WEB APP SECURITY AUDIT

Abdallah Hamdan - Inaam Kabbara – Shiron dev Newton

Table of Contents

Synthesis

The context and scope

The report contains auditing information's which took place from 28th of July to 7th of July 2023 for web application site www.e-commune.org. The website originated the services of bureaucratic official tasks and missions.

The auditing was done by the Epita cyber sec team to find the vulnerabilities, and security issues within the web application and tested the security measures by exploiting it and providing mitigation strategy with description.

Vulnerabilities categories

Critical Vulnerabilities:

	Vulnerabilities	Done by:
1	Secrets / Credentials present in the HTML comments	Abdallah Hamdan
2	Weak password accepted	Abdallah Hamdan
3	Open directory listing	Inaam Kabbara
4	Carte has directly link for the shell	Shiron dev Newton
5	CCTV has password in plaintext	Inaam Kabbara

Medium Vulnerabilities:

	Vulnerabilities	Done by:
1	Technical information disclosure	Inaam Kabbara
2	Sensitive / private files publicly accessible	Abdallah Hamdan/Shiron Dev Newton
3	CCTV has client-side authentication	Shiron dev Newton

Low Vulnerabilities:

	Vulnerabilities	Done by:
1	Missing protection against brute-force attacks	Inaam Kabbara
2	XSS possible on the Mail sending and receiving	Shiron dev Newton
3	Publicly accessible user enumeration feature	Abdallah Hamdan

The recommendations

1. The client focus should be targeted on implementing a strong server-side authentication system.
2. Improving and removing the unauthorized access to the publicly available information
3. Importantly acting towards removing the shell command link in the source code and removing the shell code access to the regular users.
4. Improving the password policy on the cctv page and implementing strong password policy, if possible including it with strong hashes like sha 256, sha 1 or 2 , etc. for the website authentication for database e-commune zip files.
5. Installing the CSP security policy on the content of web pages to disallow attacks of XSS and client-side authentication.
6. Remove the direct access /admin direct listing to the page by restricting access to this function.

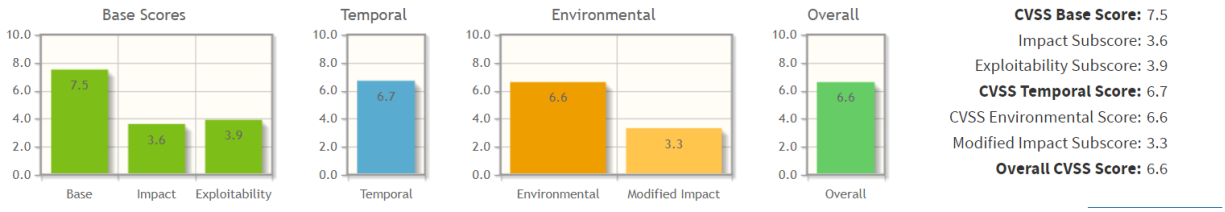
Vulnerabilities sheets

1. Secrets / Credentials present in the HTML comments:

The Common Weakness Enumeration (CWE) ID:

CWE-615: Inclusion of Sensitive Information in Source Code Comments

CVSS 3.1 Base Score Metrics:



CVSS 3.1 vector:

AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:N/A:N/E:P/RL:O/RC:C/CR:L/IR:H/AR:H/MAV:N/MAC:L/MPR:N/MUI:N/MS:U/MC:H/MI:L/MA:N

Score : 6.6

Risk : High

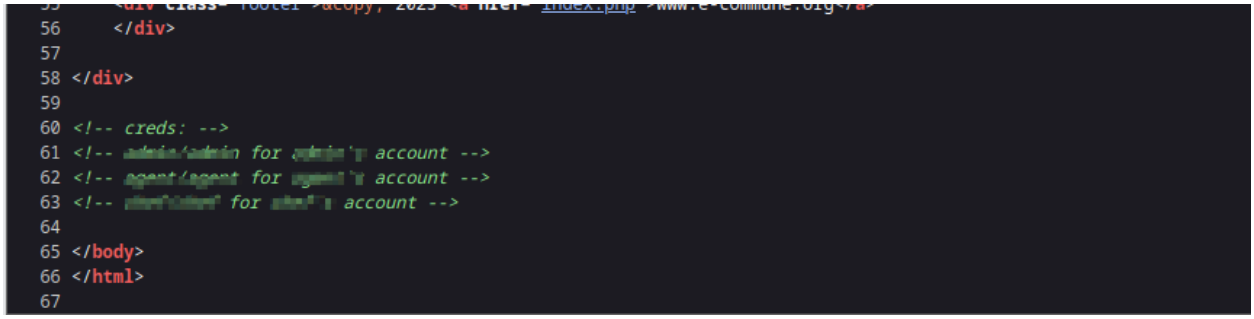
Description:

This vulnerability involves leaving sensitive information, such as username, password, filenames, old links, or other credentials, within HTML comments. This information can be accessed by anyone viewing the source code of the web application, potentially leading to unauthorized access or exposure of confidential data.

Exploitation:

- By checking the page source and going to the bottom of the code we can find some comment lines.
- These lines hold the credentials of the users (username and password)
- They extract usernames, passwords, or other credentials.
- The obtained information can be used for unauthorized access.
- Confidential data may be exposed or compromised.

View page source and scroll down.



Credentials will be found on the bottom of the page.

Remediation:

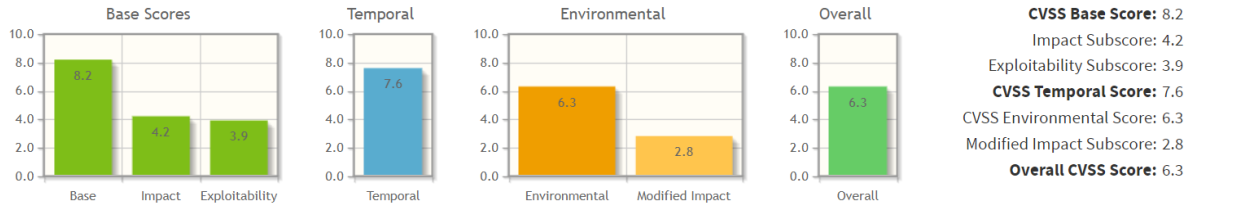
- Remove any sensitive information from HTML comments.
- Implement strict coding practices to prevent the inclusion of credentials or sensitive data in comments.
- Regularly review the source code to ensure no sensitive information is left in HTML comments.
- Educate developers about the risks and importance of not including sensitive information in comments.

2. Weak password accepted:

The Common Weakness Enumeration (CWE) ID:

CWE-521: Weak Password Requirements

CVSS 3.1 Base Score Metrics:



CVSS 3.1 vector:

AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:L/A:N/E:F/RL:O/RC:C/CR:L/IR:X/AR:X/MAV:N/MAC:L/MPR:N/MUI:N/MS:U/MC:H/MI:L/MA:N

Score : 6.3

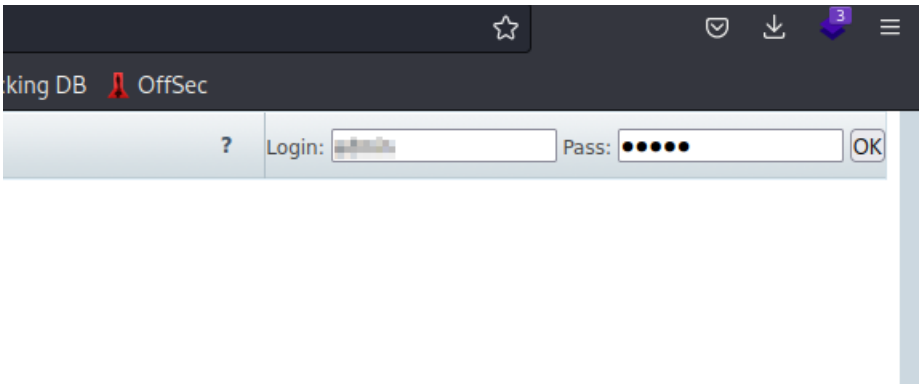
Risk : High

Description:

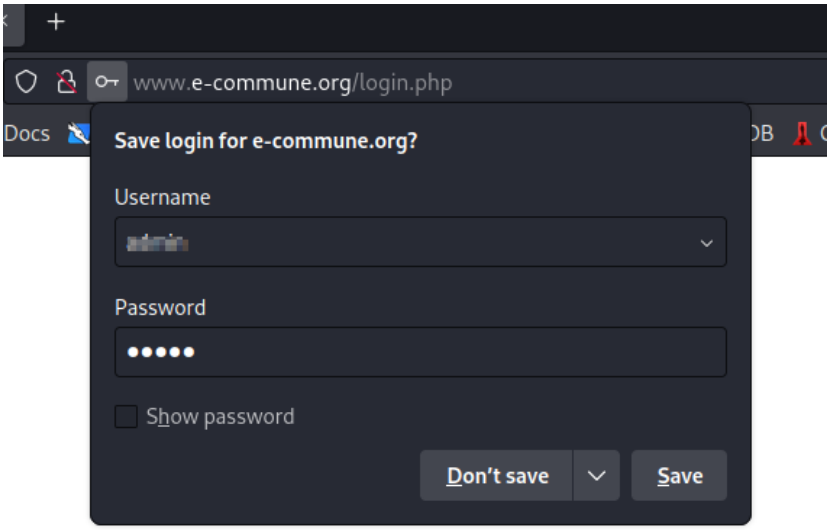
During the audit its observed that a system allows the use of weak or easily guessable passwords during user authentication. This increases the risk of unauthorized access since the authentication does not fulfill any required password protected policy to user accounts, as weak passwords can be easily compromised through brute-force attacks or dictionary-based password cracking methods.

Exploitation:

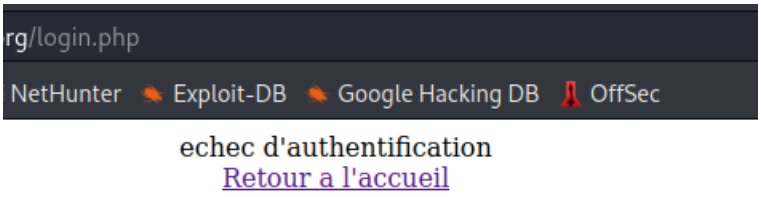
- Was able to view the credentials just surfing through the page view source.
- .Using the information from previous exploit exploiting weak passwords is easy and does not require advanced technical skills.



1. Enter a simple or common password viewed through



Asking to save the password



Password accepted and authenticated.

Remediation:

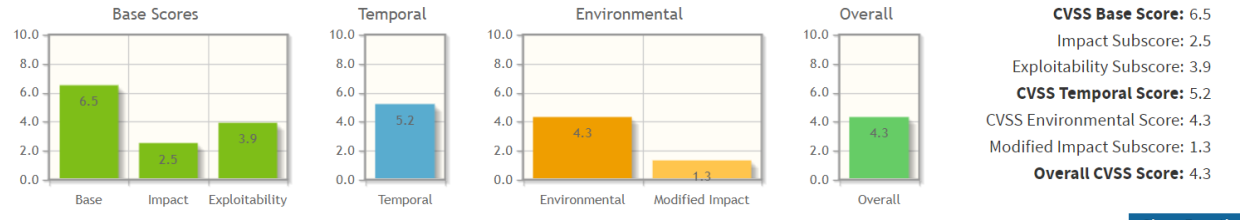
- Implement password policy with complex requirements, including minimum length, mixed character types, and avoidance of common passwords.
- providing multi-factor authentication to add an extra layer of security.
- Utilize password strength assessment tools during user registration or password change processes.
- Implement strategy and use the technique of monitoring and logging failed login attempts to detect potential brute-force attacks.
- Regularly update and patch to address any known security vulnerabilities.

3. Publicly accessible user enumeration feature:

The Common Weakness Enumeration (CWE) ID:

CWE-204: Observable Response Discrepancy

CVSS 3.1 Base Score Metrics:



AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:L/A:N/E:U/RL:O/RC:U/CR:L/IR:L/AR:X/MAV:N/MAC:L/MPR:N/MUI:N/MS:U/MC:L/MI:L/MA:N

Score : 4.3

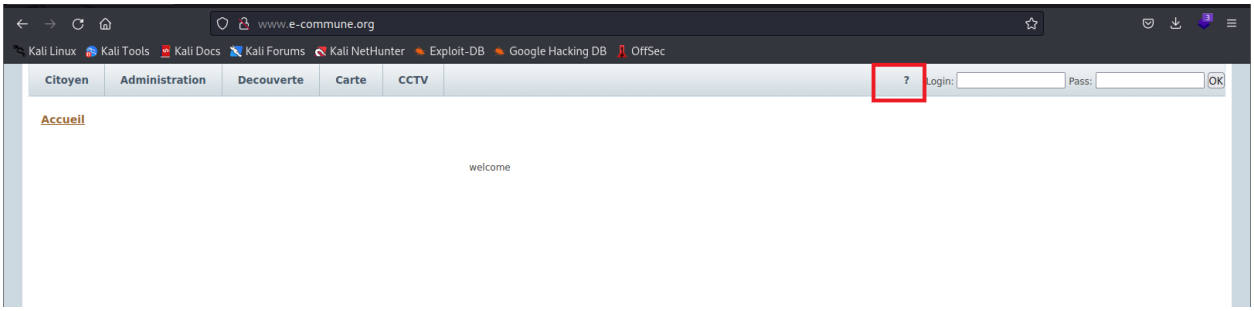
Risk : Low

Description:

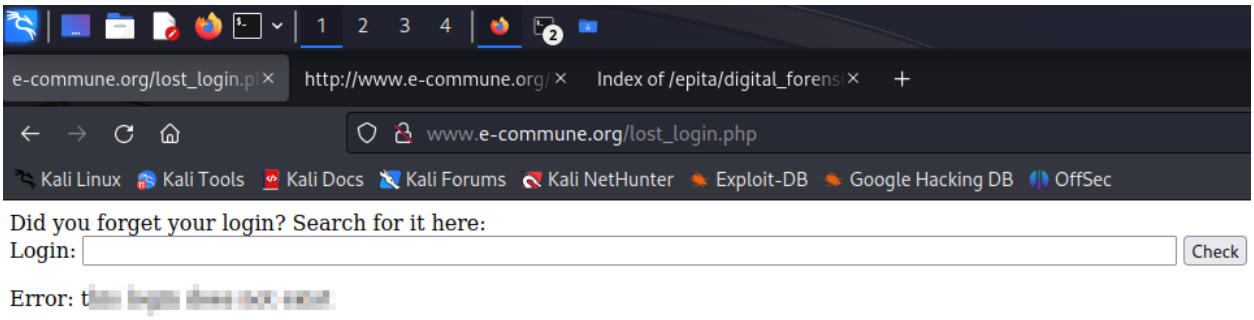
This feature allows an attacker to determine valid usernames or user passwords within a system. It can be exploited to gather information about users, potentially aiding in targeted attacks such as brute-forcing passwords or conducting social engineering attacks.

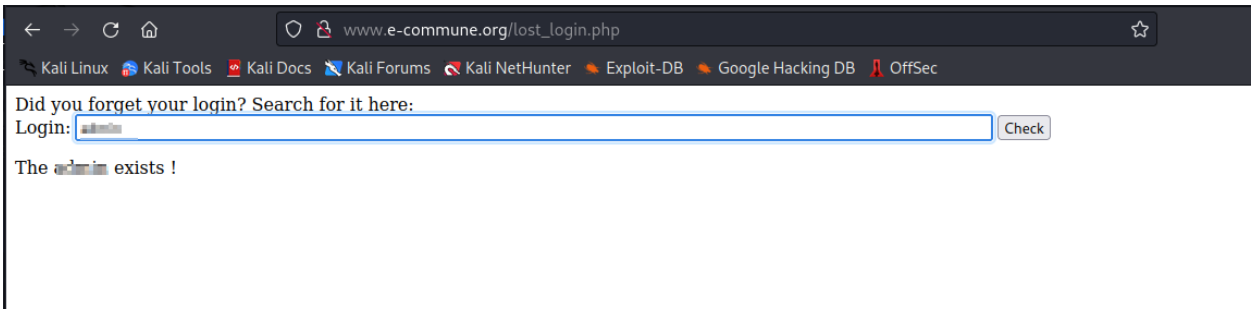
Exploitation:

- Attacker attempts to enumerate valid usernames through a publicly accessible feature.
- By analyzing system responses, they can determine the existence of valid user accounts.
- This information can be used to launch targeted attacks, such as password brute-forcing or social engineering.
- The attacker gains unauthorized access or gathers sensitive information by exploiting the enumerated user accounts.



By clicking on the “?” button in the home page will lead to lost login page then you can check if it exists or not





Remediation:

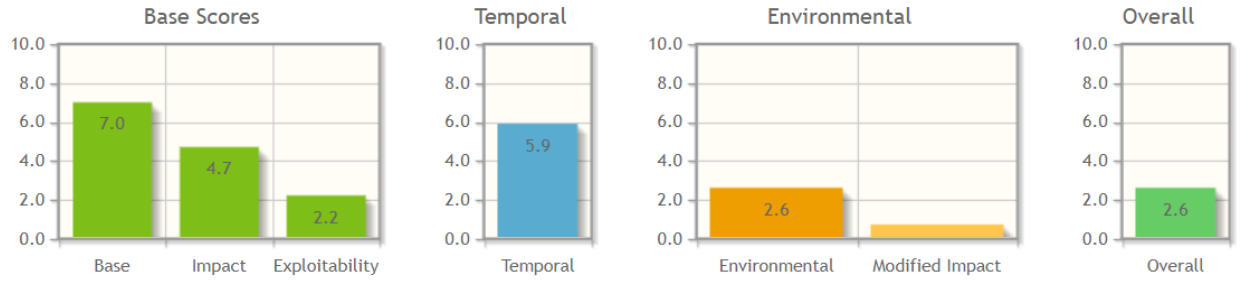
- Disable this feature completely or if you forget the password, you can provide the email if it is valid.
- Use emails to recover the username and do not tell the user that account exists.
- Monitor system logs for suspicious enumeration activities.
- Educate users on strong password practices and the importance of unique usernames.
- Enable multi-factor authentication.

4. Missing protection against brute-force attacks:

The Common Weakness Enumeration (CWE) ID:

CWE-307: Improper Restriction of Excessive Authentication Attempts

CVSS 3.1 Base Score Metrics:



CVSS 3.1 vector:

AV:N/AC:H/PR:N/UI:N/S:U/C:H/I:L/A:L/E:U/RL:U/RC:U/CR:L/IR:X/AR:X/MAV:N/MAC:H/MPR:N/MUI:N/MS:U/MC:L/MI:N/MA:N

Score: 2.6

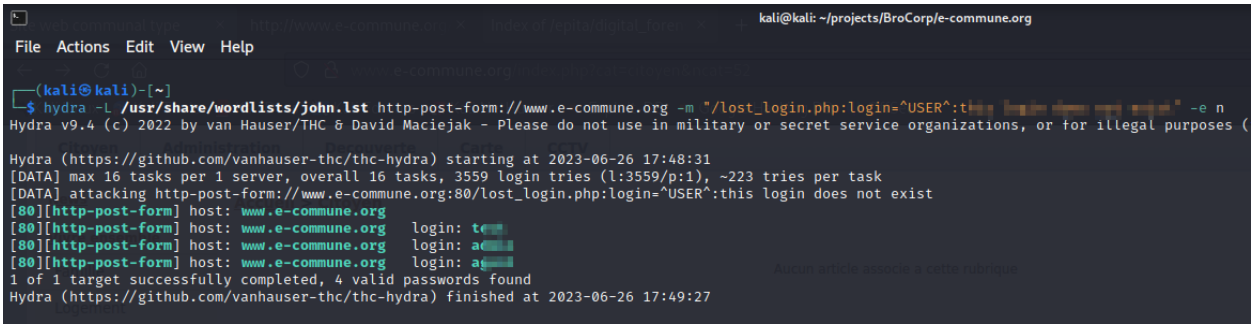
Risk: Low

Description :

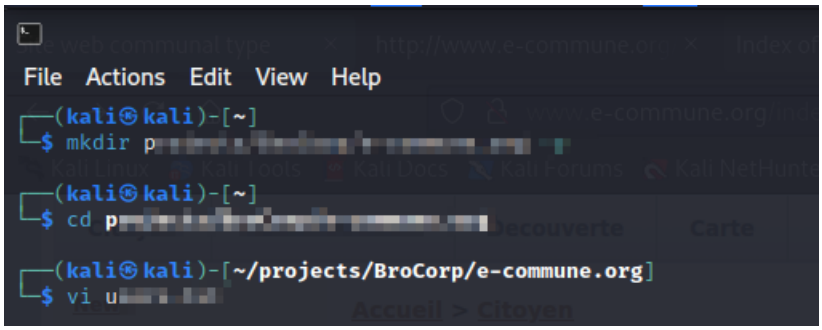
The audit indicates that e-commune.org does not have sufficient measures to identify and prevent the multiple authentication attempts to the web server and lacking configuration for implementing short time frame to prevent brute force-based attacks. The usage of applications like hydra, burp suite and gobuster etc., in the audit environment if the systems does not have proper configurations to control these techniques and does not follow the same parameters to prevent brute force attack there will be higher range of threats of attacks by using this methodology.

Exploitation:

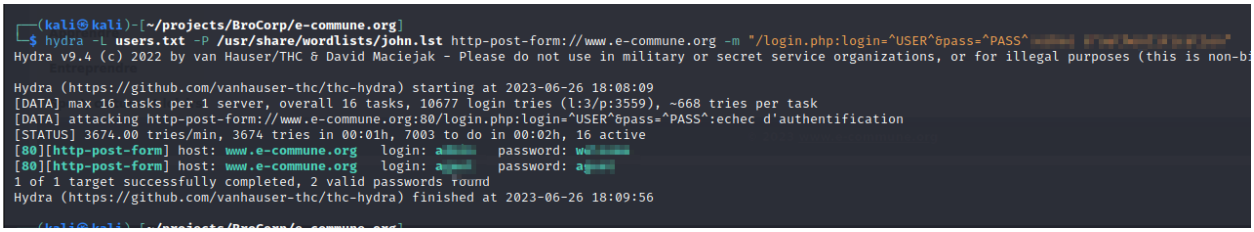
- It has been observed that getting the error message in the “?” page we can use it to find the username
- By using the “hydra” command with the list file (john.lst) and the message shown in the page we can find the usernames
- Concluding by saving the usernames in a text file.
- Then we use “hydra” command with the user text file and the message shown when the username or password is incorrect to brute-force the password for each username
- then use the username and password to login.



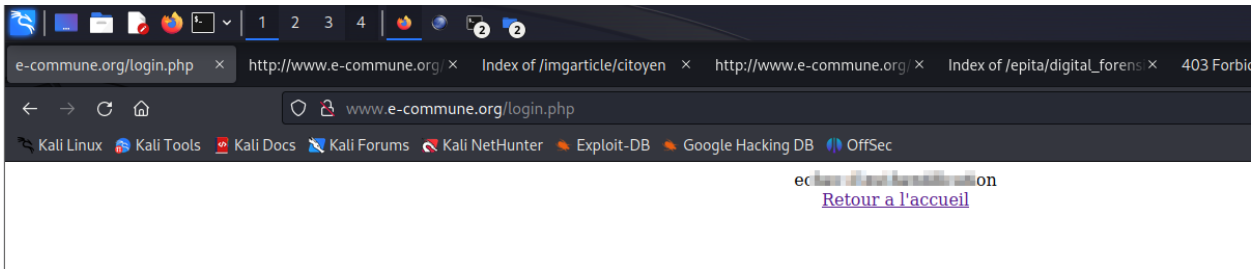
Using hydra command to find the username.



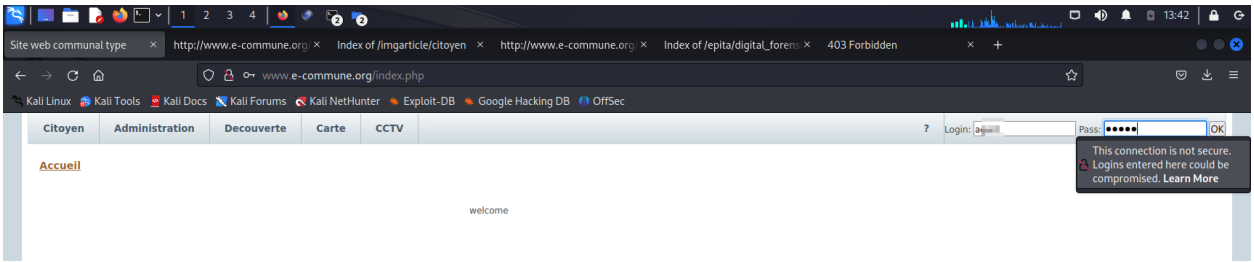
Creating the username text file



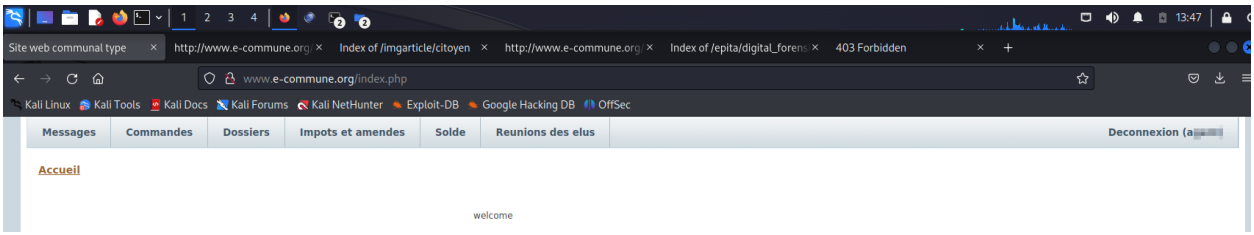
Using hydra to find the password of the username in the text file.



Message shown when username or password is incorrect.



Then use the username and password to login.



Remediation:

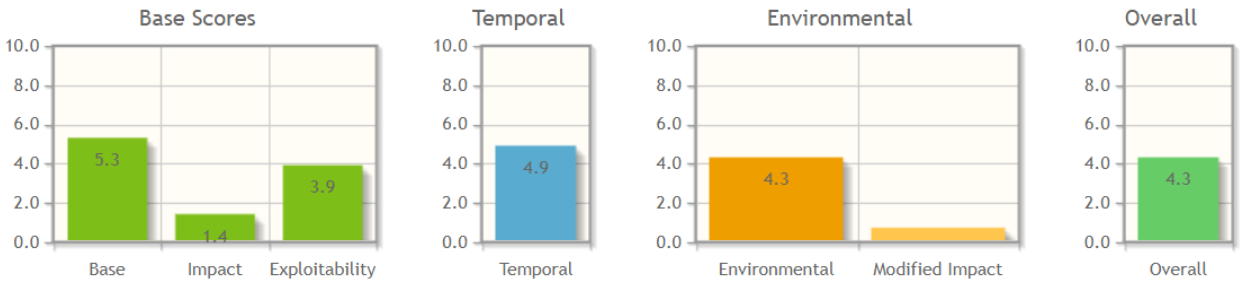
- Implement account lockouts.
- Enforce strong password policies.
- Implement rate limiting.
- Use MFA "Multi-Factor Authentication".
- Monitor and log login attempts.
- Deploy IDS/IPS "Intrusion Detection System" and "Intrusion Prevention System".

5. Technical information disclosure:

The Common Weakness Enumeration (CWE) ID:

CWE-497: Exposure of Sensitive System Information to an Unauthorized Control Sphere

CVSS 3.1 Base Score Metrics:



CVSS 3.1 vector:

AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N/E:H/RL:U/RC:U/CR:L/IR:X/AR:X/MAV:N/MAC:L/MPR:N/MUI:N/MS:U/MC:L/MI:N/MA:N

score : 4.3

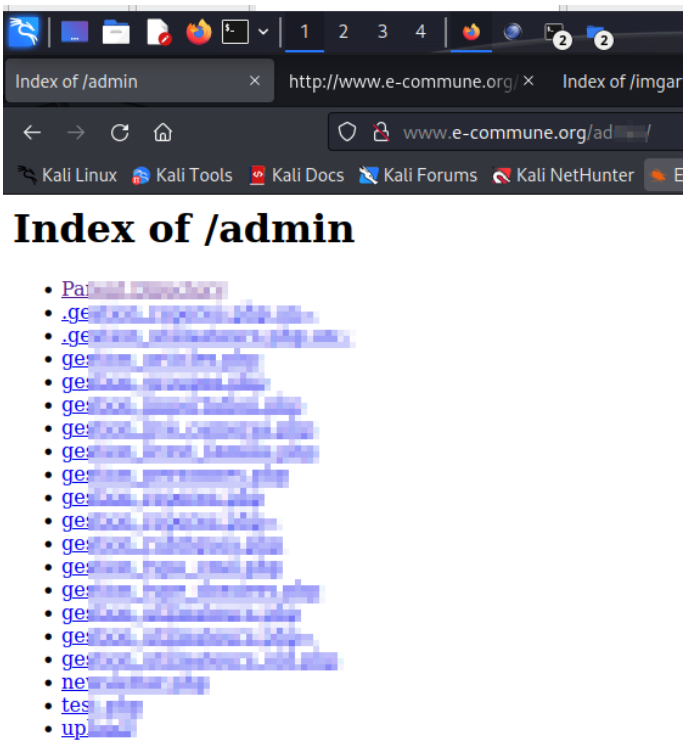
Risk : Medium

Description:

This vulnerability allows people to get into important information without permission. The seriousness of this can change depending on the situation and what kind of information is shared. Sensitive information refers to private and important details like personal information, the state of a system, confidential business information, the setup of a network, and so on. Different groups, like people who use the information, people who manage it, and people who create it, may all have different ideas about how to keep it safe. Information exposure can occur when mistakes, whether intentional or unintentional, allow unintended access to sensitive information.

Exploitation:

- By trying to fill the path example (admin) the website will open and show the information or files for this admin



Remediation:

- Not giving access to normal user and disable the access while synchronizing the file and folder to the server.

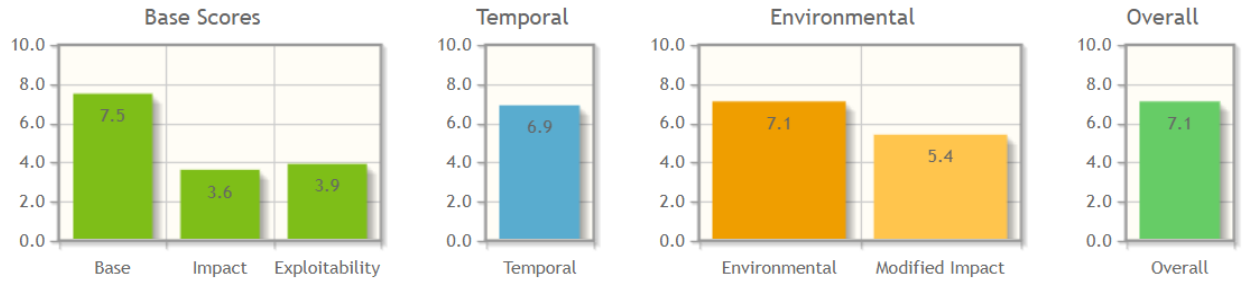
6. Open directory listing:

The Common Weakness Enumeration (CWE) ID:

CWE-552: Files or Directories Accessible to External Parties

CWE-23: Relative Path Traversal

CVSS 3.1 Base Score Metrics:



CVSS 3.1 vector:

AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:N/A:N/E:H/RL:U/RC:U/CR:H/IR:X/AR:X/MAV:N/MAC:H/MPR:N/MUI:N/MS:U/MC:H/MI:N/MA:N

Score: 7.1

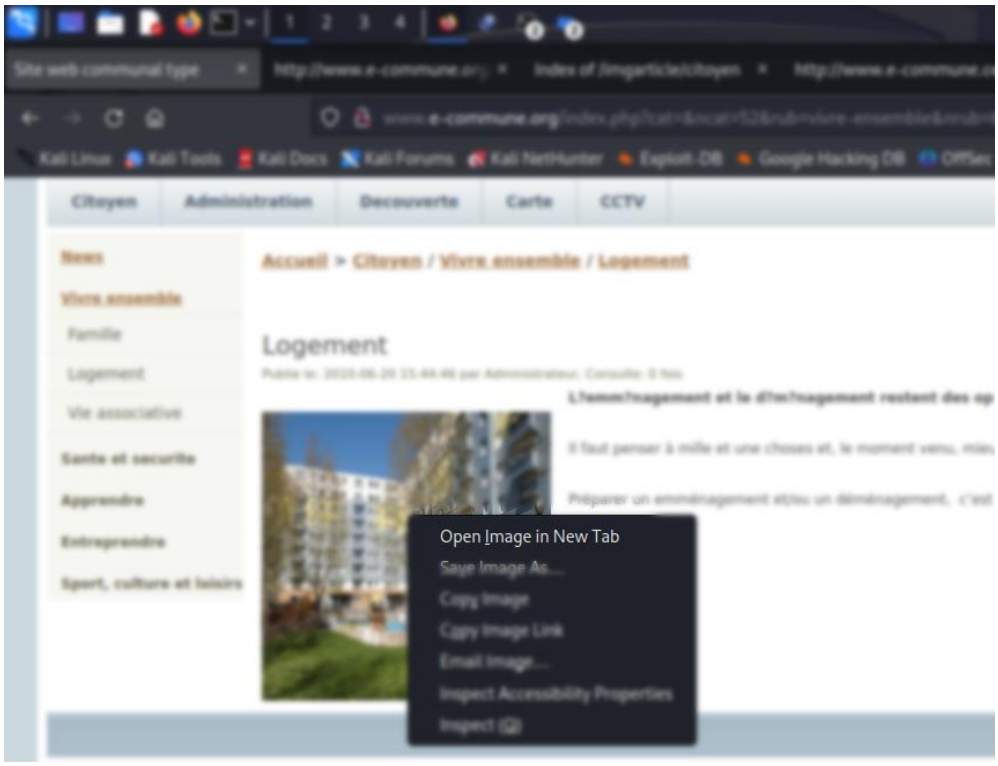
Risk: High

Description:

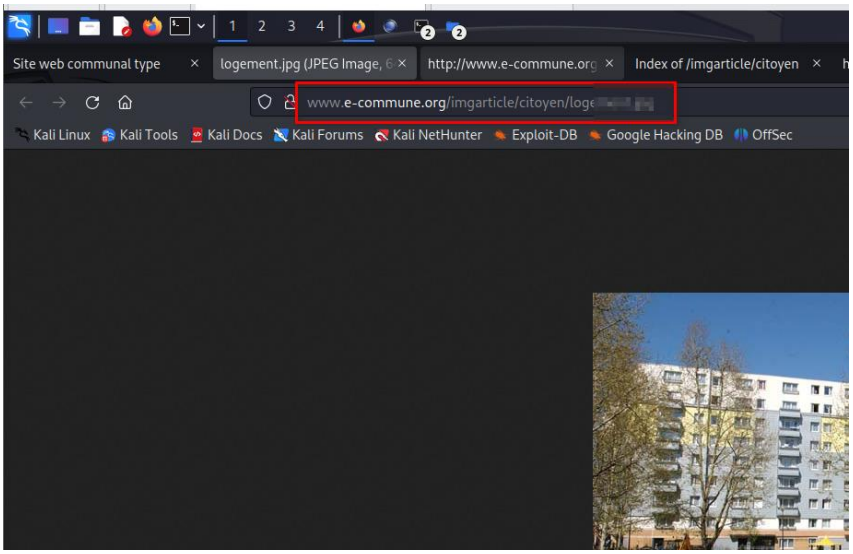
This vulnerability lets people who should not be able to access certain files and folders access them. It can happen on web servers, FTP servers, and other similar systems when important files are not protected with the right access restrictions. It can also happen in cloud technologies and containers if storage accounts are set up incorrectly, allowing anyone to access or change data without permission.

Exploitation:

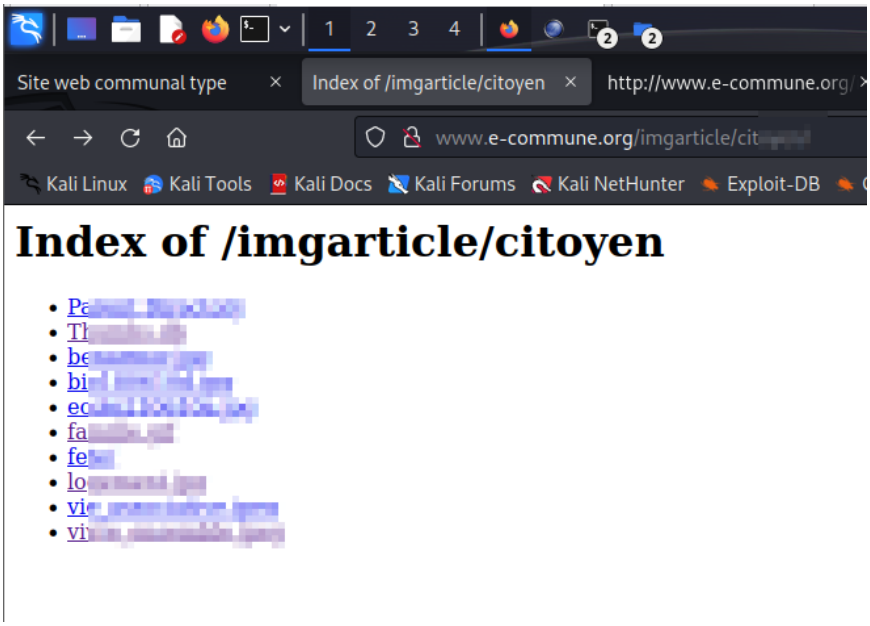
- By using any picture opening it in new tab
- Then delete the name of the picture in the path.
- Then we can access the files saved in the parent path of the image



Opening any picture in new tab



Targeting the link of this picture



Accessing the data in this path

Remediation:

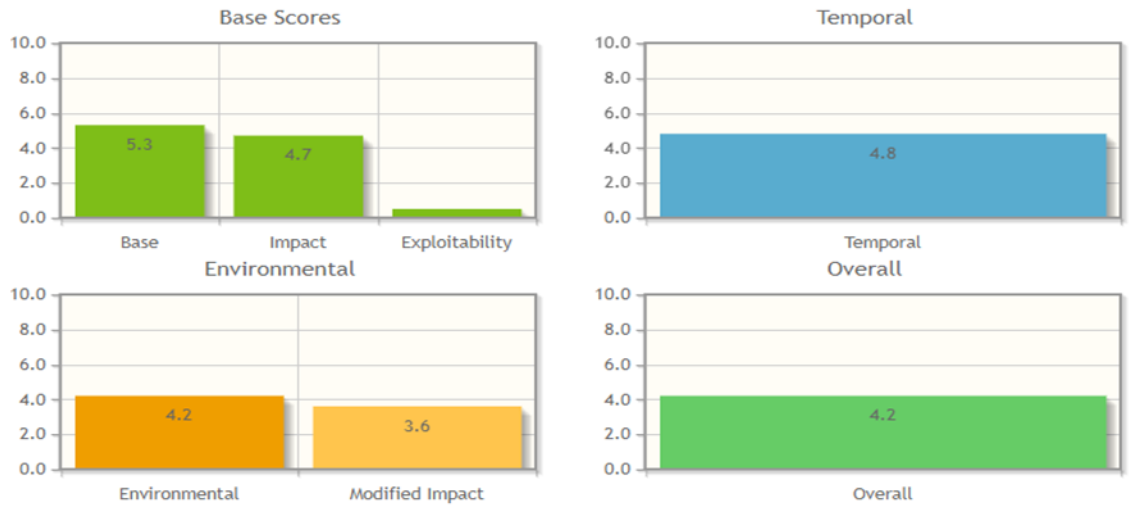
- Disable public access while saving the file and folders in the server.
- Giving access just to the files that should be shown for the user.

7. Sensitive / private files publicly accessible:

The Common Weakness Enumeration (CWE) ID:

CWE-200: Exposure of Sensitive Information to an Unauthorized Actor

CVSS 3.1 Base Score Metrics:



CVSS 3.1 vector :

AV:N/AC:H/PR:H/UI:R/S:U/C:H/I:L/A:L/E:P/RL:U/RC:R/CR:L/IR:M/AR:M/MAV:N/MAC:L/MPR:H/MUI:R/MS:U/MC:H/MI:L/MA:L

Score : 4.3

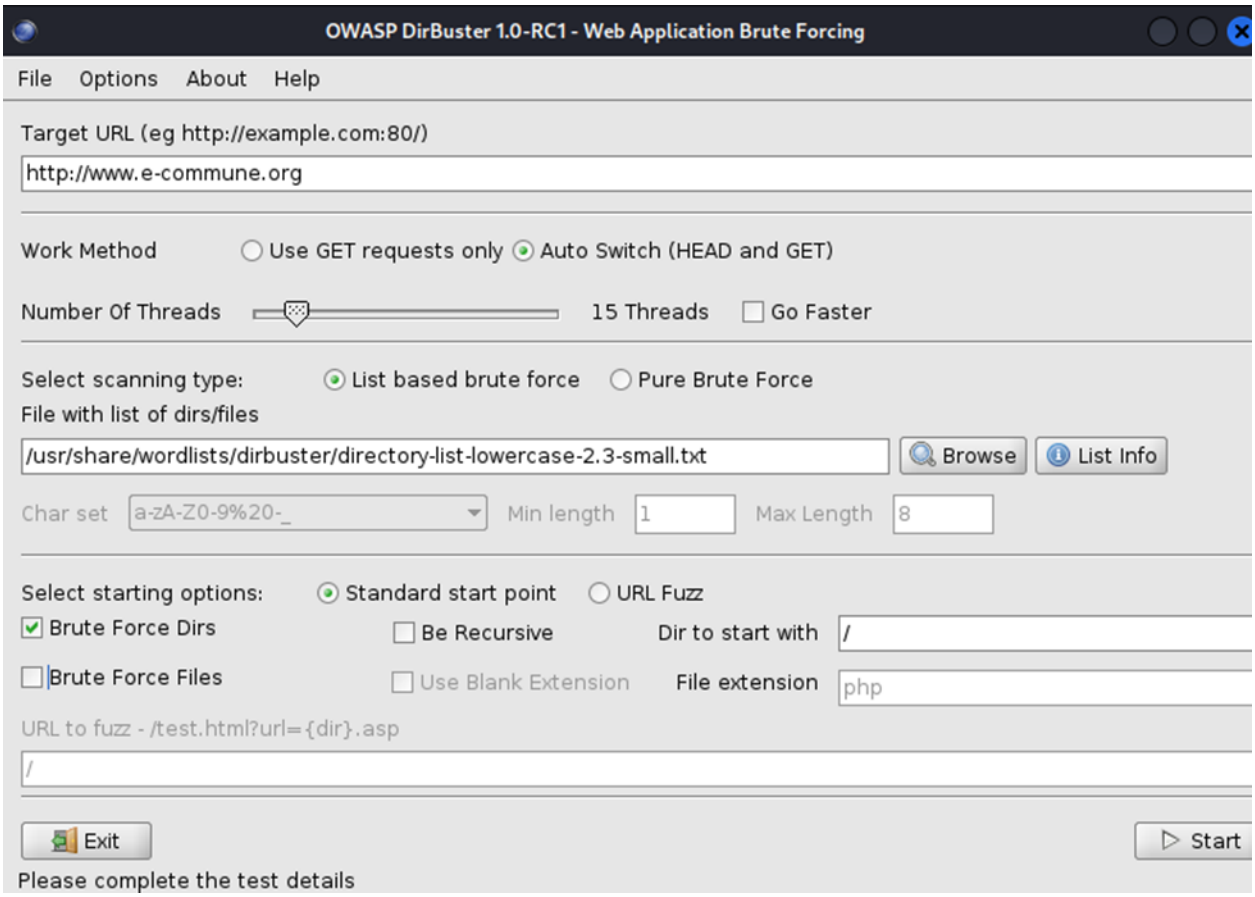
Risk : Medium

Description:

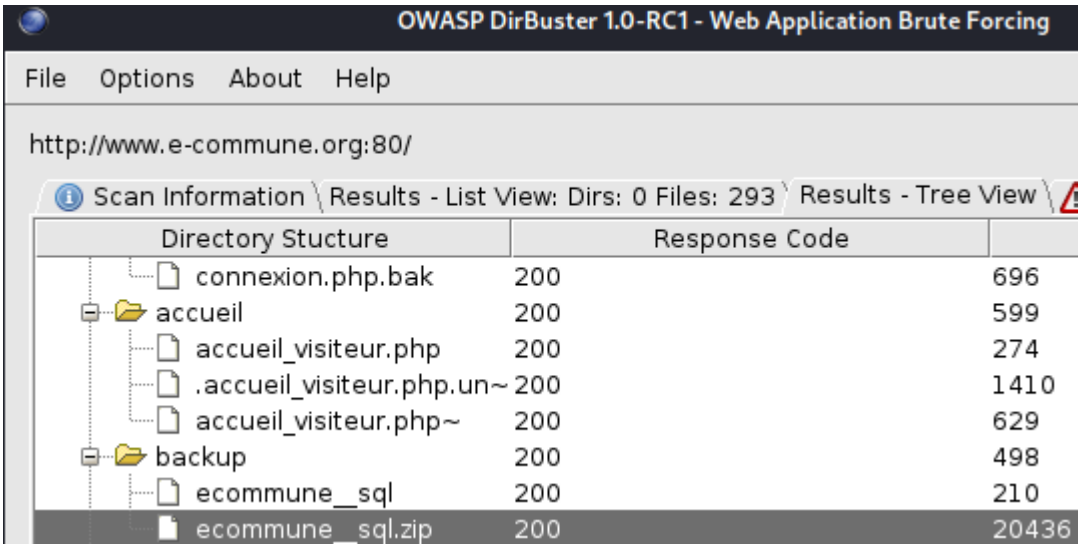
This type of vulnerability in the web application system allows the threat vector to gain access for the sensible directories, in this case the e-commune site allows the attacker to gain access to the 'back up file' using dirtbuster. The Risk of this outcome of the attack is gaining access to the sensitive files in the cloud server etc. through the entire audit and process it possible to gain the access to the SQL zip file and gaining access to the password of the credentials in the file.

Exploitation:

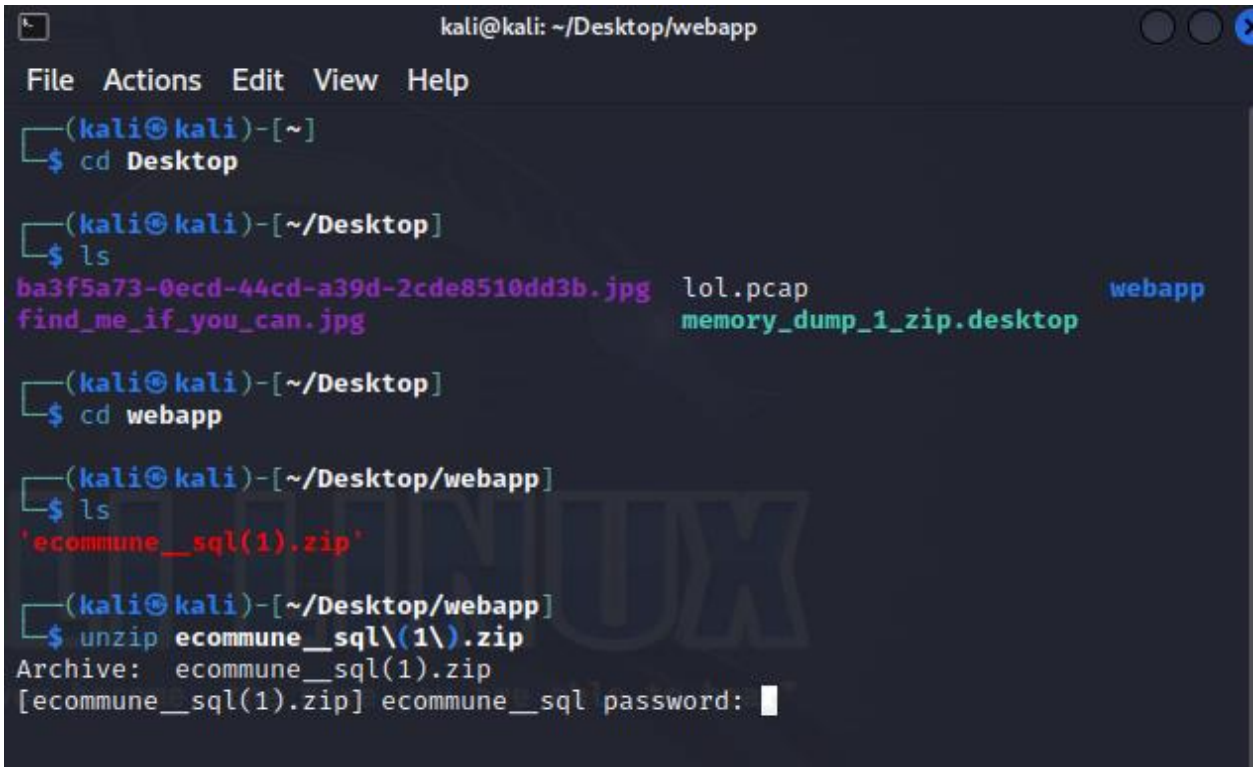
Steps:



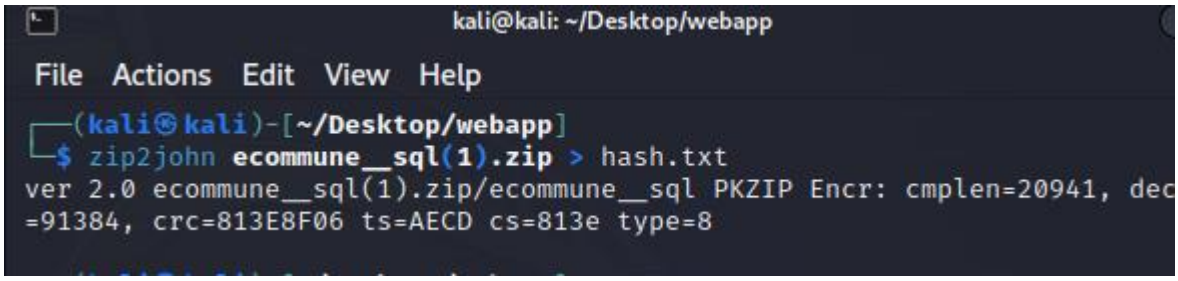
1. Using the dirbuser with inserting necessary information of the site , threads number and providing the brute force dirs. To directories will run the possible outputs of the accessible directories



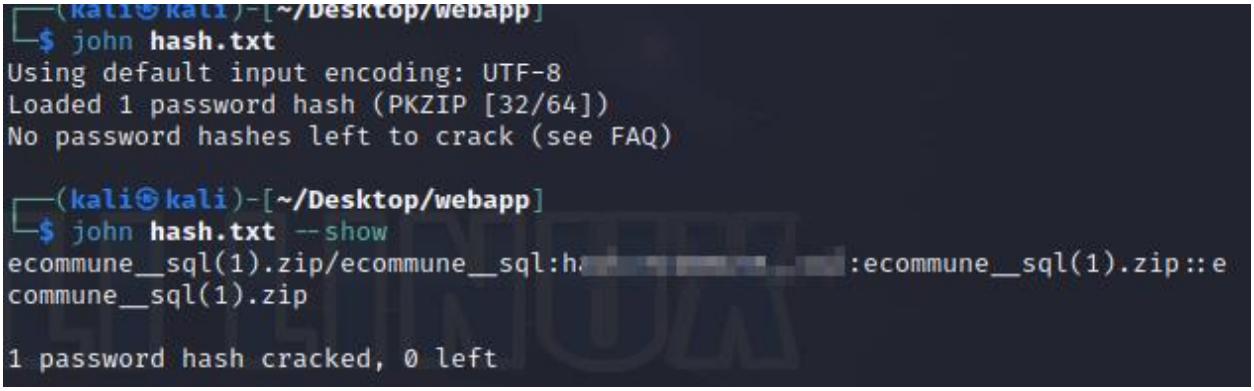
2. The results of the process paved way to find the interesting file names backup and which contains e-commune_sql file and by clicking it automated the downloading of the file to the system.



3. requirement of the password built the direction to crack the hashes.



4. sed zip2john to provide the hash list to created text file

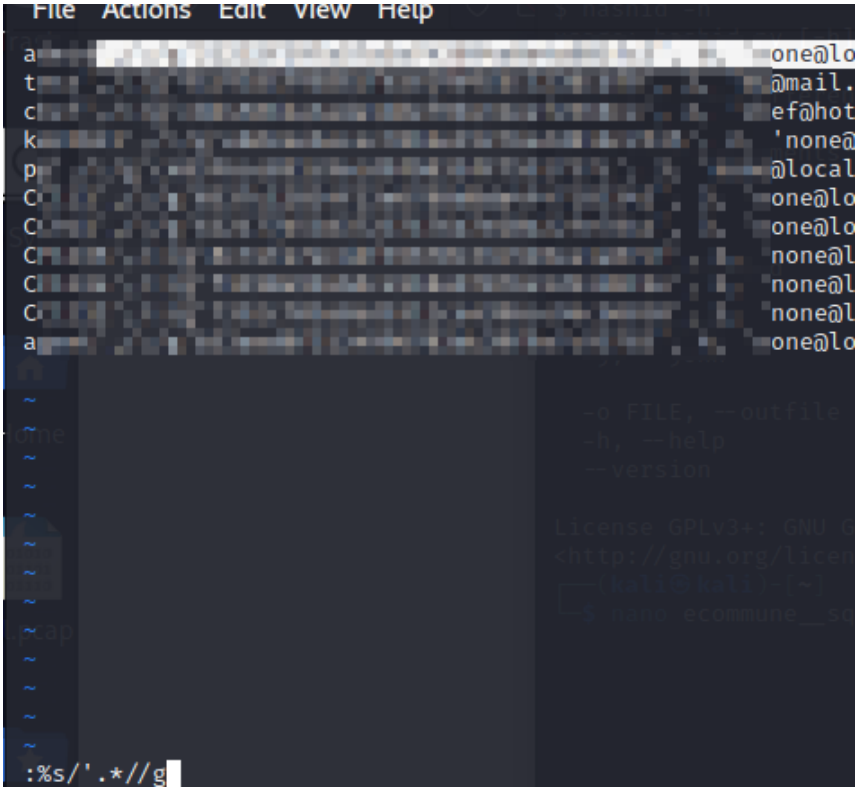


5. Using the “john” and guessed hash text file the application provided the cracked hash

```
1 password hash cracked, 0 left

┌──(kali㉿kali)-[~/Desktop/webapp]
└─$ unzip -P hack ecommune__sql(1).zip
Archive:  ecommune__sql(1).zip
  inflating: ec
```

6. using the cracked hash, the ecommune_sql zip file was able to unzip the content



7. using the editor created new file by copying the credentials list form unzipped file to new edited text

```
┌──(kali㉿kali)-[~/Desktop/webapp]
└─$ john --format=Raw-MD5 ehash.txt
Using default input encoding: UTF-8
Loaded 11 password hashes with no different salts (Raw-MD5 [MD5 128/128 AVX 4×3])
Warning: no OpenMP support for this hash type, consider --fork=4
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
ag               (a
pp               (p
ke               (k
ch               (c
ad               (a
Almost done: Processing the remaining buffered candidate passwords, if any.
Proceeding with wordlist:/usr/share/john/password.lst
Proceeding with incremental:ASCII
a                (te
```

8. exploited the credentials using ‘john’ with the available information of the unzipped file

Remediation:

- implement access control of providers to restrict public access of the directories.
- since the usage of dirbuster to brute force to web application directories the rate limiting. protocol should be applied for the IP address whilst the brute force indication.
- Increasing security on the web application by encrypting the request access from various network address will slow down the process of brute force.

Reference

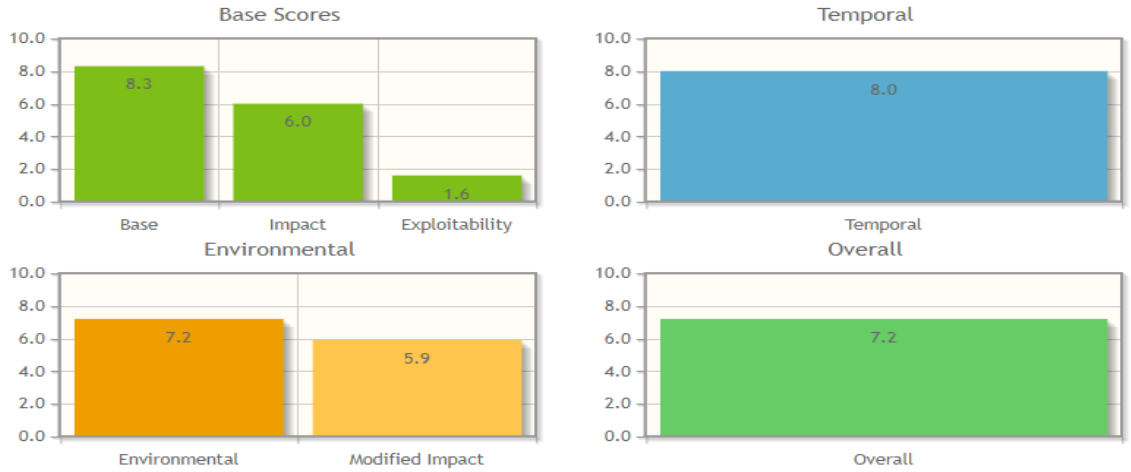
- https://www.openwall.com/john/ - the program is referenced and written for the purpose of capturing hash, network traffic captures and password cracking etc.
- Dirbuster – a fork project of the original application site and its effective in this testing purposes

8. Carte has directly link for the shell:

The Common Weakness Enumeration (CWE) ID:

CWE-553: Command Shell in Externally Accessible Directory

CVSS 3.1 Base Score Metrics:



CVSS 3.1 vector:

AV:N/AC:H/PR:N/UI:R/S:C/C:H/I:H/A:H/E:H/RL:U/RC:R/CR:L/IR:H/AR:H/MAV:N/MAC:H/MPR:N/MUI:R/MS:U/MC:N/MI:H/MA:H

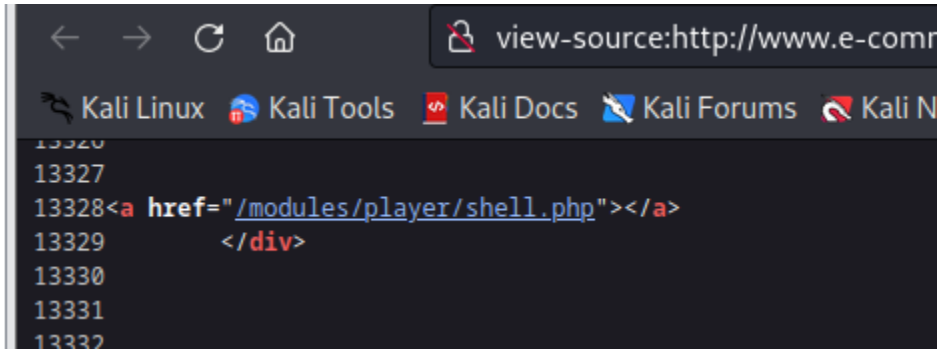
Score :7.2

Risk: High

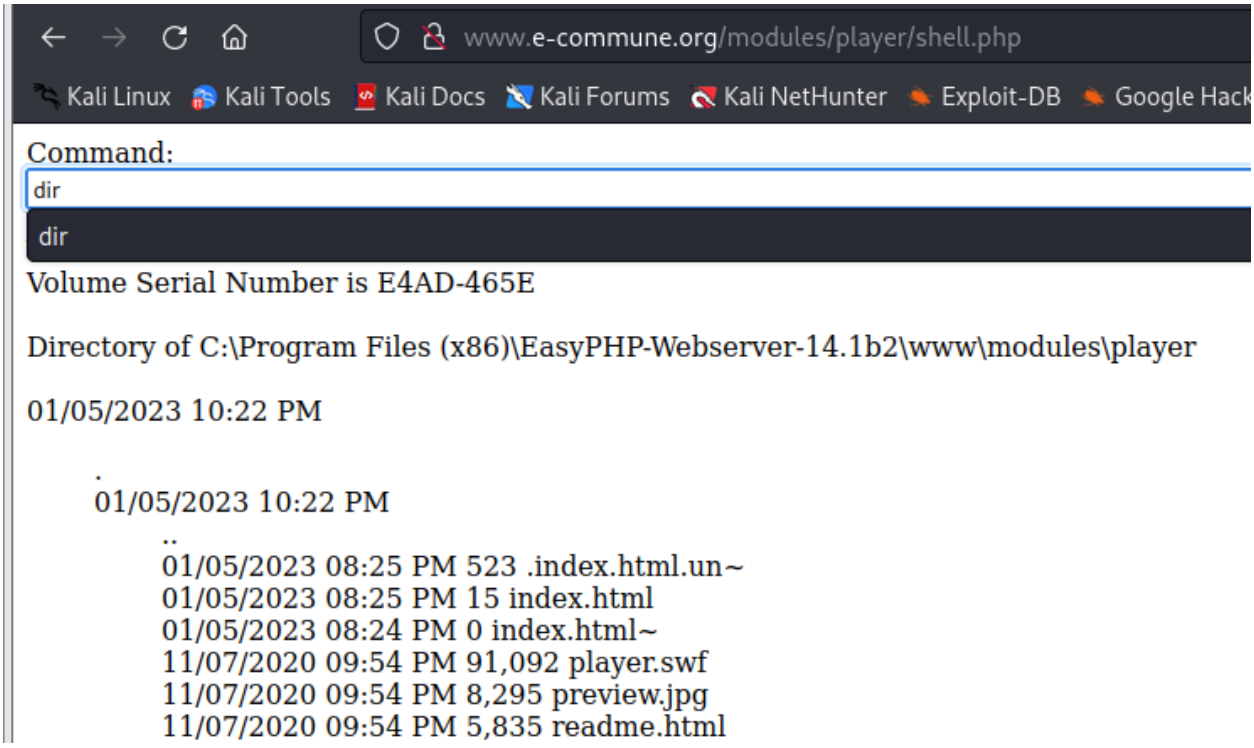
Description:

During the observation on the e-commune.org, ‘carte’ page, a finding of shell directory exists within the web server. The risk is high since threat actors can access the shell and manipulate to identify, steal, and modify data by executing shell commands. This is q good indication of the web server either primarily being compromised and the attacker built a backdoor or the system administrator left this for possible reason which is known and its dangerous.

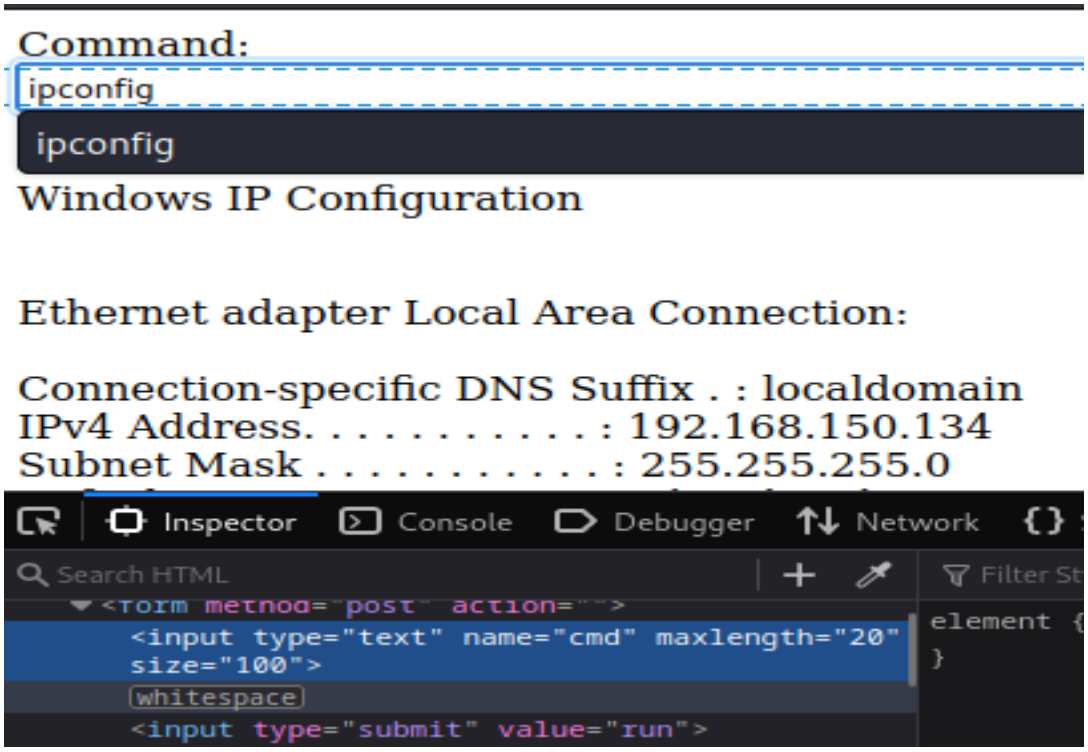
Exploitation:



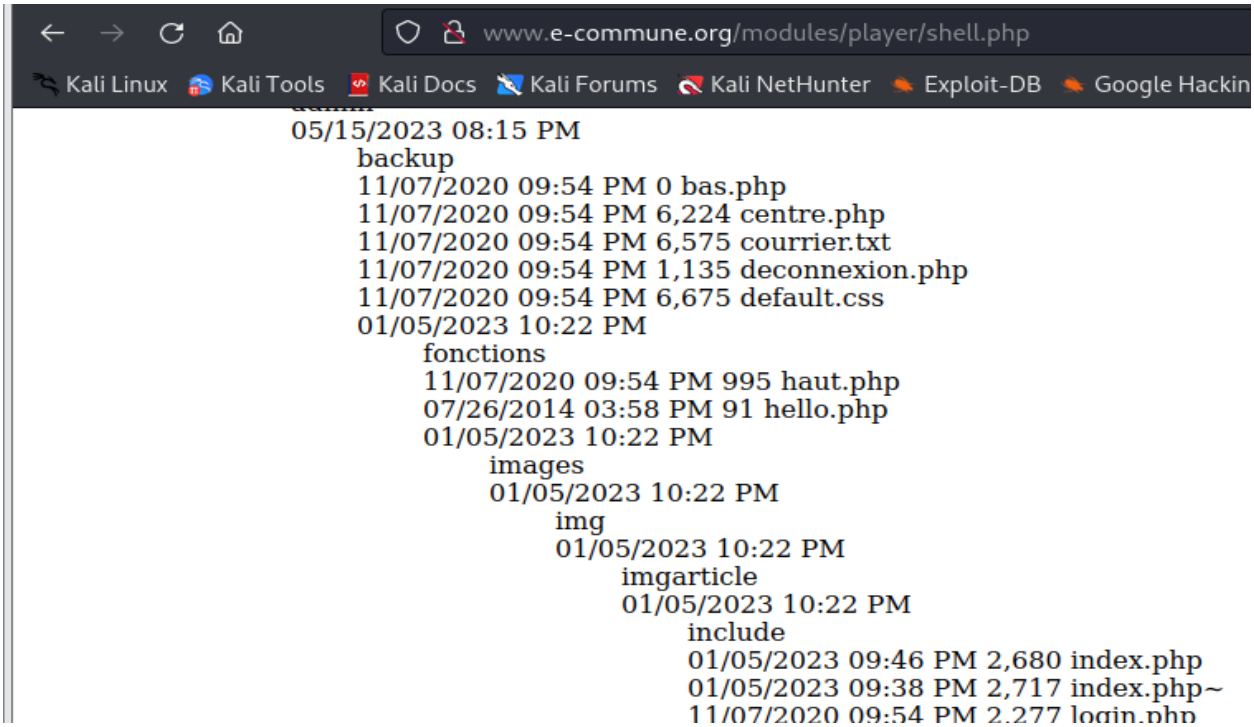
1. Upon viewing the source of the page, was able to navigate through the bottom of the page and identify the possible link leading to the shell page.



2. Initial the page is 3 limit character exploitable command and from the previous finding of the directory listing, this known as a windows system and “dir” command list out pages includes ‘~’ files with meta data.



3. By editing the character length increased the risk of exploiting to find more information by executing shell commands “network details.”



4. Executing the command “dir ../../ “ exposed the important information of backup files

Remediation:

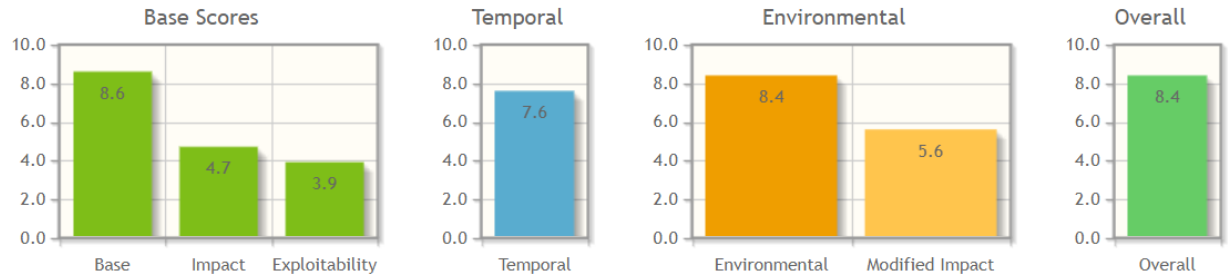
1. Removing the shell direct access.
2. Taking control of the web link directory in the web source code will prevent access to the shell page.
3. Removing the shell accessible to any linking directories to important folders.

9. CCTV has password in plaintext:

The Common Weakness Enumeration (CWE) ID:

CWE-549: Missing Password Field Masking

CVSS 3.1 Base Score Metrics:



CVSS 3.1 vector:

AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:L/A:L/E:H/RL:O/RC:U/CR:H/IR:L/AR:L/MAV:N/MAC:L/MPR:N/MUI:N/MS:U/MC:H/MI:L/MA:L

Score: 8.4

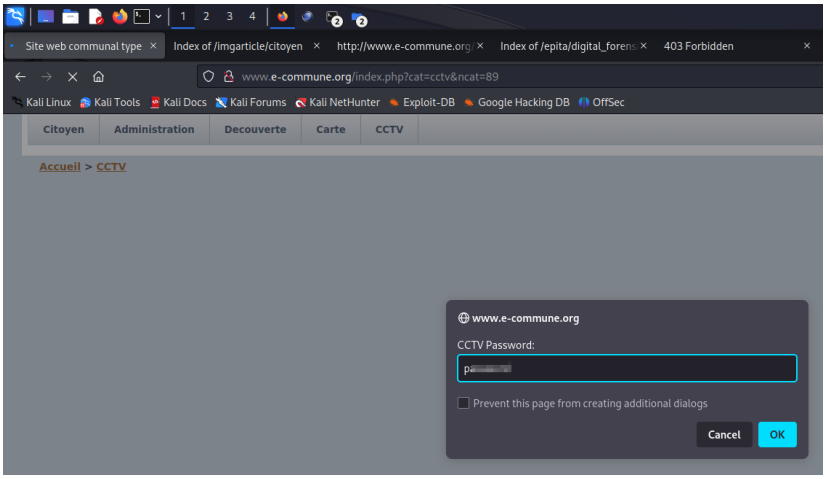
Risk: High

Description:

This vulnerability refers to a security issue where the passwords are stored or transmitted without encryption, leaving them easily readable to anyone who has access to the system. This vulnerability puts the CCTV system at risk of unauthorized access and compromises the security of the surveillance footage and the overall system.

Exploitation:

- By going to the “CCTV” page it asks for password
- By typing the password it is shown as plaintext



Plaintext Password shown.

Remediation:

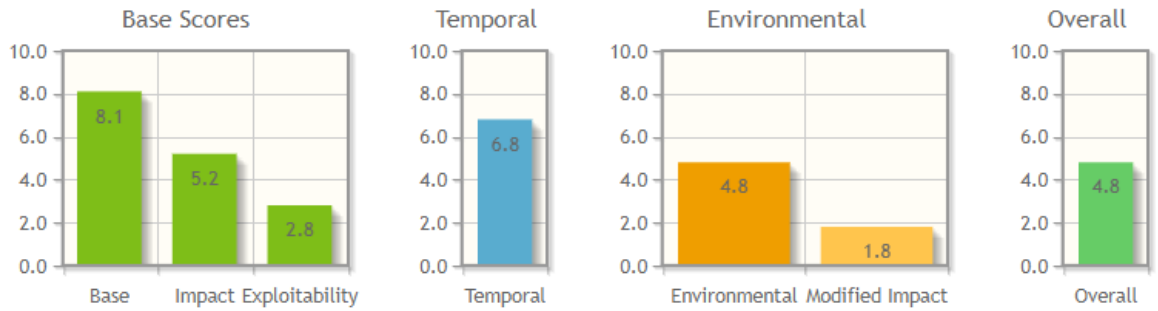
- Restrict access to configuration interfaces: Limit access to the CCTV system's configuration interfaces to authorized personnel only.
- Changing the type of the textfield to password instead of text.
- Encrypt passwords: Store passwords in an encrypted format to prevent unauthorized access.
- Use strong passwords: Enforce the use of strong, unique passwords for CCTV systems.

10. CCTV has client-side authentication:

The Common Weakness Enumeration (CWE) ID:

CWE-603: Use of Client-Side Authentication

CVSS 3.1 Base Score Metrics:



CVSS 3.1 vector:

AV:N/AC:L/PR:N/UI:R/S:U/C:H/I:H/A:N/E:U/RL:U/RC:U/CR:L/IR:L/AR:L/MAV:N/MAC:L/MPR:N/MUI:N/MS:U/MC:H/MI:N/MA:N

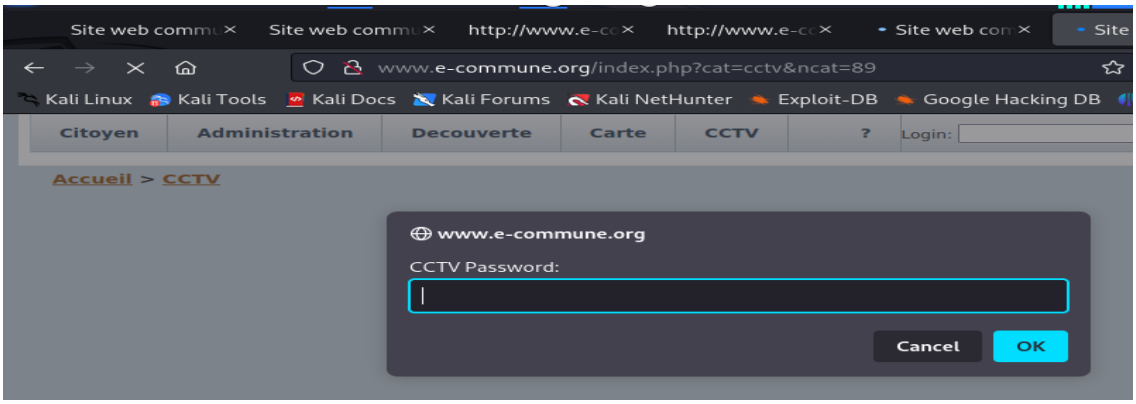
Score: 4.8

Risk: Medium

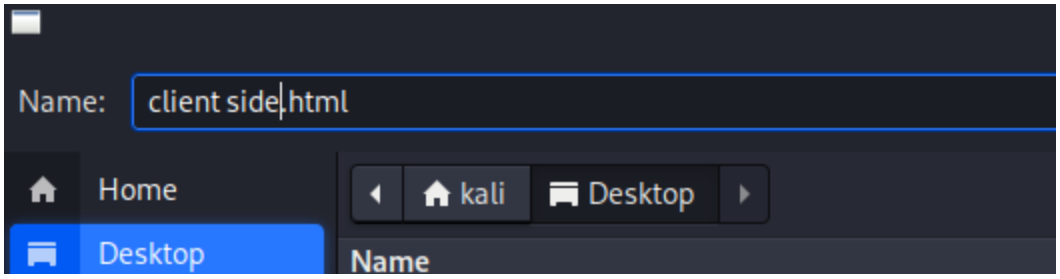
Description :

During to the audit through the cctv page the testing of client-side authentication was success, and this vulnerability will allow the attacker to modify data “e.g. – modifying the html page and bypassing the restricted page”, The client-side authentication is a weak mechanism to a website since it allows to read the source code and reverse engineer the code and exploiting to the site pages.

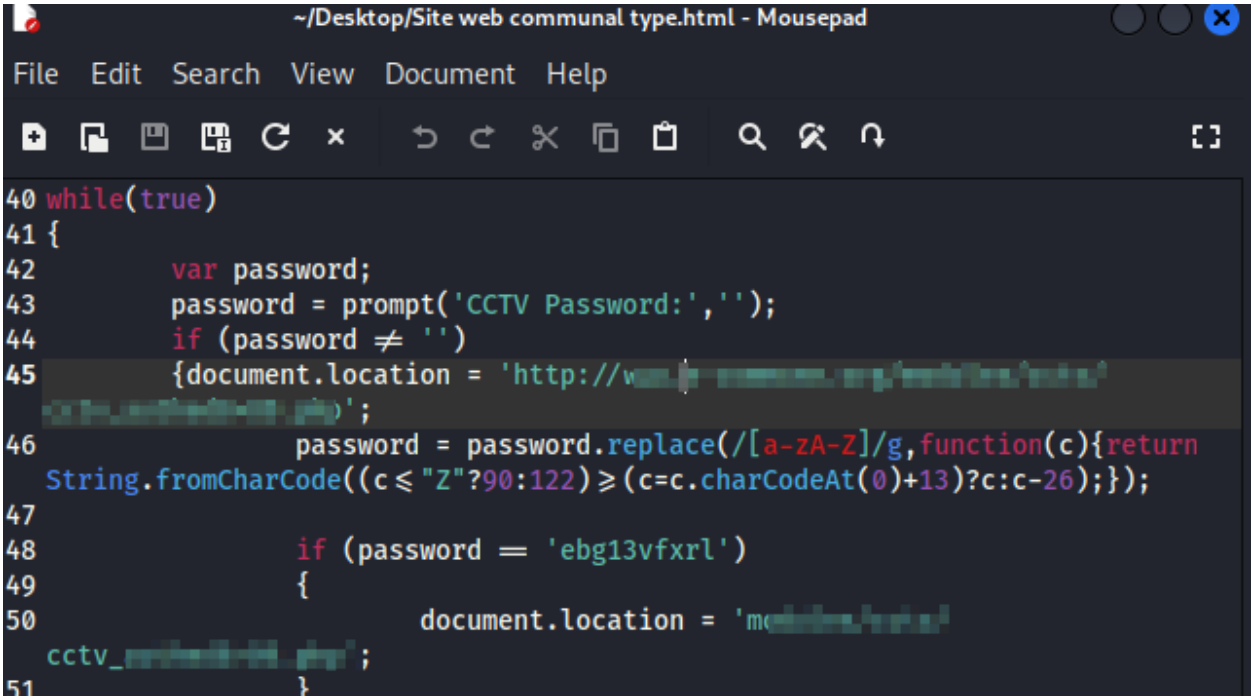
Exploitation:



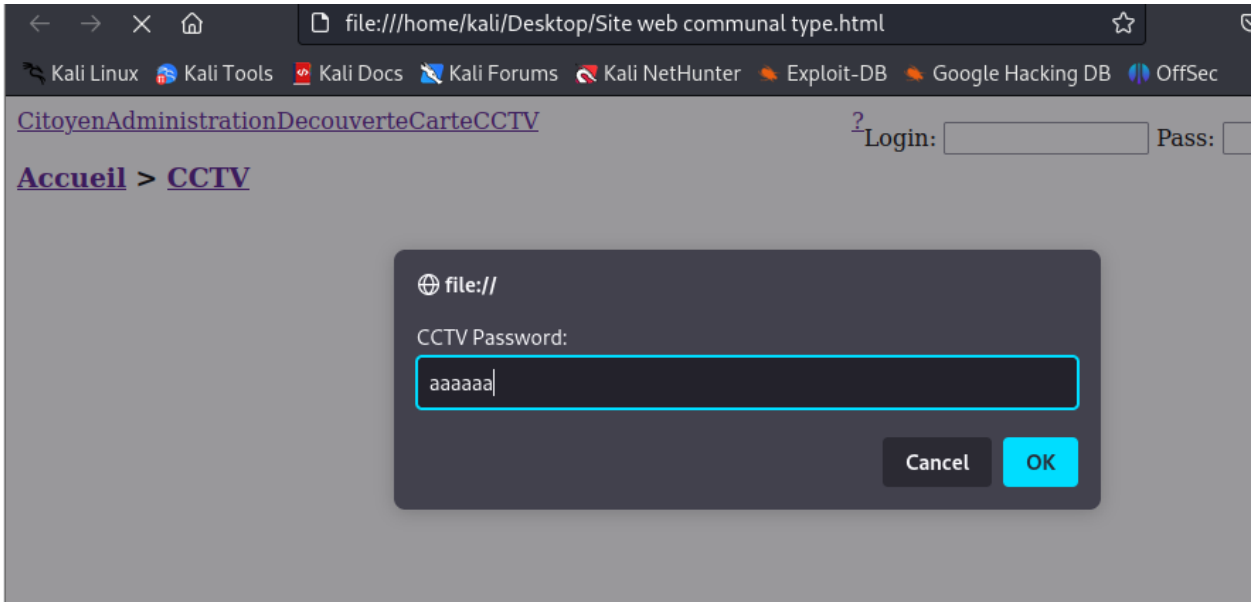
1. The first access to the page and the initial founding of the plaintext password on the cctv



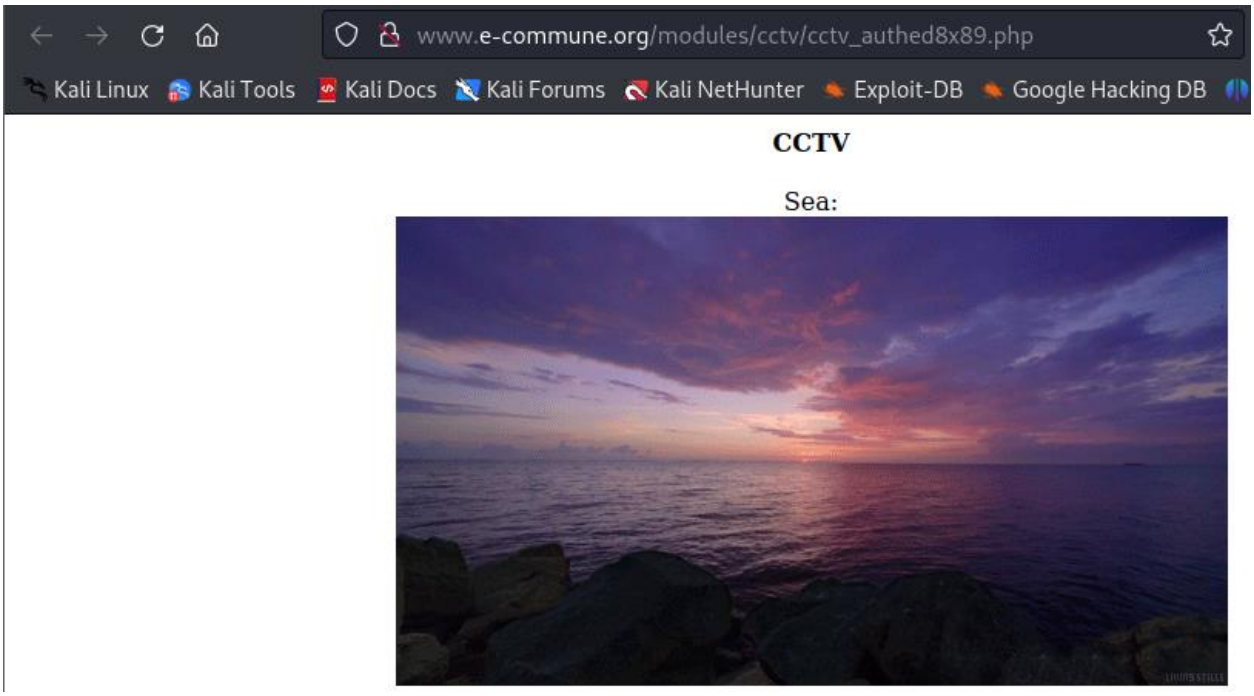
2. saving the web page for modification



3. changing the script and modifying it for auto execution page from locally



4. after the modification, the password restriction authentication removes and provides the flexibility to type any text



5. The access granted to the CCTV content page.

Remediation:

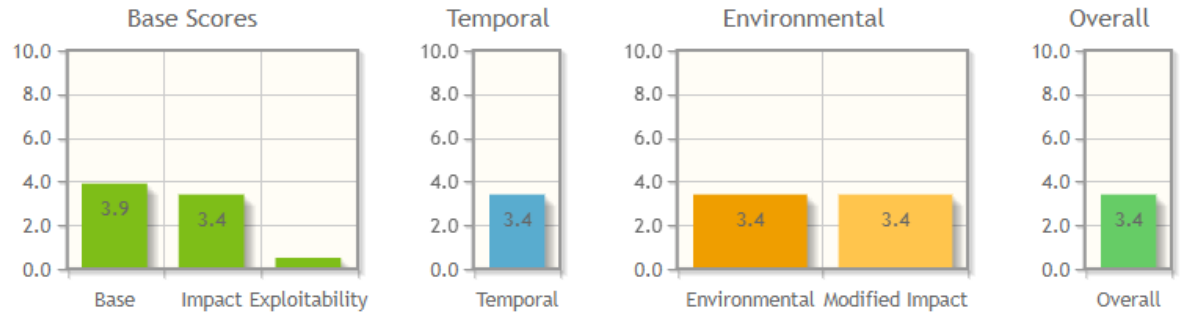
1. Remove access to the client-side authentication and restrict to the only server-side authentication.
2. Remove scripts sitting on page source code.
3. Implement CSP by defining the restriction of certain action on the page – e.g. saving content of the page locally

11. XSS possible on the Mail sending and receiving:

The Common Weakness Enumeration (CWE) ID:

CWE-79: Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting')

CVSS 3.1 Base Score Metrics:



CVSS 3.1 vector:

AV:N/AC:H/PR:H/UI:R/S:U/C:L/I:L/A:L/E:F/RL:W/RC:U/CR:X/IR:X/AR:X/MAV:N/MAC:H/MPR:H/MUI:R/MS:U/MC:L/MI:L/MA:L

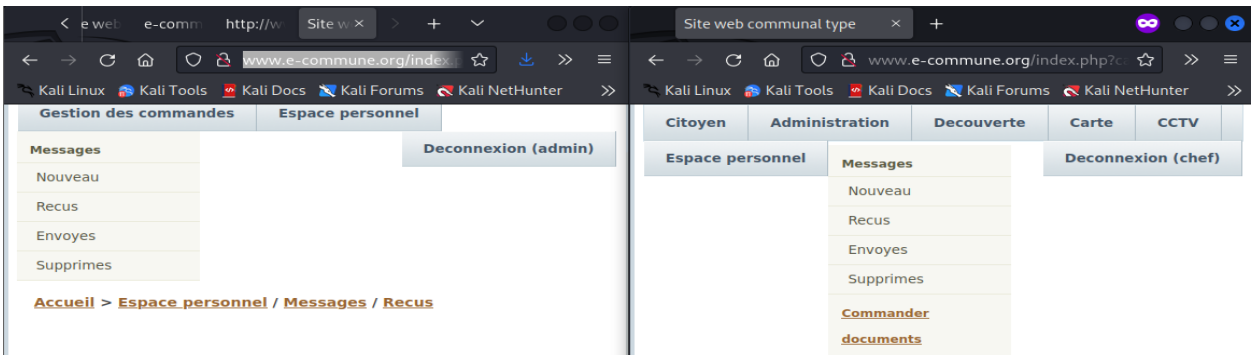
Score: 3.4

Risk: Low

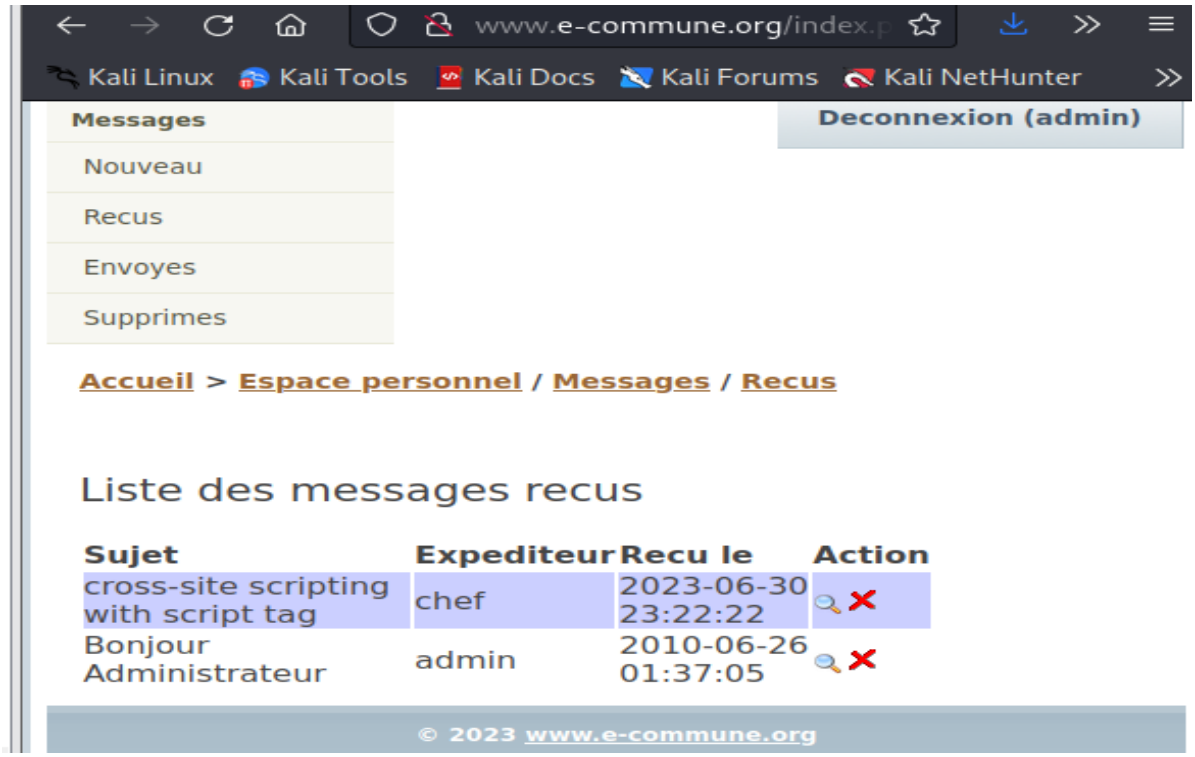
Description:

Cross-site scripting is possible after clear observation and successful testing to the e-commune site. Cross site scripting targets the sites when there is untrusted data enters to web site, lacking prevention of content containing java script and html tags by the web browsers. The technique is complicated but if the attacker executes cross site scripting successfully it will affect all the privileged and user access and loss of important data occurs.

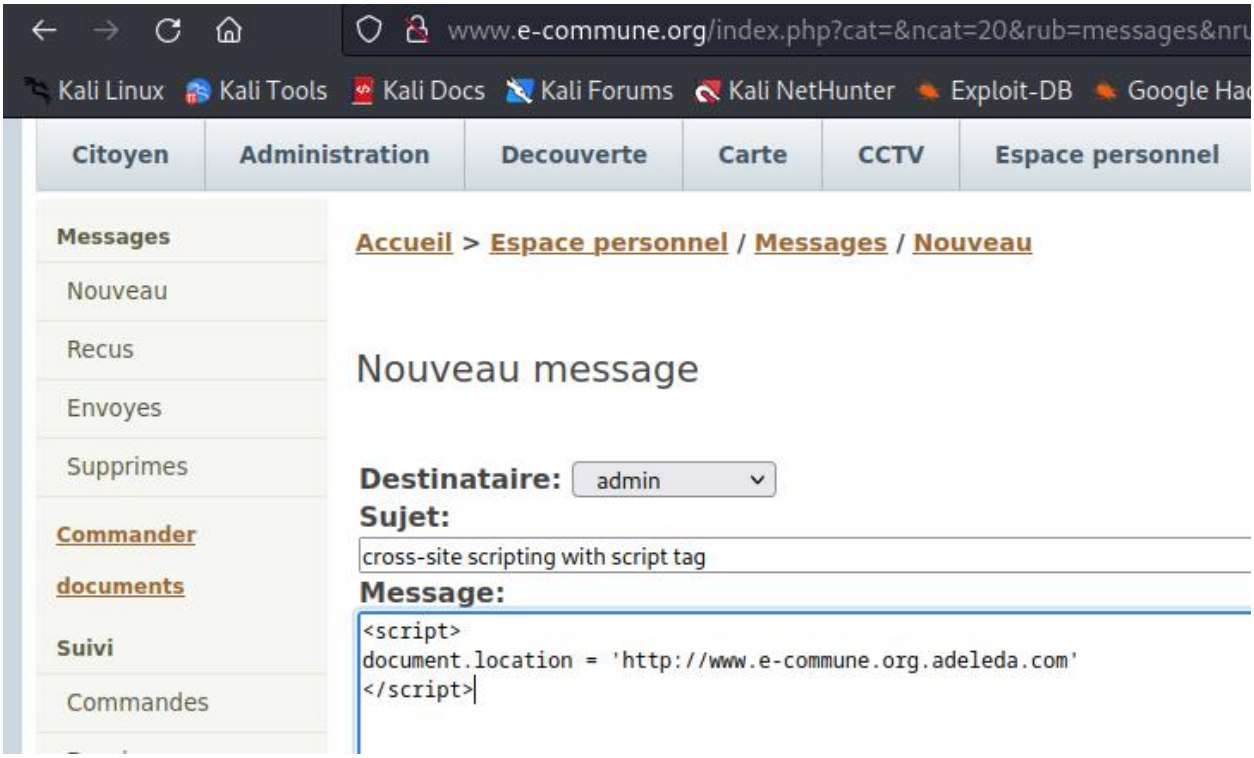
Exploitation:



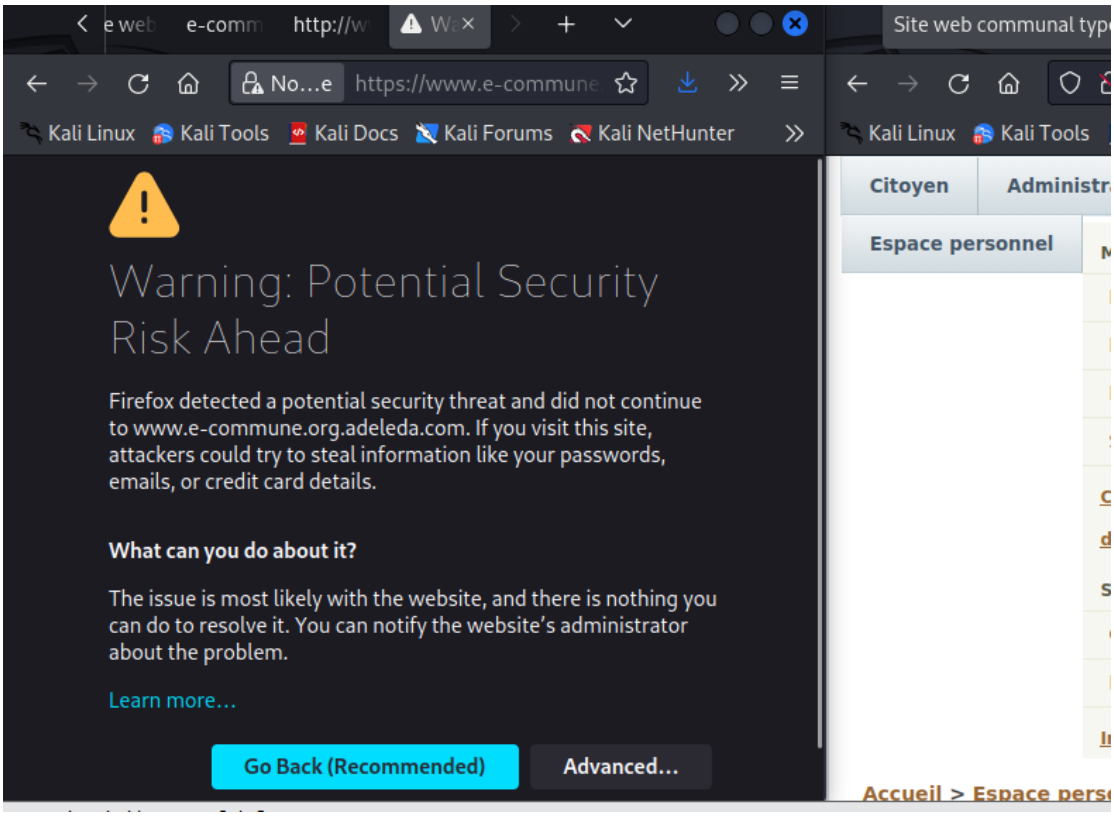
1. Used the accessed credentials to login 1) on the admin and 2) as a regular user



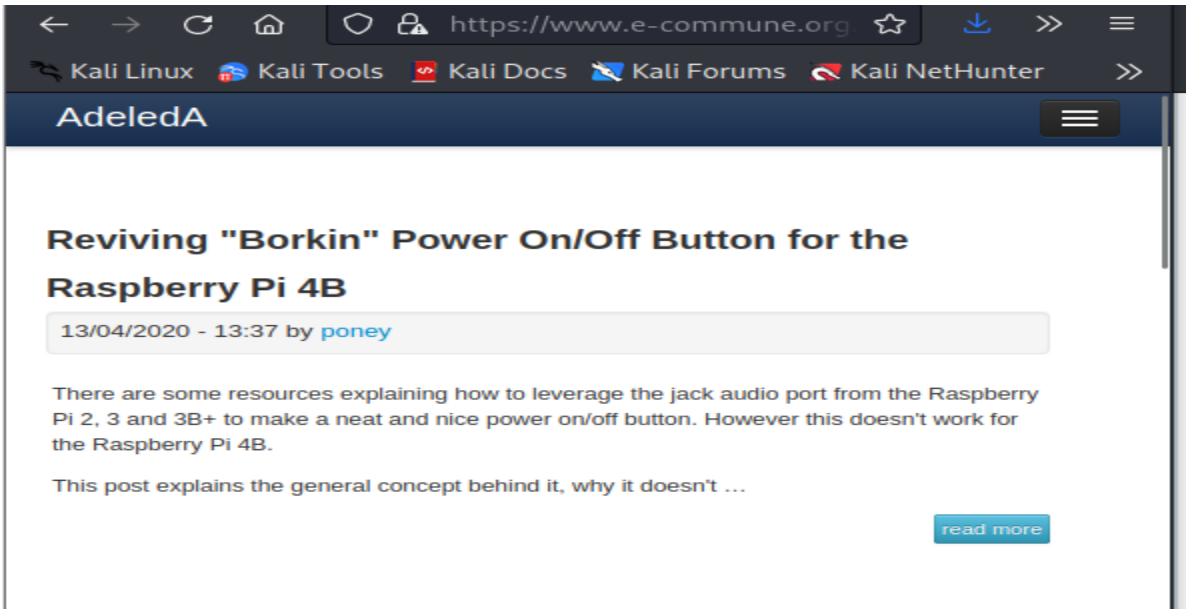
2. message and communication are successful from the user to admin



3. Used a basic script to send the message to the admin



4. Redirection to the phishing page success



5. Accessed the page when admin clicks the message

Remediation:

1. Applying or using vetted library which prevents from the weakness to occur. E.g. – using libraries like Microsoft anti -Xss library

2. Applying appropriate encoding to al alpha numeric based characters

    - HTML body
    - JavaScript sections
    - URIs
3. Building awareness of incoming where untrusted inputs reach the site of the software :cookies , parameters, arguments, and anything from the network
4. When using php based application configure it to avoid using register_globals this will limit the user supplied data on the cookies, query parameters etc.

Referencing

1. Learning Center. What is Content Security Policy (CSP) | Header Examples | Imperva. [online] Available at: https://www.imperva.com/learn/application-security/content-security-policy-csp
2. Using hydra tool. https://www.kali.org/tools/hydra/ - brute force attacks
3. https://www.mediawiki.org/wiki/Register_globals - register_globals
4. dirbuster tool- https://www.kali.org/tools/dirbuster

Projects folder structure:

~/projects: root folder for all projects
~/projects/_epita: template folder duplicated and renamed for each project/customer.

~projects/_epita/_e-commune/info.txt: notes about the scope, contact, app's description, credentials, …
~projects/_ epita /_ e-commune /screenshots: contains all screenshots from the audit (raw images)
~projects/_ epita /_ e-commune /vulnerabilities: contains one folder per vulnerability (e.g. 01_Cross-Site_Scripting)
~projects/_ epita /_ e-commune /vulnerabilities/<id_folder>/: contains selected screenshots that will be used in
the report
~projects/_ epita /_ e-commune /report: contains the report in docx and pdf
~projects/_ epita /_ e-commune /others: has everything else