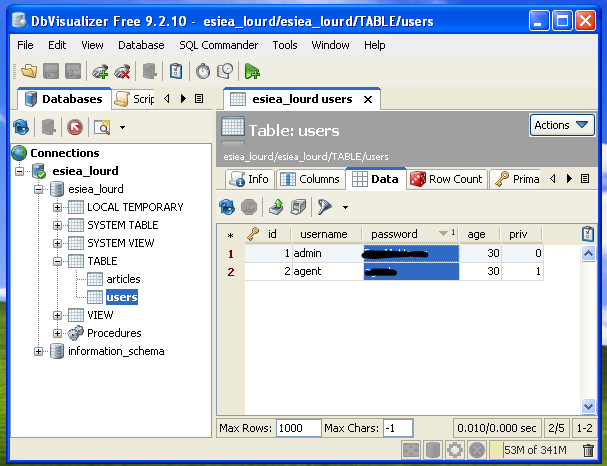
We can use the application called **DbVisualizer 9.2.10** to access the data base of application, you can use db username and db password. You can change the dp host number and dp port.

After entering all details, you can see that dp host is connect to your current host and do not forget to run easy php application. Inside user profile we can clearly seen the admin username and password.

**EXPLOITATION:**

User can access the username and password easily. Using admin credentials user can access, modifying and erase the data inside the data base.





**RECOMMENDATION:**

The better recommendation for this vulnerability is to secure the data base with using secret fully protected password.

Inside the database also, admin can set a password in encrypted text.

**Vulnerability 7:**

**Technical information disclosure:**

**CRITICALITY INDEX:**

|  |  |
| --- | --- |
| AREA | INDEX |
| Risk | **MEDIUM** |
| Exploitation | **MEDIUM** |
| Correction | **EASY** |

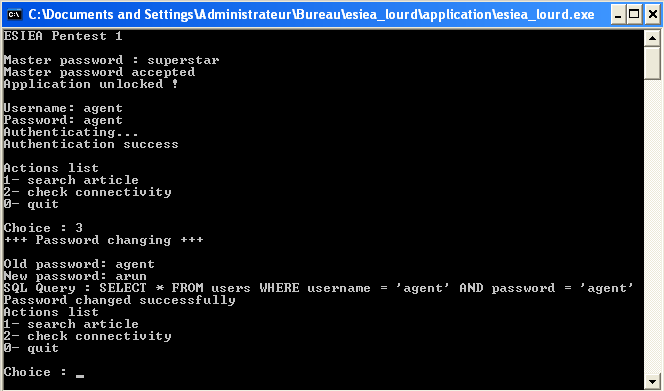
**DESCRIPTION:**

When we login into **ESIEA pentest 1** using password as **superstar,** it displays application is unlocked. After login into account it will displays 3 options to choose for more information.

But if you give choice 3 and option 3 shows the executed SQL query to update the user’s password and if you give new password, as a admin you cannot access the application next time.

**EXPLOITATION:**

User can update the password and he can access the data base.



**RECOMMENDATION:**

The best recommendation for this vulnerability is user can set a mega super passwords option like,

When admin enter a password and select enter, user can get an OTP his mail id and if that OTP matches, then you can access otherwise it will display error.

One more possibility is, user can set an encrypted password by using **sha32 or md5** algorithms.

**Vulnerability 8:**

**SQL Injection:**

**CRITICALITY INDEX:**

|  |  |
| --- | --- |
| AREA | INDEX |
| Risk | **MEDIUM** |
| Exploitation | **HIGH** |
| Correction | **EASY** |

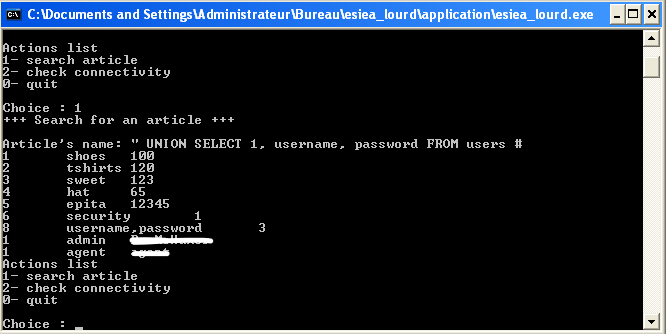
**DESCRIPTION:**

In the early days of the internet, building websites was straightforward: no JavaScript, no CSS. But as the web gained popularity, the need for more advanced technologies like ASP, JSP, PHP.

The SQL injection vulnerability is one of the most dangerous issues for data confidentiality and integrity in web applications.

**EXPLOITATION:**

This SQL injection effectively removes the password verification and returns a dataset for an existing user-'admin' in this case. The attacker can now log in with an administrator account, without having to specify a password.



**RECOMMANDATION:**

**1.Don't use dynamic SQL – don't construct queries with user input:**

Even data sanitization routines can be flawed, so use prepared statements, parameterized queries or stored procedures instead whenever possible.

**2.Update and patch**:

Vulnerabilities in applications and databases that hackers can exploit using SQL injection are regularly discovered, so it's vital to apply patches and updates as soon as practical. A patch management solution might be worth the investment.

***Vulnerability 9:***

**system command injection:**

**CRITICALITY INDEX:**

|  |  |
| --- | --- |
| AREA | INDEX |
| Risk | **MEDIUM** |
| Exploitation | **MEDIUM** |
| Correction | **EASY** |

**DESCRIPTION:**

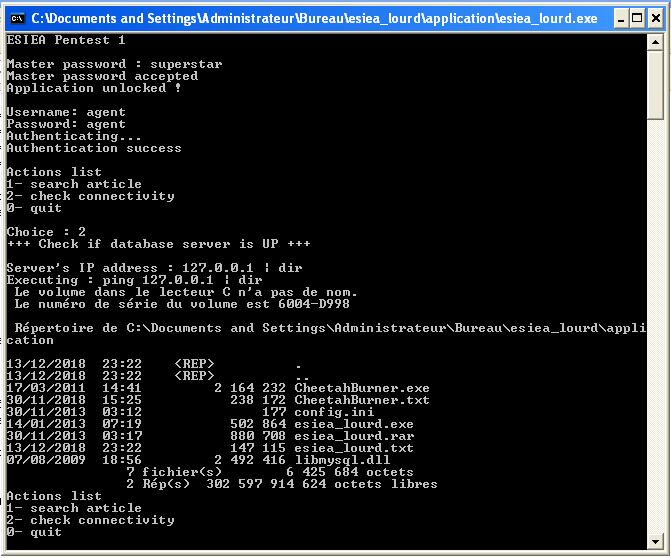
After enters the data base, where it is showing 3 options and if you select 2nd option, it is asking to enter data base server IP.

If you enter Internet Protocol address with & or ; or |, it will display all the directories inside the application.

**EXPLOITATION:**

Attacker can inject system commands by using those special characters. By using this attack attacker can access, modifies and delete he data.

This means attacker can easily take complete control over a web server; therefore, developers should be very careful how they pass user input into one of those functions



**RECOMMANDATION:**

In order to prevent an attacker from being able to insert special characters into the command, you should try to generally avoid system calls where possible.

Under all circumstances, avoid user input of any kind inside them unless it is necessary and deactivate that function in your language's configuration file if you don't need it