WEB APPLICATION SECURITY

Audit Report for e-commune.org

Akhilesh Bandi

Contents

Synthesis

Why is this Audit performed?

To observe the web security of the domain and hosted content of e-commune.org with main objective to identify and evaluate the key vulnerabilities, weaknesses and assess their potential business impacts.

What did this Audit find?

The application is unsuitable to be deployed in real world scenario having zero protection against the OWASP Top 10 Web application vulnerabilities.

What are the recommendations?

An entire overhaul is recommended as there are disastrous vulnerabilities in almost all stages of using the web application. A redesign following zero trust and good cyber hygiene is required.

Scope of the Audit

The audit was performed on e-commune.org by pointing attacker computer to HostVM over same local network, but essentially the attacker can be in any external network once the site is deployed. No credentials or prior knowledge about it are provided before the audit.

Common Vulnerability Scoring System

Severity	Description	Total Vulnerabilities
CRITICAL	Compromised everything confidentiality, Integrity, and Availability. Immediate remediation is required to avoid business impact	2
HIGH	One of CIA is compromised, and the exploitation is slightly complex.	6
Medium	Potential impact on one or all of CIA but the attack is complicated	1
Low	No measurable impact on CIA with a highly complicated attack and low probability	0
Information	No actual vulnerability identified but odd behavior or information identified	0

The CVSS scoring system allows us to depict a vulnerability and its characteristics in the form of a score based on its severity. This numerical score is then translated into 5 stages of severity. CVSS calculators enable generating a severity score based on the total attack vectors and other features of the vulnerabilities.

Vulnerabilities

1. Secrets in Plain Text

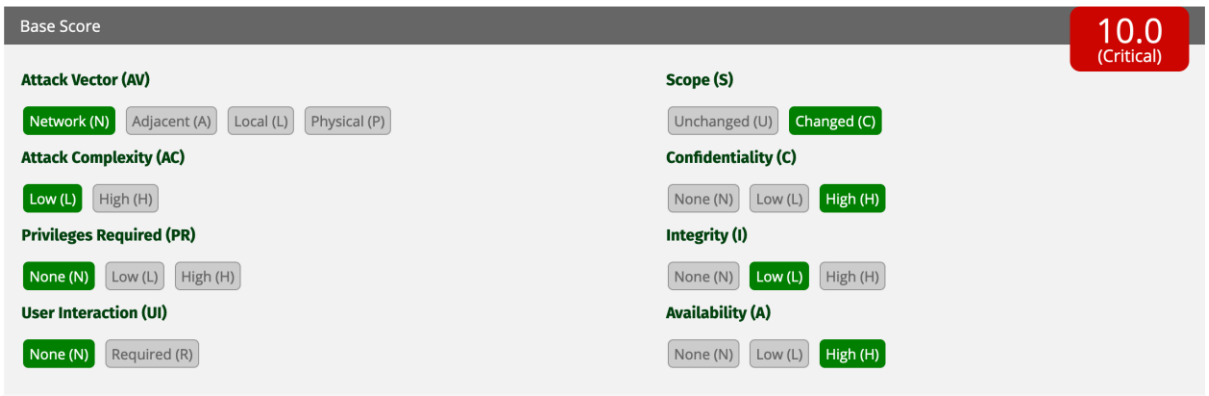
CWE ID : CWE-200

CVSS Score : CRITICAL(10)

Business Impact Criticality : High

Exploitation Difficulty : Easy

Remediation Difficulty : Easy



Description:

The admin password is bombarded onto the user irrespective of their authorization. Going to the login page immediately lists out the admin password. The attack has zero complexity and can be exploited by anyone who visits the webpage.

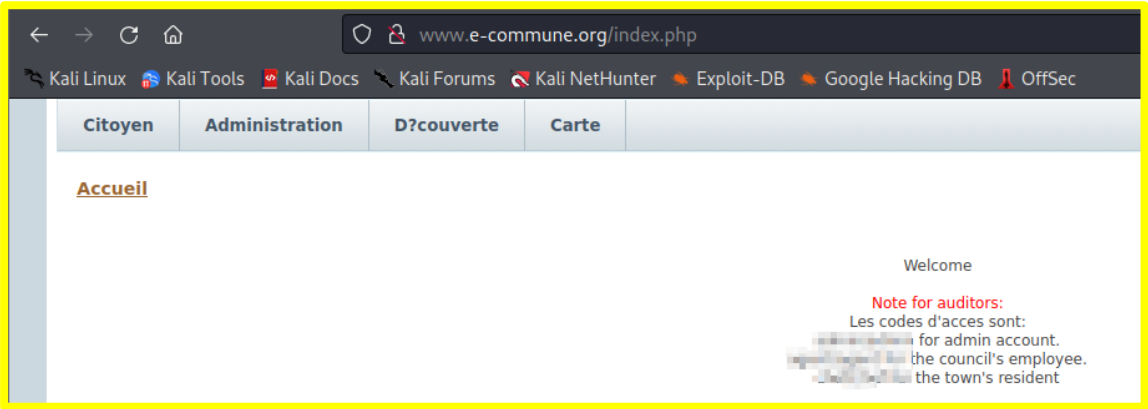
There are few other vulnerabilities where personally identifiable information is compromised with much less criticality but fall under the same CWE.

Usernames and their passwords are presented by going through the website.

Viewing the page source also reveals a lot of info about files that have usernames, passwords, addresses and GPS Locations.

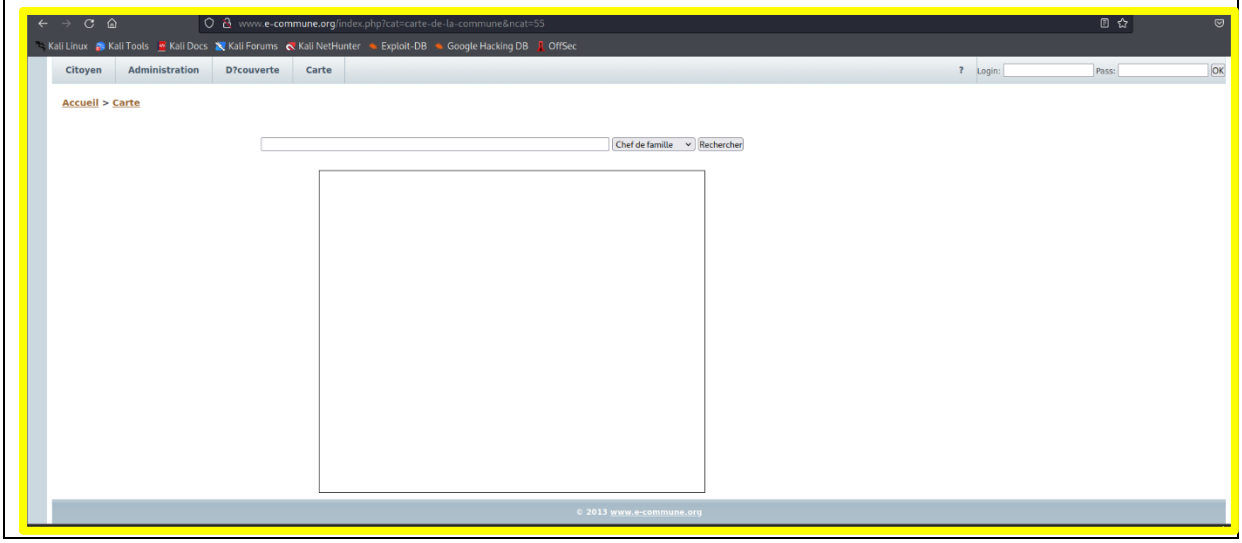
Exploitation:

a) Open any browser of your choice and visit the e-commune.org website.



Admin and other usernames and password being displayed right on the main page.

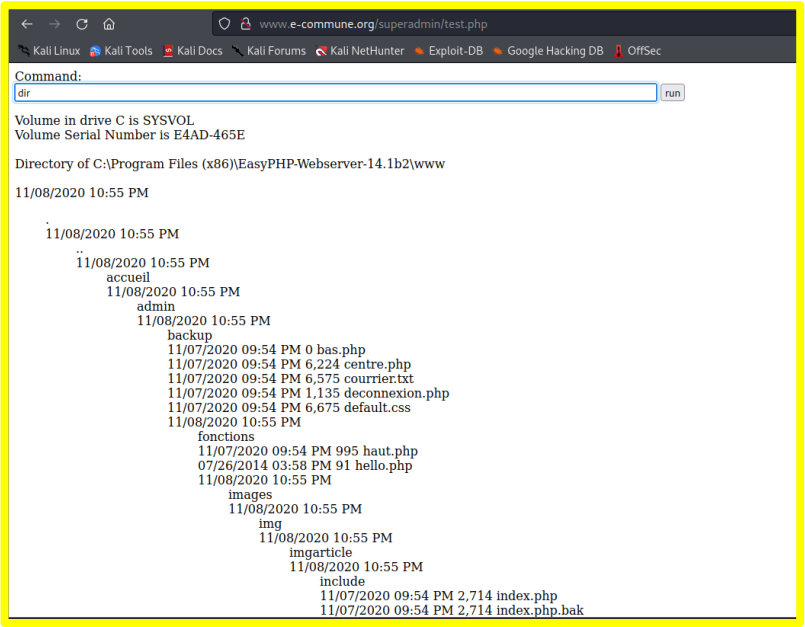
b) Visit the carte and hit view page source



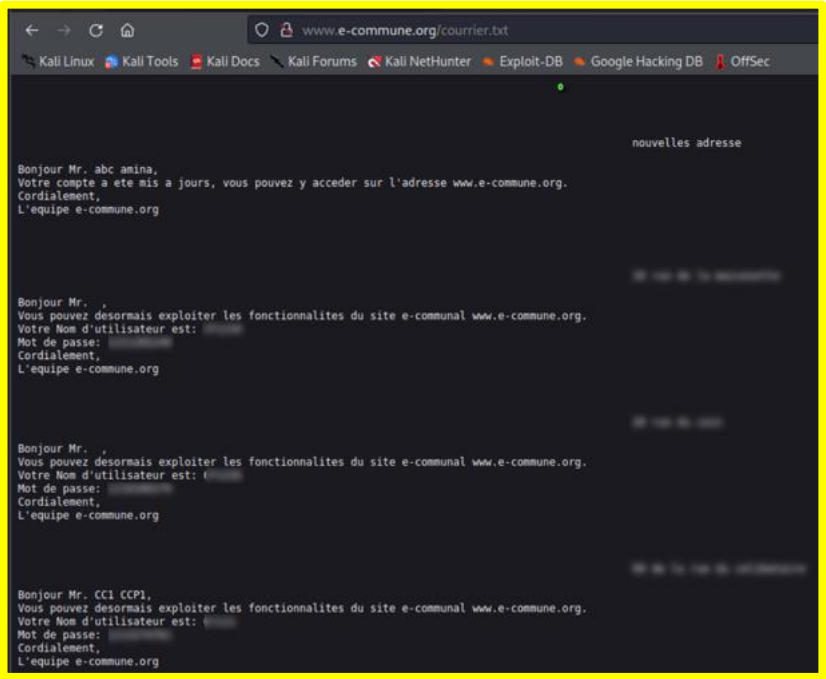
Scroll to the bottom in the page source to find the below

```
<a href="superadmin/test.php"></a>      </div>

        <div class="clearer"><span></span></div>

    </div>

    <div class="footer">&copy; 2013 <a href="index.php">www.e-commune.org</a>
    </div>

</div>

</body>
</html>
```

Going to the superadmin folder in which the following can be searched

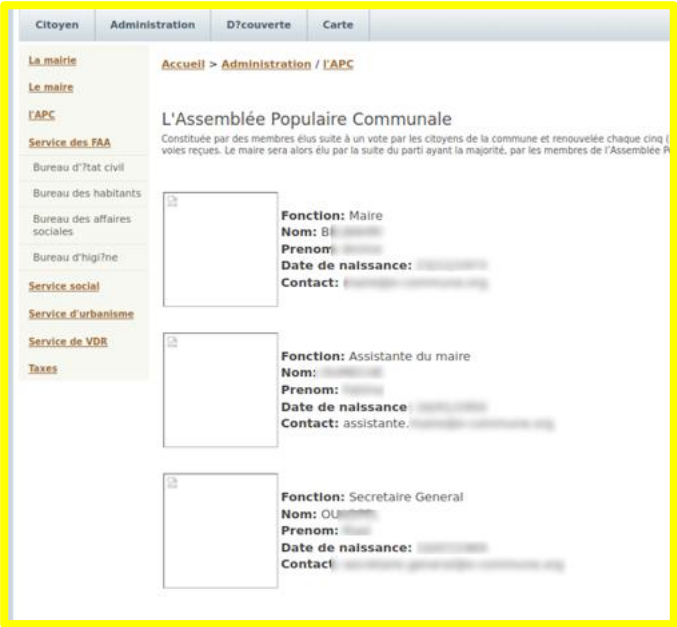


Going through the found files courier being one of them reveals the follows. The attacker can also directly navigate to e-commune.org/courier.txt if he has knowledge of the aplication's file structure.



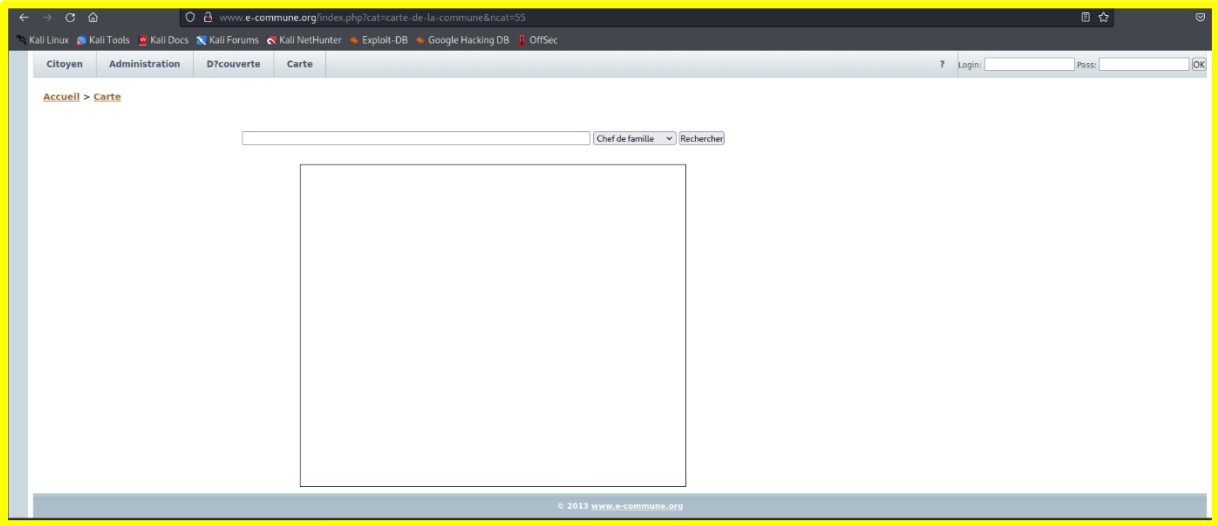
User's login, password and addresses showing up in courier.txt

c) On the website visit accuel>Administration>L'APC



Keep exploring the webpage to find the above details published on the given page

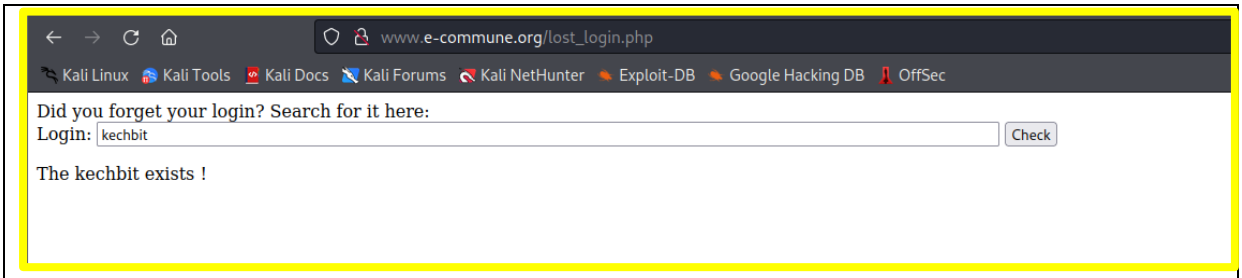
d) Blank search in carte revealing details.



Visit the same carte page again and iniate a blank search.

```
profil_html0='KHALDI Adel <br/> 10 rue de la maison<br /> Latitude: 36.7523<br /> Longitude: 3.04181<br />';
var marker = new GMarker(new GLatLng(36.7523, 3.04181));
GEvent.addListener(marker, "click", function() { marker.openInfoWindowHtml(profil_html0);});
map.addOverlay(marker);
profil_html1='KECHBIT Abdelkrim <br/> 22 rue docteur fares<br /> Latitude: 36.424<br /> Longitude: 3.10448<br />';
var marker = new GMarker(new GLatLng(36.424, 3.10448));
GEvent.addListener(marker, "click", function() { marker.openInfoWindowHtml(profil_html1);});
map.addOverlay(marker);
map.setCenter(new GLatLng(36.424, 3.10448),13);
map.setMapType(G_HYBRID_MAP);
}; // initMap
```

We are presented with usernames and their GPS locations.



Searching for the same username reveals that the user infact exists.

Remediation:

1. Avoid storing passwords in easily accessible locations

2. Adopting strong encryption- using a strong one-way hash algorithm

3. Salting the password and performing multiple key hashes

The above actions will prevent brute force attacks or lookup attacks on passwords. In addition, OWASP describes the exploitation and remedial actions

2. Cleartext Transmission of Sensitive Information

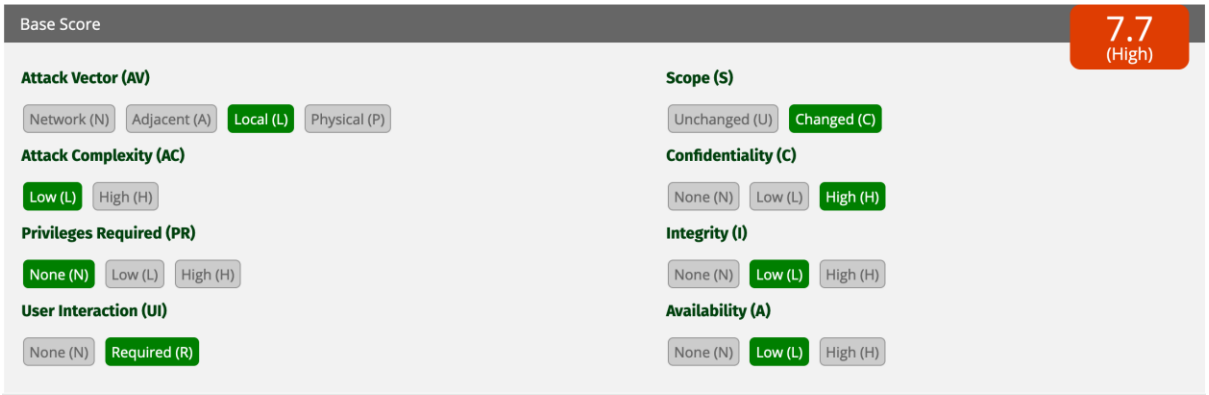
CWE ID : CWE-319, CWE:523

CVSS Score : HIGH(7.7)

Business Impact Criticality : Medium

Exploitation Difficulty : Medium

Remediation Difficulty : Easy

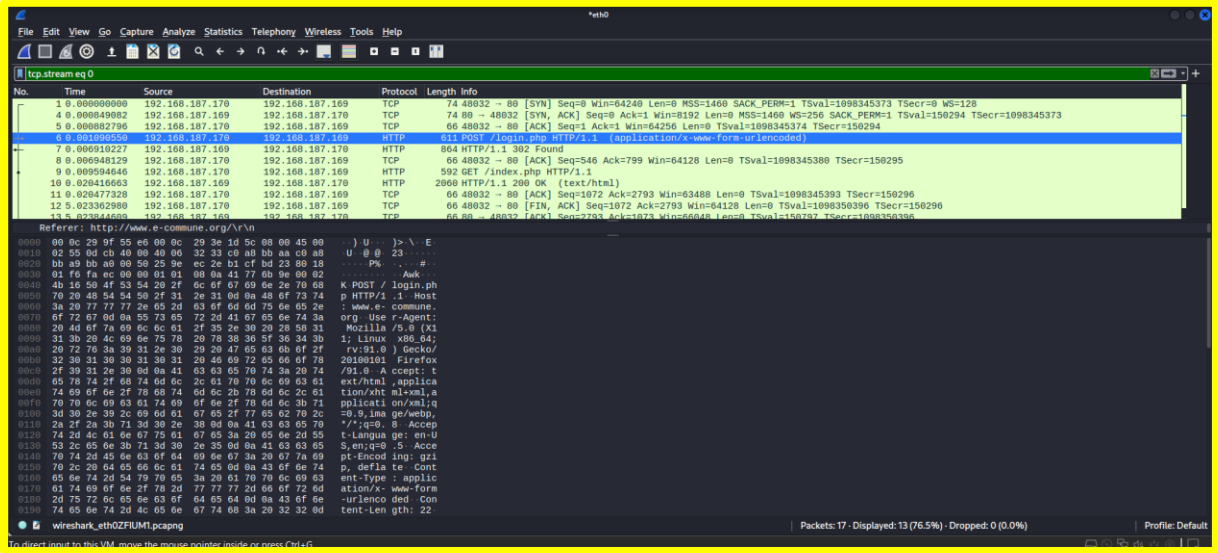


Description:

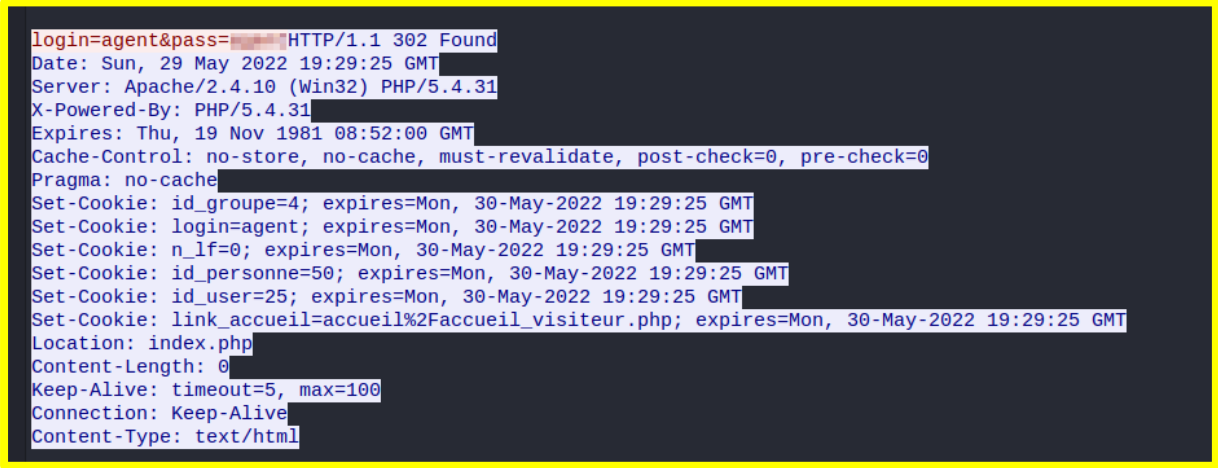
Plan text network transmission of sensitive information. While logging in the network packet can be intercepted. As the packet isn't communicating over HTTPS an attacker can view it. The exploit is relatively easy to perform with sniffing software and requires no permission for the attacker. Compromises everything depends on the user level whose details are compromised. Attacker can never be identified due to obscured trails. An attacker can intercept being in the same network.

Exploitation:

Download wireshark and listen to the packets from the client session



A login.php can be observed in wiresharkt.



Opening the above would expose the login credentials of the user.

Remediation:

All communication should be mandated over HTTPS to ensure encryption. The application should deploy session tokens to identify specific session instances allowing for future audits, On the Server side only accept packets with some form of TLS encryption. More about it here.

3. User Enumeration

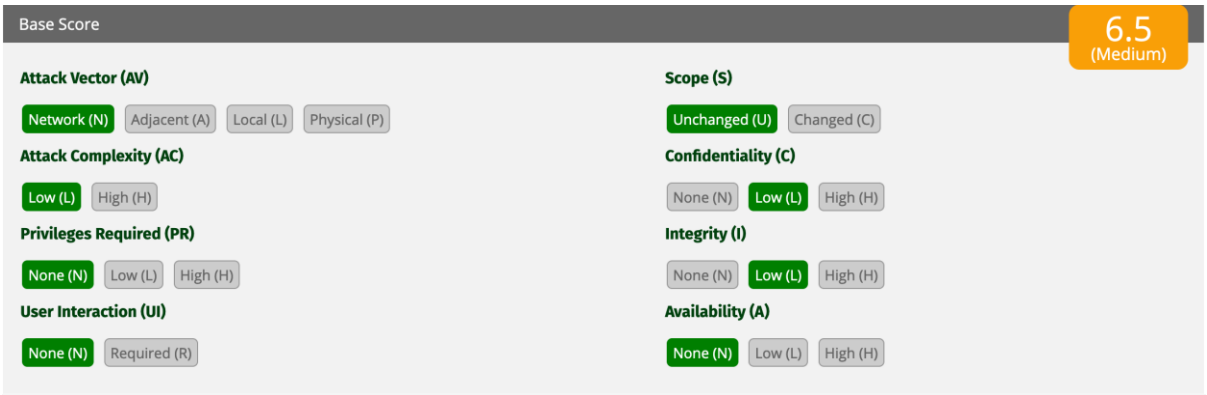
CWE ID : **CWE-204**,

CVSS Score : MEDIUM(6.5)

Business Impact Criticality : Low

Exploitation Difficulty : Medium

Remediation Difficulty : Medium

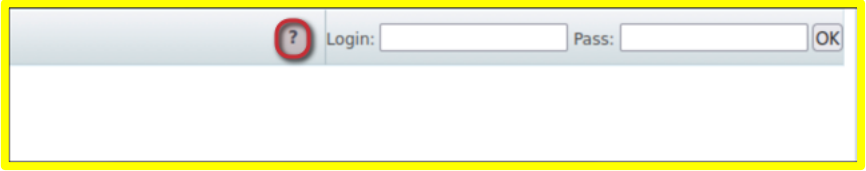


Description:

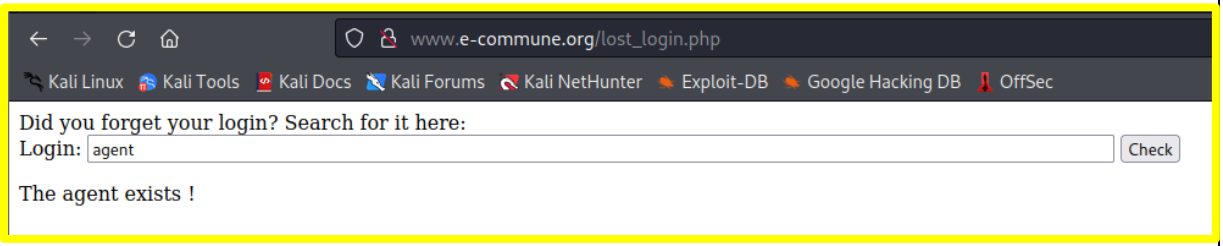
The attacker can verify if a particular user exists in the system or not. The attacker can guess or perform information gathering to obtain these usernames. The website has a forgot password page which returns if a user exists or not instead of simply taking the inputs and performing password reset tasks in the background. The said page eases the process for attacker as they should otherwise figure out a backup plan to not alert the system while bruteforcing or guessing values.

Exploitation:

Visit the website and click on the Question Mark (?) button

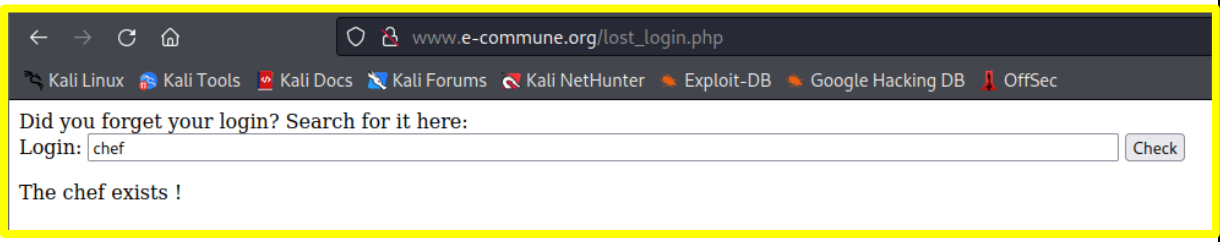


In the page that follows search for common or researched usernames to verify



Observe that the website returns that the given user infact exists

other examples



Remediation:

Effective remediation would be to have the server respond with a generic message that does not indicate which field is incorrect. When the response does not indicate whether the username is incorrect, the malicious actor cannot infer whether usernames are valid. More on it here.

4. Directory Listing

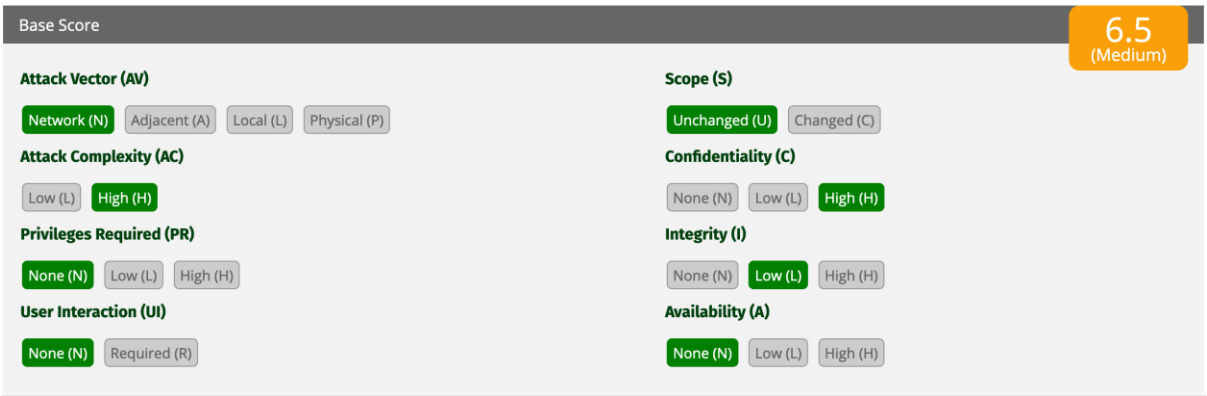
CWE ID : **CWE-ID:548**

CVSS Score : MEDIUM(6.5)

Business Impact Criticality : Medium

Exploitation Difficulty : Medium

Remediation Difficulty : Easy



Description:

Directories are easily exposed when inspecting the console of any image. Entering similar URL in browser lets the attacker see all the folders of website, Although replacing or editing them isn't possible the attacker can access, download contents within these folders.

Exploitation:

Find any image from the website and hit inspect from the right click menu. The inspector section from the browser shows the filepath of the particular image.

Observe That src="imgarticle/administration/mairie.jpg"

Entering the path into the URL reveals the entire folder architecture.

Remediation:

Content The server can be configured such that Directory Listing is disabled. Once disabled when someone tries to enter a directory into the URL bar they'd be greeted with an error or the configured page. The ways to disable Directory listings on different webservers are different. Here is a link that talks about them all

5. Remote Execution as admin

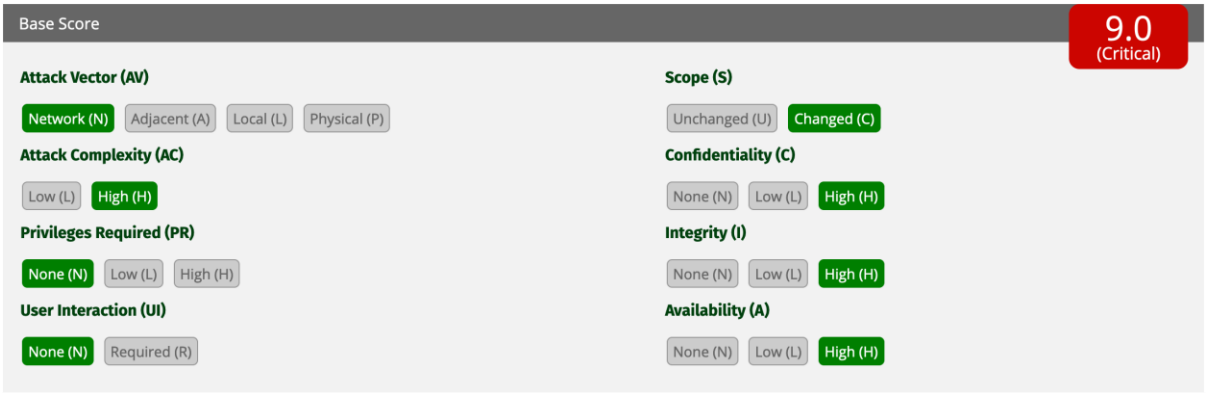
CWE ID : CWE-250

CVSS Score : CRITICAL(9.0)

Business Impact Criticality : High

Exploitation Difficulty : High

Remediation Difficulty : Medium

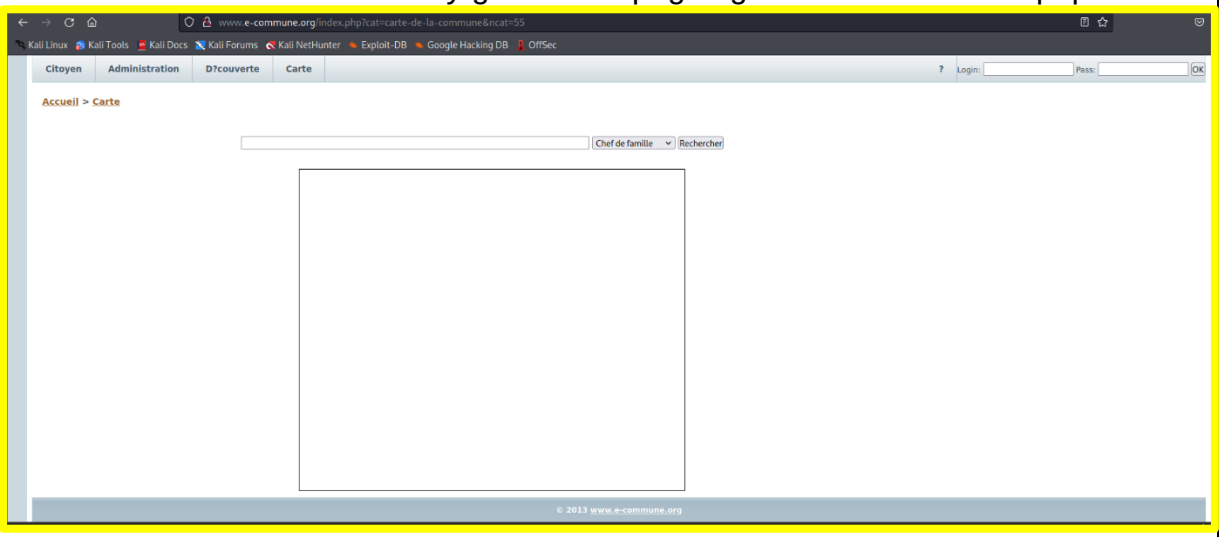


Description:

With the help of previous vulnerability there is a possibility for an unauthorized user to find the superadmin.php in the inspector section of their browser. They can execute commands as this privileged user without any credential verification. Logs if any will be stored on behalf of that user and the attacker's trails would be fully covered. Although there is a word limit for these commands the attacker can again change the value from the inspector section.

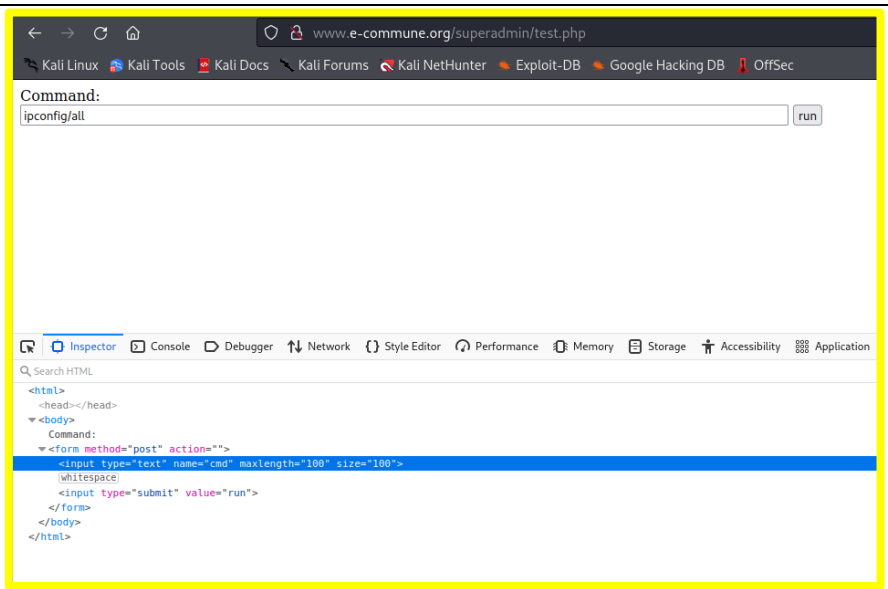
Exploitation:

Like the first vulnerability go to carte page again and find the test.php

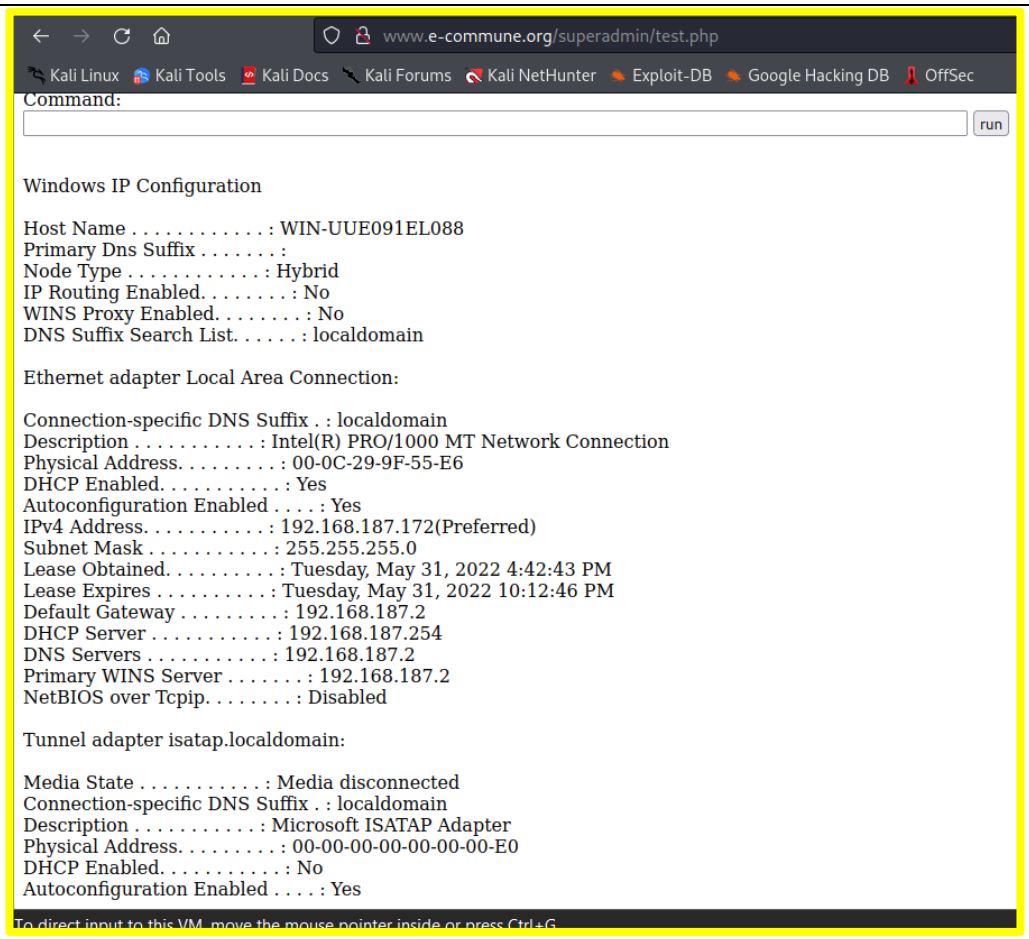



```
<a href="superadmin/test.php"></a>      </div>

        <div class="clearer"><span></span></div>

    </div>

    <div class="footer">&copy; 2013 <a href="index.php">www.e-commune.org</a>
    </div>

</div>

</body>
</html>
```

Use the superadmin/test.php from above



open the inspector to change the allowed command length from 3 to any other integer



The attacker can now execute any command as superadmin

Remediation:

Implementing some kind of access control before accepting commands is needed. A command window within the application can be deleted entirely and opt for alternatives like remote desktop or ssh which would update based on repositories solving the issue and also saving overhead in developing the command page.

6. Insecure Password Storage

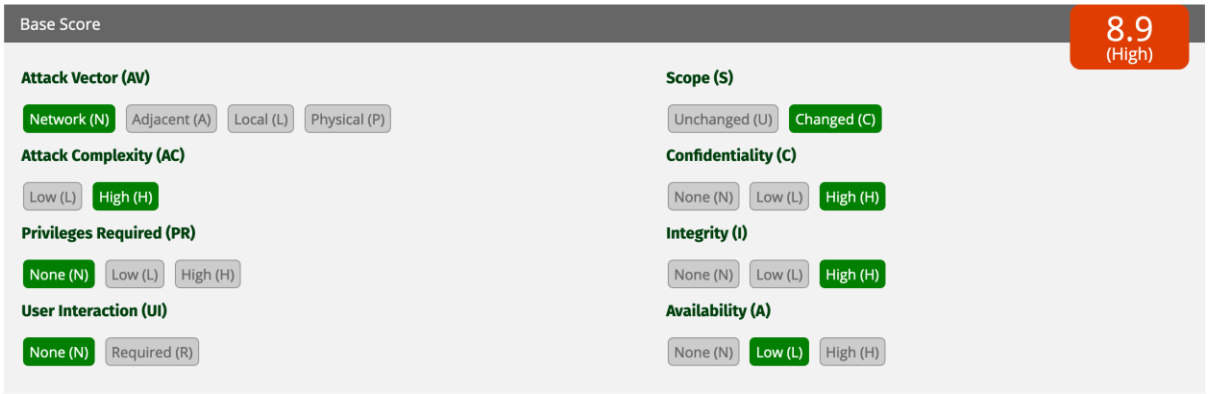
CWE ID : CWE-260, CWE-916

CVSS Score : HIGH(8.9)

Business Impact Criticality : High

Exploitation Difficulty : Medium

Remediation Difficulty : Medium

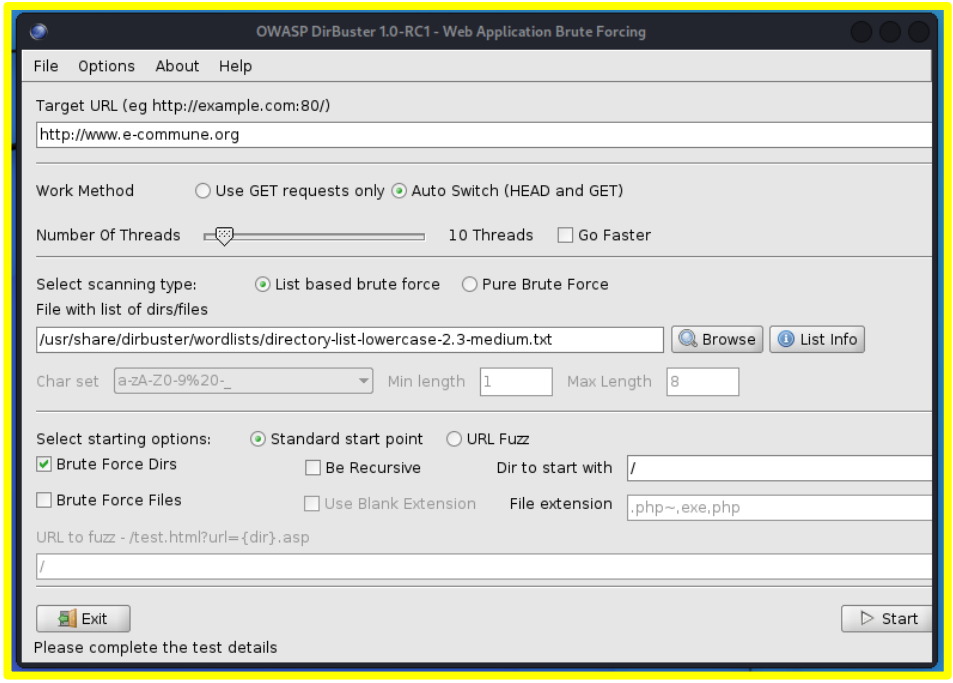


Description:

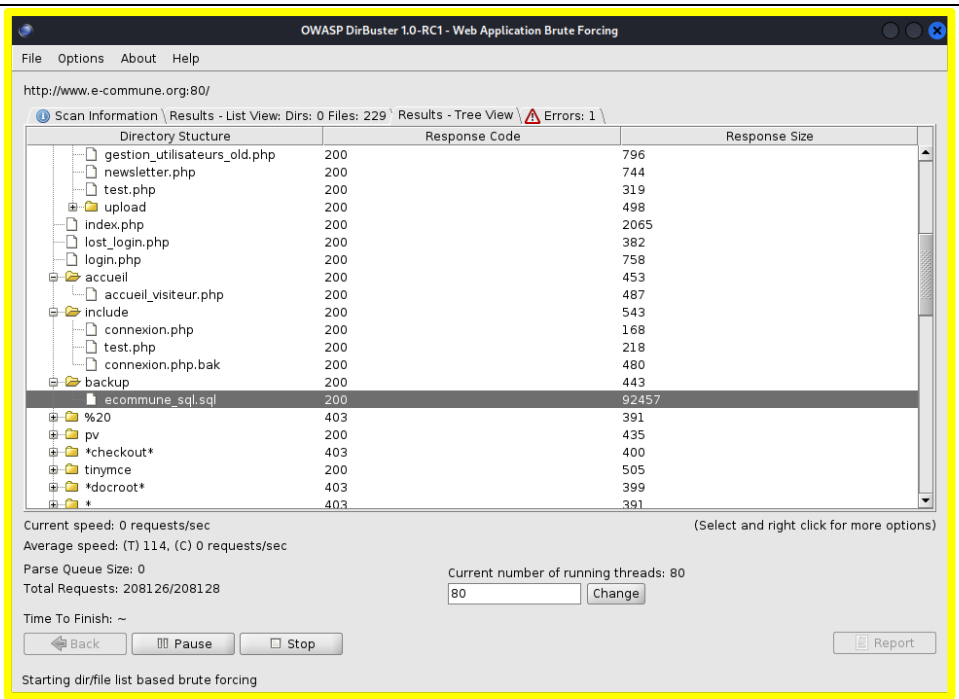
Passwords can be extracted out of a SQL file that is found within viewable file system architecture. The web application can be bruteforced into revealing the entire directory lisitng which inturn can be utilized to perform severe attacks. The said SQL file holds table values of usernames and hashed passwords. The passwords are hashed with MD5 without any salt hence allowing the attacker to reverse the hashing to obtain the password in plaintext.

Exploitation:

Download and install DirBuster or with $ sudo apt install dirbuster.



Provide the target URL in DIR Buster



Browse through the directories and arrive at the above file

```
INSERT INTO `utilisateur` (`id_user`, `login`, `password`, `id_groupe`, `email`, `solde`, `n_lf`, `id_personne`) VALUES
(1, 'admin', '21232f297a57a5a743894a0e4a801fc3', 0, 'none@localhost', 0, 0, 5),
(2, 'test', '94c93d4f9686cfeae29e9cbc3230cbf9', 2, 'my@mail.com', 0, 0, 0),
(8, 'chef', 'cbb4581ba3ada1ddef9b431eef2660ce', 2, 'chef@hotmail.com', 1245, 1, 0),
(10, 'kechbit', 'e8d2ee64c4c701be152764dd018c2535', 2, 'none@localhost', 3, 2, 0),
(12, 'pp', 'c483f6ce851c9ecd9fb835ff7551737c', 3, 'none@localhost', 0, 0, 1),
(22, 'CC111', '0337c12814f13e6788d8da4f47183342', 3, 'none@localhost', 0, 0, 46),
(23, 'CC911', '632177b53d37237176a7e20a22fe7844', 3, 'none@localhost', 0, 0, 47),
(21, 'CF1235', '8c74fe7ec4bf35670c33c520ea1a07cf', 2, 'none@localhost', 0, 1235, 0),
(20, 'CF1234', 'f1190a994bdc4910a9787649d93d819e', 2, 'none@localhost', 0, 1234, 0),
(24, 'CC1212', '7352c705a8ad5311ca4396ce96cb8830', 3, 'none@localhost', 0, 0, 4),
(25, 'agent', 'b33aed8f3134996703dc39f9a7c95783', 4, 'none@localhost', 0, 0, 50);
```

Opening the file reveals SQL table data with login, password and other details.

```
-- Contenu de la table `live_camera`
--

INSERT INTO `live_camera` (`id_lc`, `adresse_ip`, `password`, `etat`, `titre`, `description`) VALUES
(2, '4.2.2.2', '123', 1, 'Centre ville', 'Visualisez le centre ville en direct !!!'),
(4, '1.1.1.1', '123', 0, 'sdf', 'dsf'),
(5, '1.1.1.1', '123', 1, 'sdf', 'dsf');
```

Locations and other details of LiveCamera are also obtained

Install a password cracking utility like John the ripper or Hashcat

```
┌──(kali㉿kali)-[~]
└─$ john --single hashes.txt --format=raw-md5
Using default input encoding: UTF-8
Loaded 10 password hashes with no different salts (Raw-MD5 [MD5 128/128 AVX 4×3])
Remaining 5 password hashes with no different salts
Warning: no OpenMP support for this hash type, consider --fork=4
Press 'q' or Ctrl-C to abort, almost any other key for status
Almost done: Processing the remaining buffered candidate passwords, if any.
0g 0:00:00:00 DONE (2022-05-31 14:16) 0g/s 0p/s 0c/s 0C/s
Session completed.

┌──(kali㉿kali)-[~]
└─$ john --show hashes.txt --format=raw-md5
?:admin
?:adel
?:chef
?:kechbit
?:pp
```

Using the John the Ripper to crack the passwords

Remediation:

Choosing a better hashing function that makes reverse hashing and cracking non-viable.
Using a strong and slow hashing algorithm like **Argon2 or Bcrypt**, combined with salt (or even better, with salt and pepper) is required.

Preventing directory enumeration attacks that effectively work by placing a lot of requests to pages that don't exist returning an error 404. All these requests need to generate logs on the server which can in turn be used to detect and prevent additional requests from that IP either temporarily or permanently. Fail2Ban is one such software that prevents malicious activity.

7. Technical Information leakage

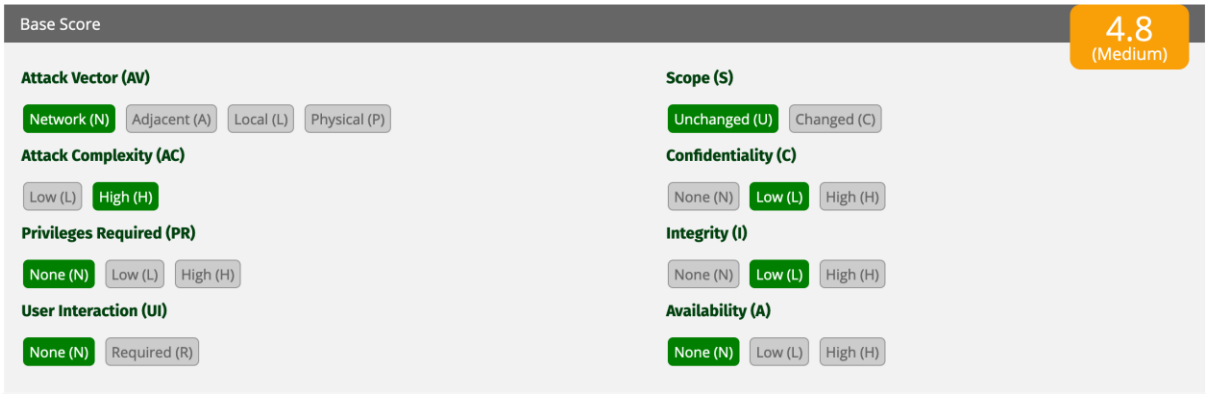
CWE ID : CWE-200, CWE-ID:497

CVSS Score : MEDIUM(4.8)

Business Impact Criticality : Low

Exploitation Difficulty : Medium

Remediation Difficulty : Easy

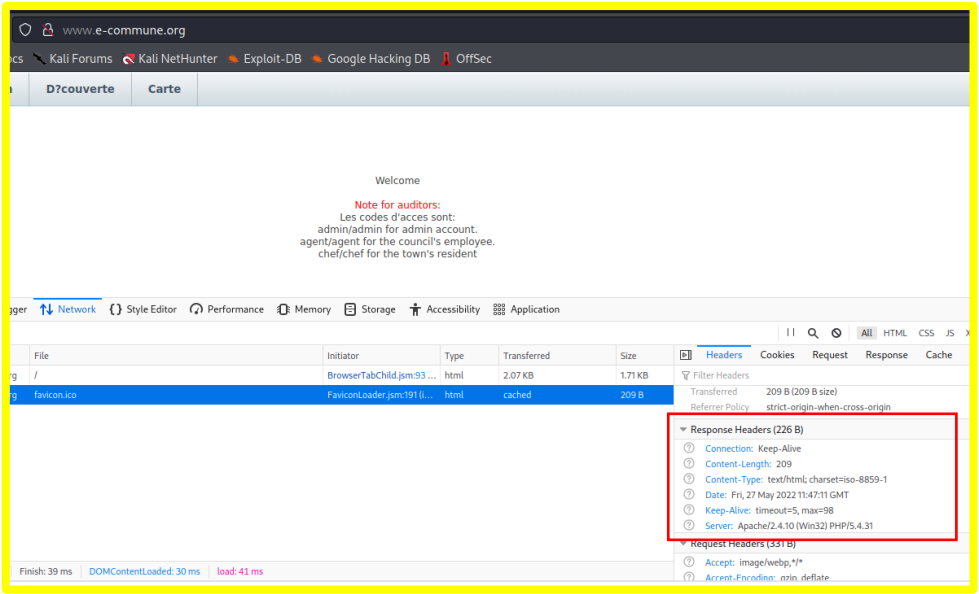


Description:

Technical information about the webserver answering the requests can be obtained from the network section of the browser console. This gives the attacker an understanding about the web server and eases his information gathering process. The attacker can then perform a targeted attack researching vulnerabilities based on the information obtained.

Exploitation:

Open the website and open the console.



Scroll down the header in the network section to find Server versions

Remediation:

Web server needs to be configured such that no unwanted response headers are displayed to the attacker. Various technologies have different methods to mitigate this like Link1 and Link2.

8. Backup and Temporary File Leakage

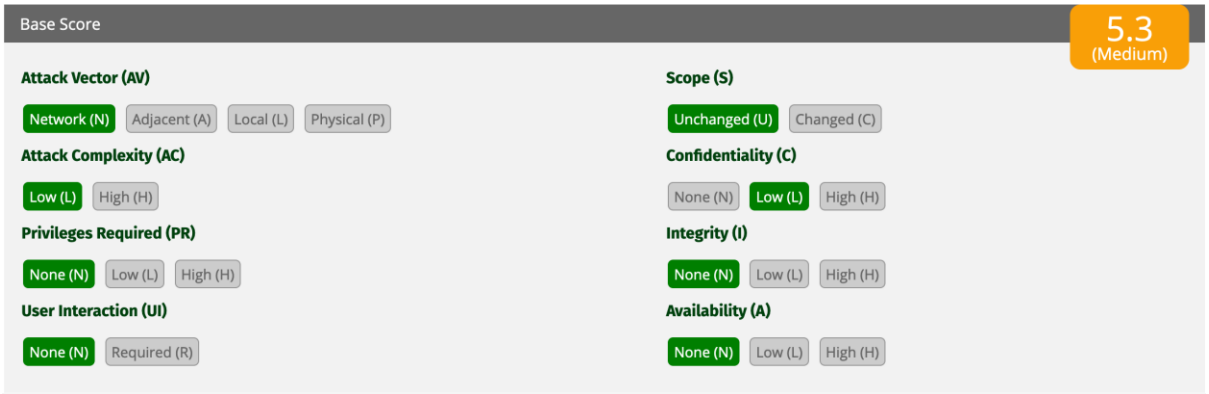
CWE ID : CWE-377, CWE-530

CVSS Score : MEDIUM(5.3)

Business Impact Criticality : Medium

Exploitation Difficulty : Medium

Remediation Difficulty : Easy

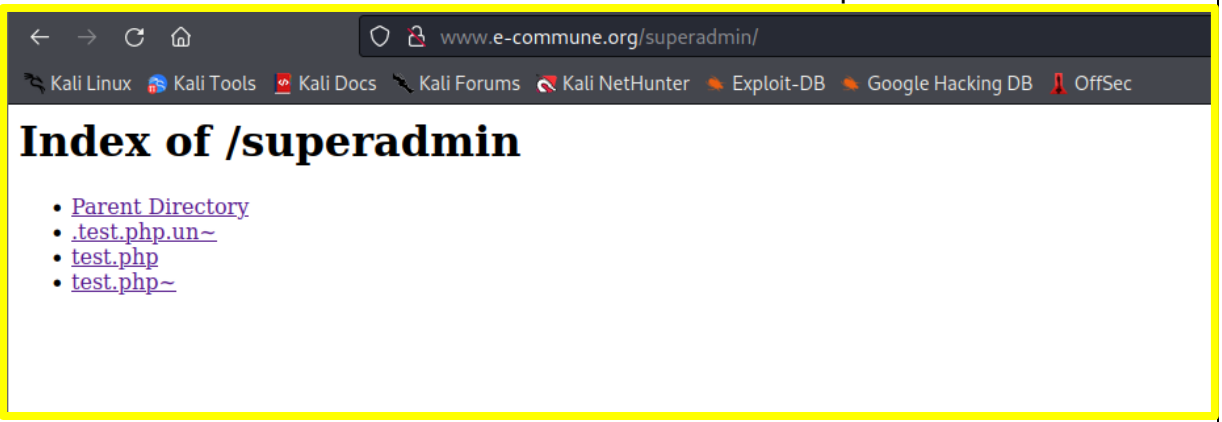


Description:

Backup and Temporary files can be found in the root tree that lets the attacker to observe the state and operations of web servers from back in time. Based on how the server is configured they might also compromise important and sensitive data.

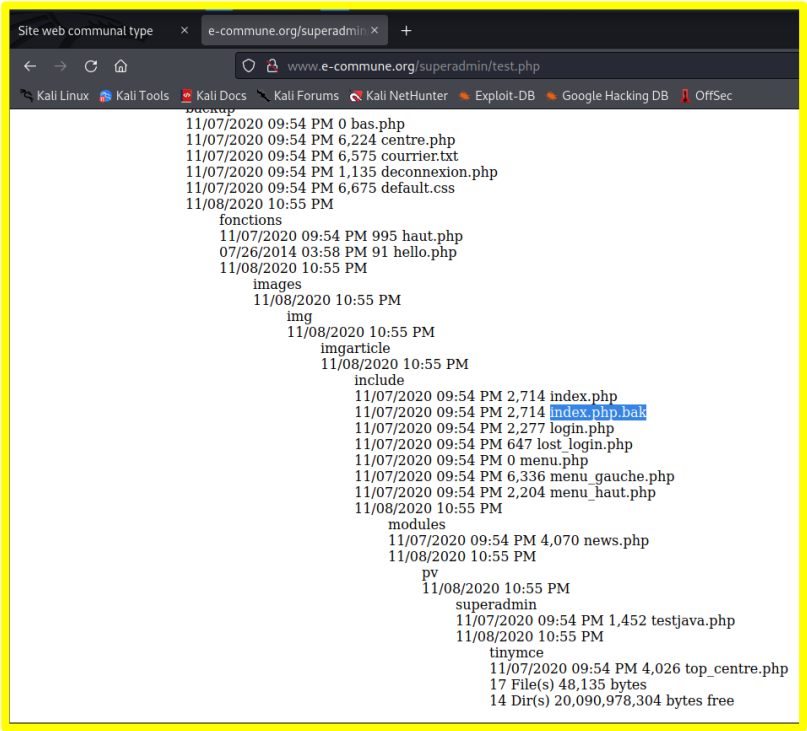
Exploitation:

Attacker can visit the below site. He can obtain this from previous vulnerabilities



Observe the ~ highlighting that the files are temporary.

A backup file can be obtained from the following tee



Remediation:

Reviewing content and their intended viewership. Removing them from root tree if they are needed publicly. Applying appropriate configuration management to remove temporary files and unused files or simply a script to delete them. More about it here.

9. Login Page Bruteforcing

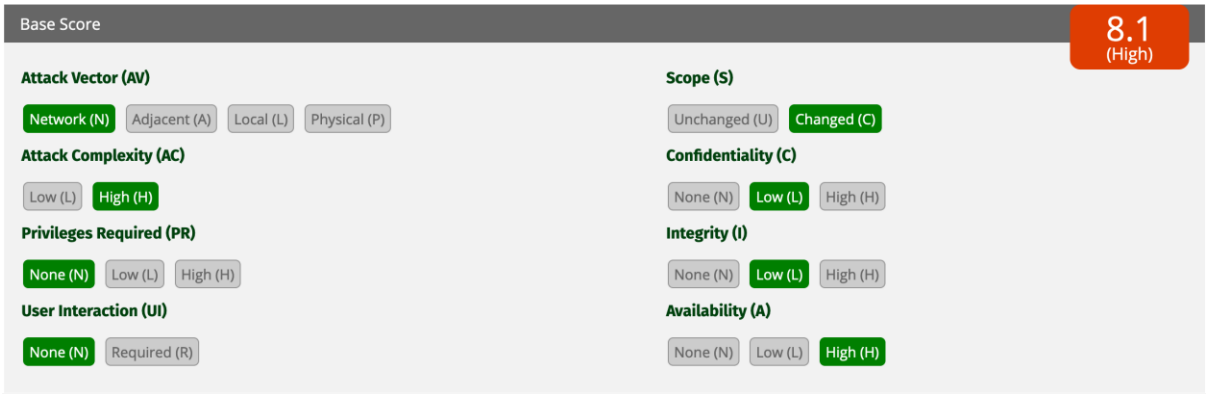
CWE ID : CWE-319, CWE-ID:799

CVSS Score : HIGH(8.1)

Business Impact Criticality : High

Exploitation Difficulty : Medium

Remediation Difficulty : Medium

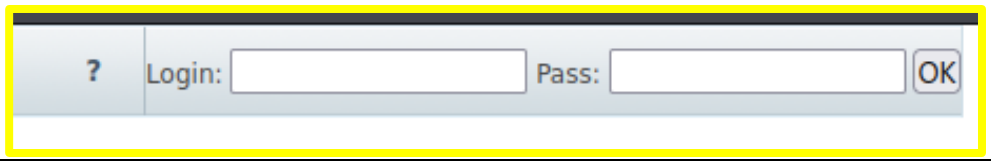


Description:

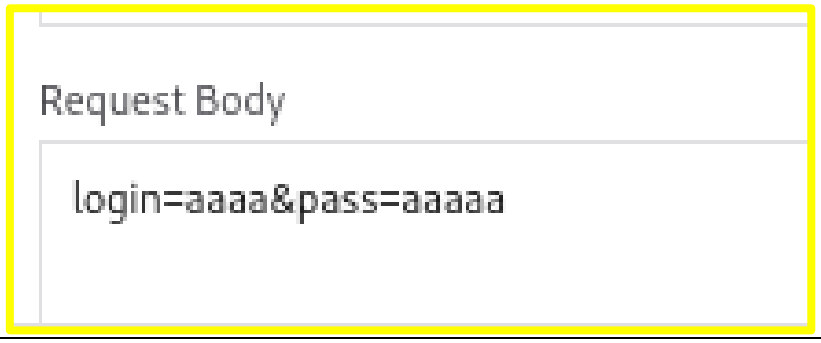
The login page has no preventive measure against too many failed authentication requests hence, making it susceptible to brute force attacks. This may cause bottlenecking for other users to answer every request of the attacker and the login credentials can be compromised by an attacker possessing limited knowledge about bruteforcing tools. The passwords themselves are weak and the attack takes very less time to perform.

Exploitation:

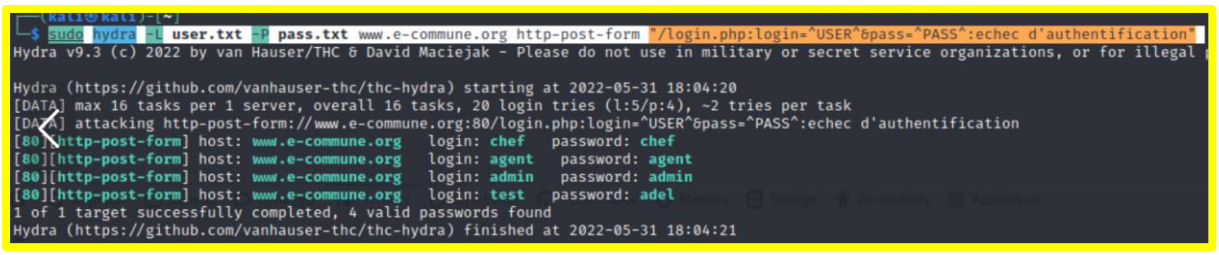
Navigate to the login section in the website



Open the inspector and observe the format of the fields



Install and open Hydra to perform a bruteforce attack by specifying URL in this syntax



All login credentials have been bruteforced in the attack.

Remediation:

Enabling configurations such that the account is temporarily locked after a few failed login attempts. Disabling too many requests from the same IP. Using cookies and session identifiers. Enabling a Captcha on authentication field to prevent automated attacks. More on this here.

10. Open Redirect to any URL

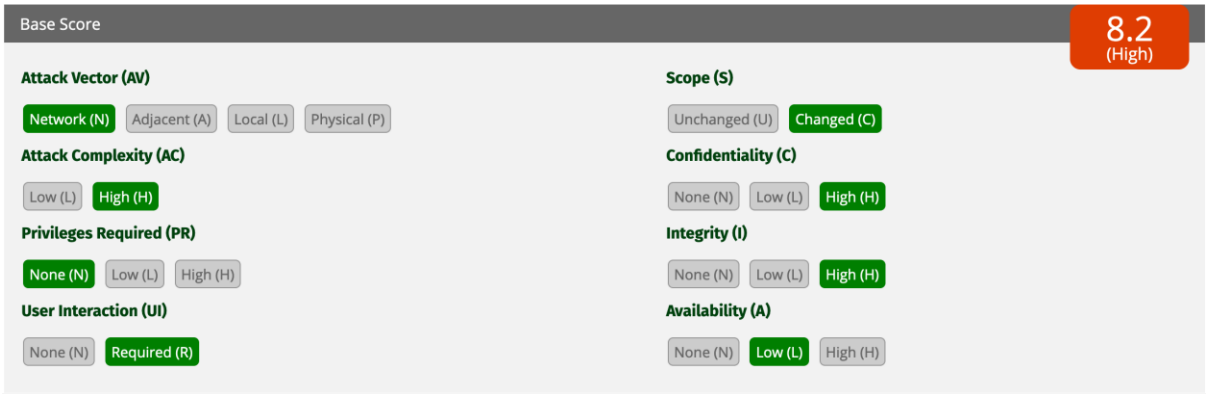
CWE ID : **CWE-601**

CVSS Score : HIGH(8.2)

Business Impact Criticality : Medium

Exploitation Difficulty : Easy

Remediation Difficulty : Medium

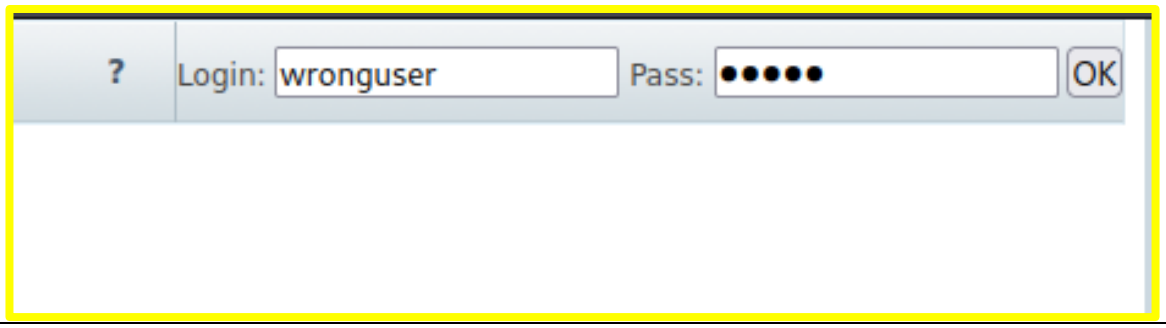


Description:

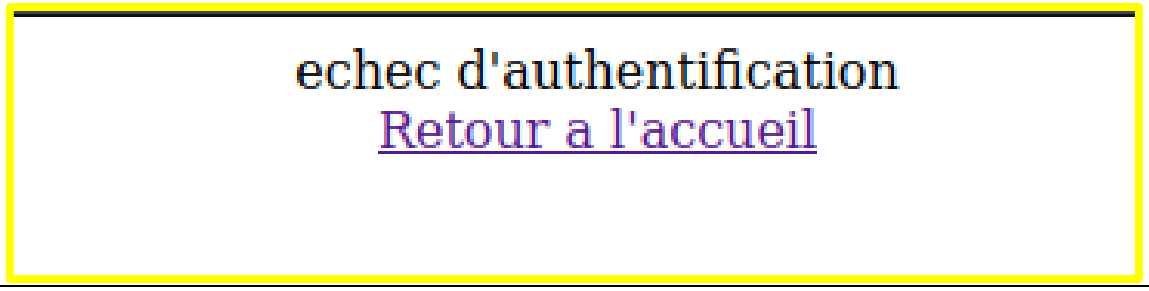
An attacker with no prior authentications can change the redirection URLs of the web application. It can be changed to any internal or external URL making it easily prone to phishing attacks and other malicious activities.

Exploitation:

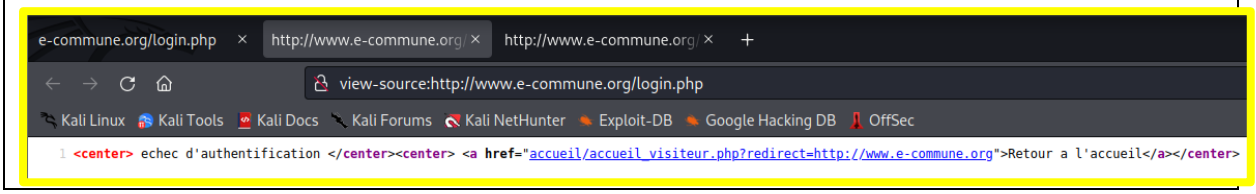
Navigate to login section and enter wrong credentials.



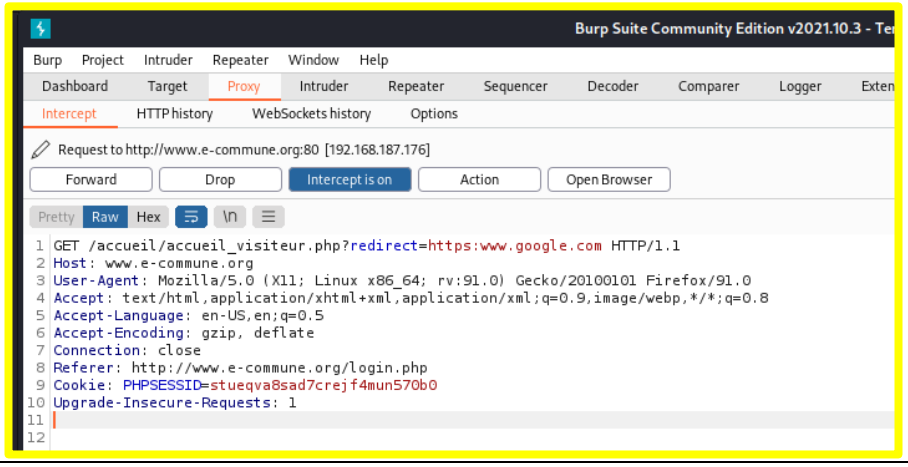
The below message will appear after that.



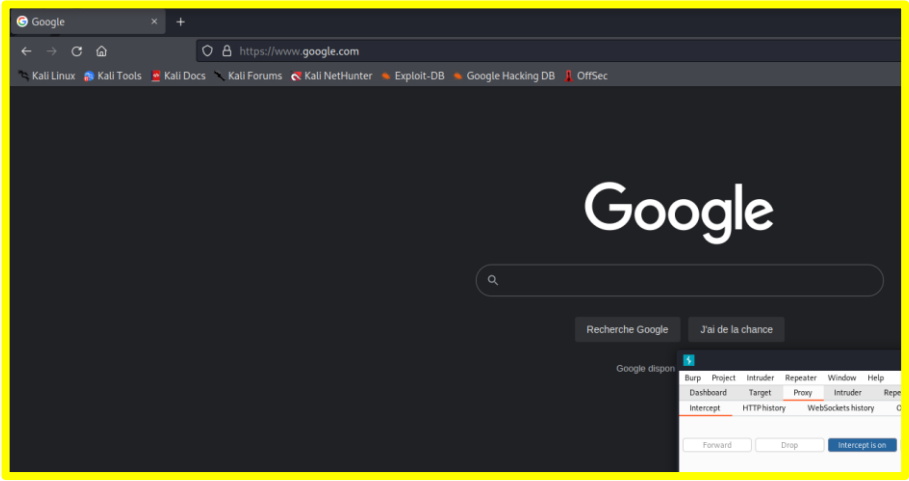
Open the inspector to identify URL.



Use Burpe Suite to intercept it and choose a URL ex google.com



Observe it now redirecting to given URL



Remediation:

Configuring the server to avoid all redirects to external links or only a few specific internal links. Comparing a part of the URL against a URL database and only letting the validated ones pass through or Declaring the URL exclusively in the source code. Link

11. Bad Profile Segregation

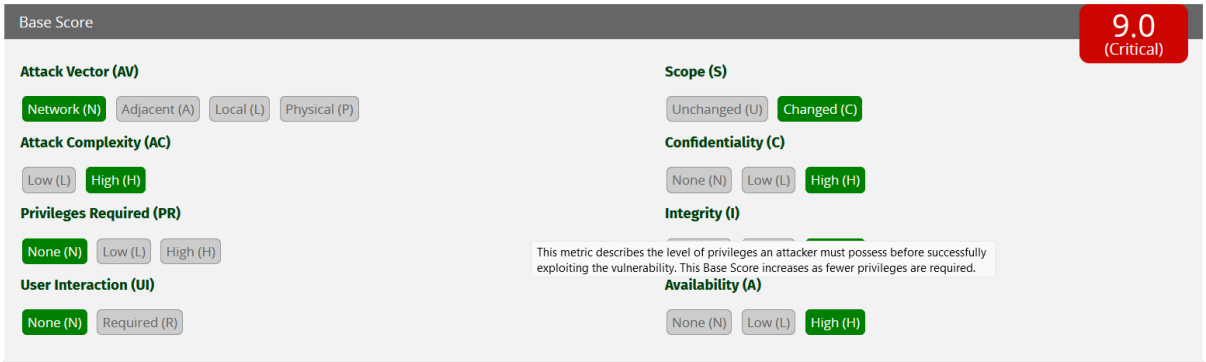
CWE ID : CWE-653, CWE-269

CVSS Score : CRITICAL(9.0)

Business Impact Criticality : High

Exploitation Difficulty : Medium

Remediation Difficulty : Medium

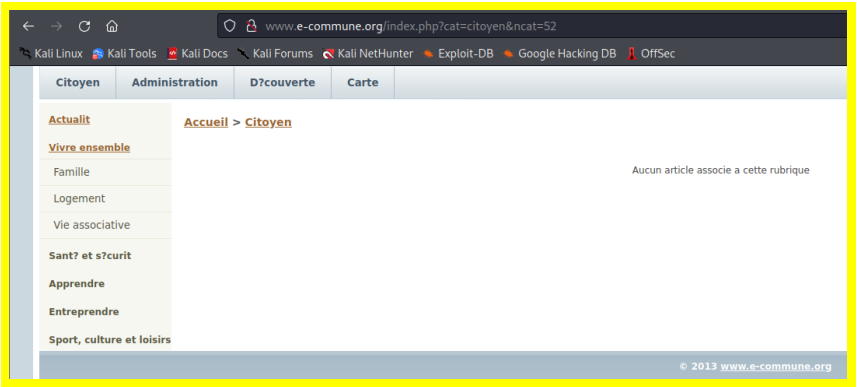


Description:

An attacker can change the values in the URL to access assets of any user including higher privileged users. No access control implemented when an under privileged user is navigating to a folder or file that isn't in the designated scope. An attacker can keep guessing with many values additionally with the directory listing vulnerability.

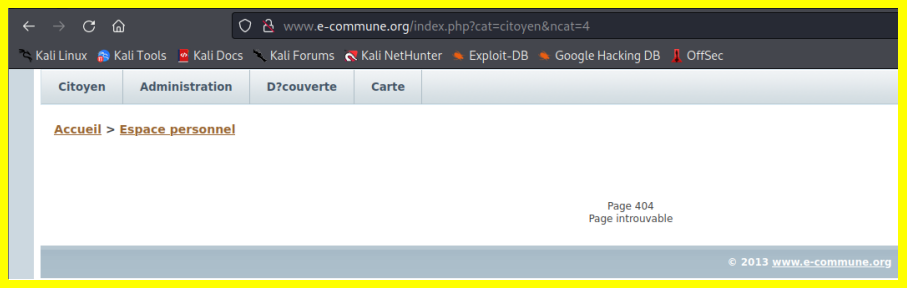
Exploitation:

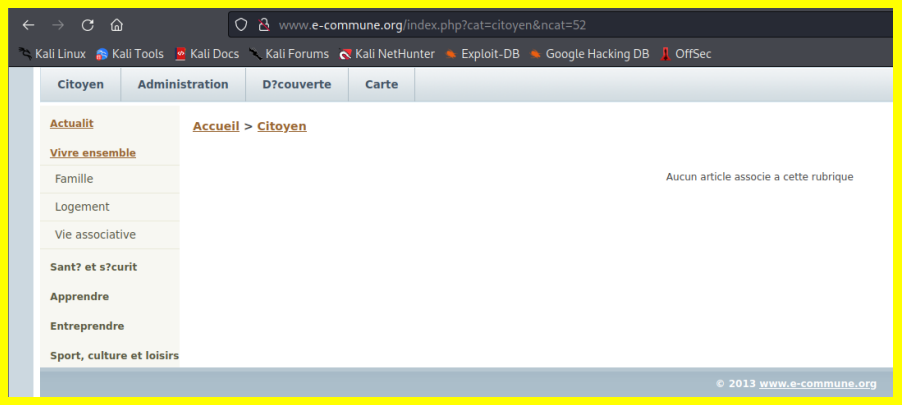
Navigate to the below page and observe the URL



There is a value specified after ncat

Change the number to something else to land in a different page





Remediation:

Users have to be segregated into groups and strict access control has to be implemented on them. The principle of least privilege has to be followed and every user has to be assigned only the utmost required permissions. More on it here and here

12. SQL Injection .

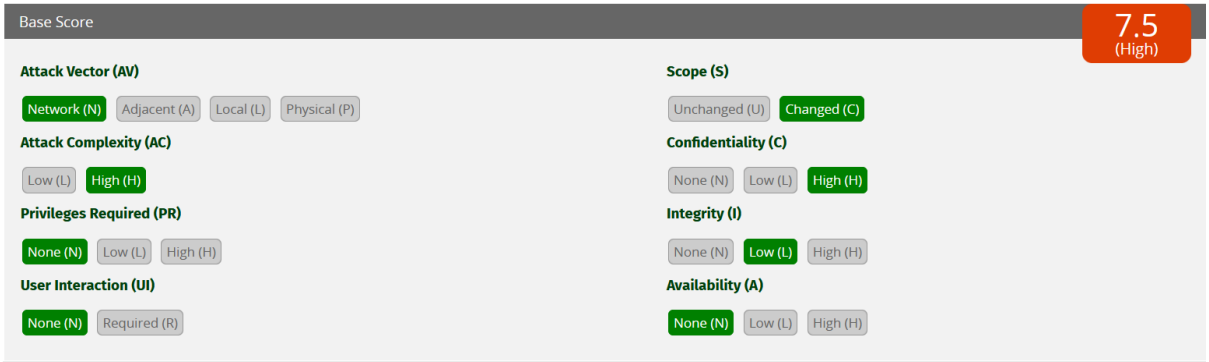
CWE ID : CWE-74,

CVSS Score : HIGH(7.5)

Business Impact Criticality : High

Exploitation Difficulty : Medium

Remediation Difficulty : High

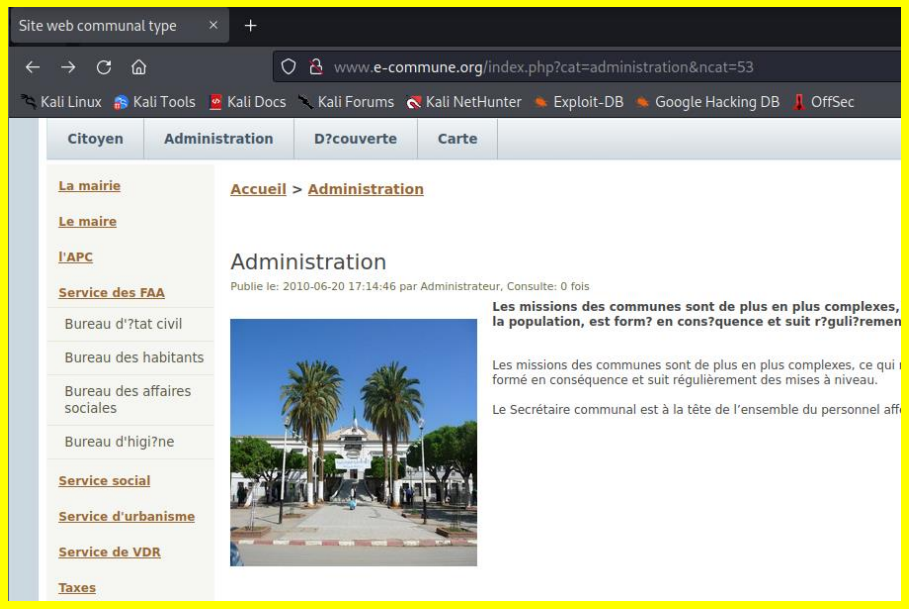


Description:

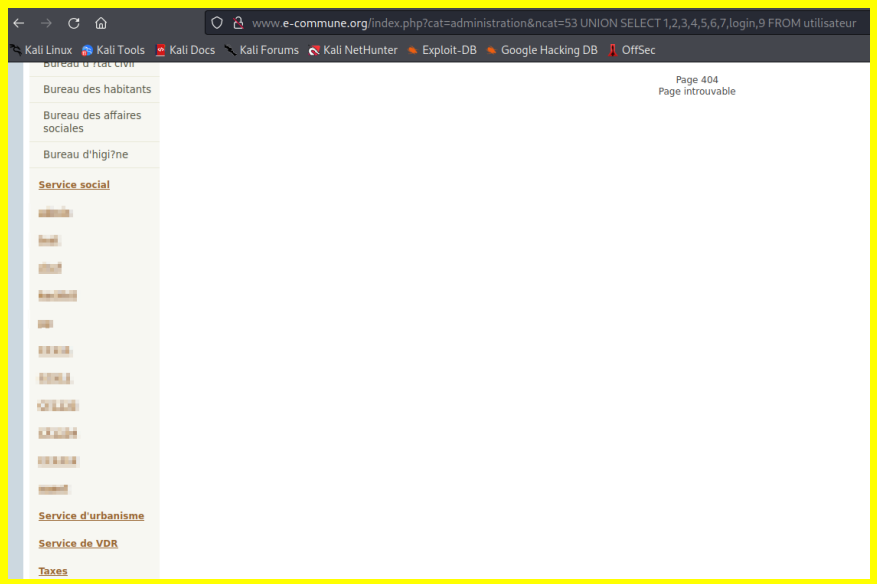
Web Application is susceptible to SQL Injection by entering the query into the URL bar. Leaving the current URL state and giving a union SQL query wont break the page but would retrieve the query and render into the same page of URL before the query.

Exploitation:

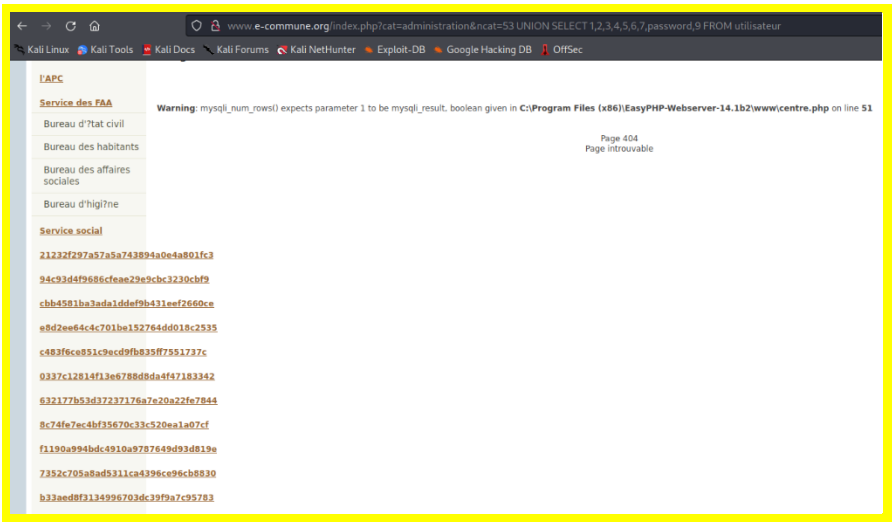
Navigate to any page on the website



Type the intended SQL query into the URL bar as follows



Query to retrieve Login details from the table



Query to retrieve hashed passwords from the table

Remediation:

Prevent sensitive information and querying over the URLs even in HTTPS. They are exposed for users, browser extensions, logged and intercepted etc. They would compromise the design architecture of the application. A redesign of the application is needed with code and API based delivery instead of URL querying. Link.

13. Over Privileged User

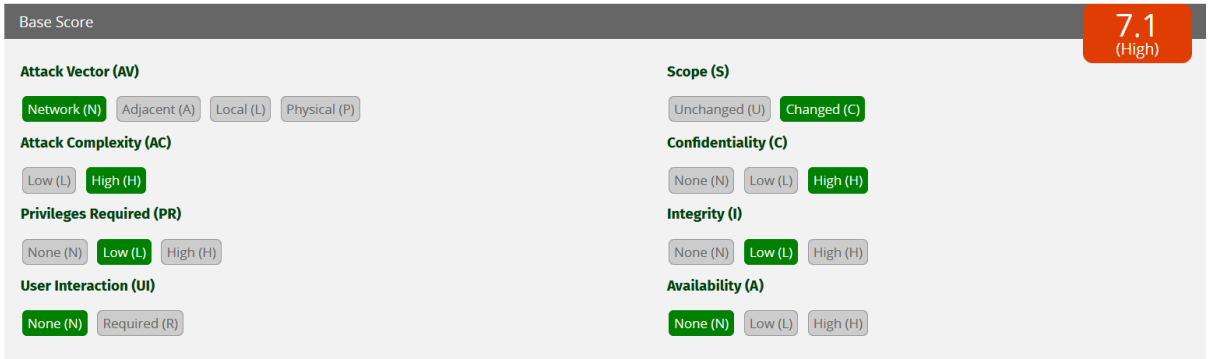
CWE ID : CWE-270

CVSS Score : HIGH(7.1)

Business Impact Criticality : Medium

Exploitation Difficulty : Medium

Remediation Difficulty : Easy



Description:

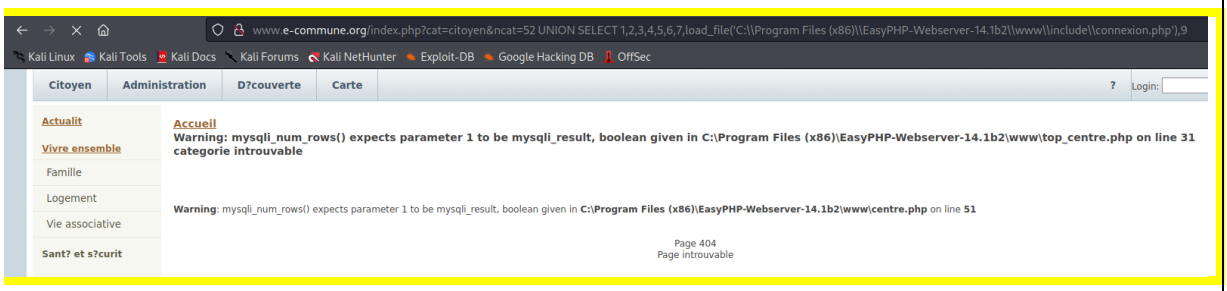
Any attacker is allowed to put SQL queries into the URL bar and retrieve files from the webserver file system. The database user details can be compromised, this user has access to change values from the database as if they were root. An attacker will be able to gain access to any resources that are allowed by the extra privileges even with the ecommune login including, code, disabling services, deleting data, and modification of data.

Exploitation:

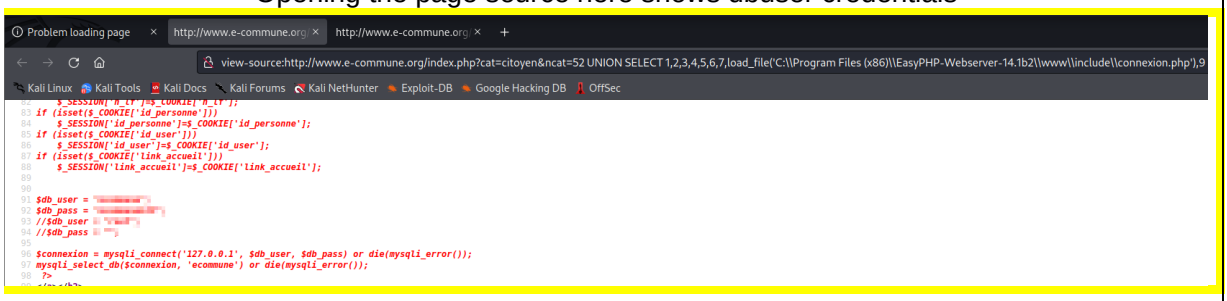
A connection.php can be observed being loaded in the page source

Observe similar SQL queries being made over URL

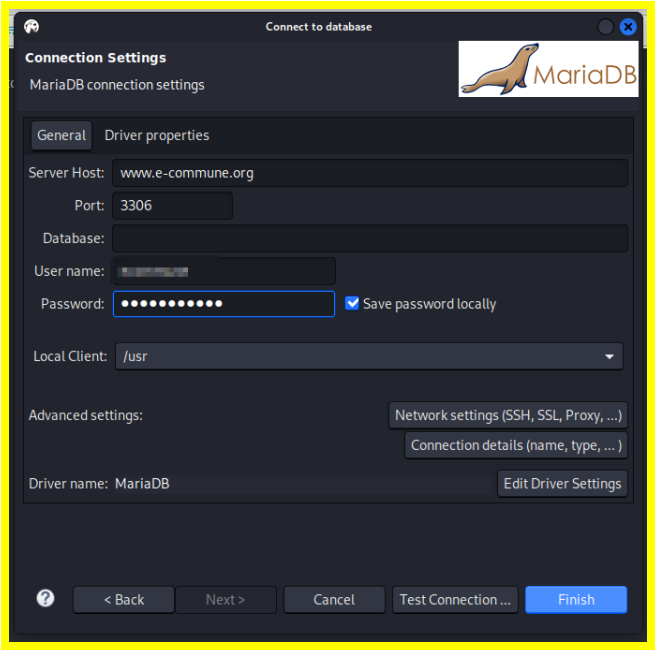
Querying to retrieve above file will result as

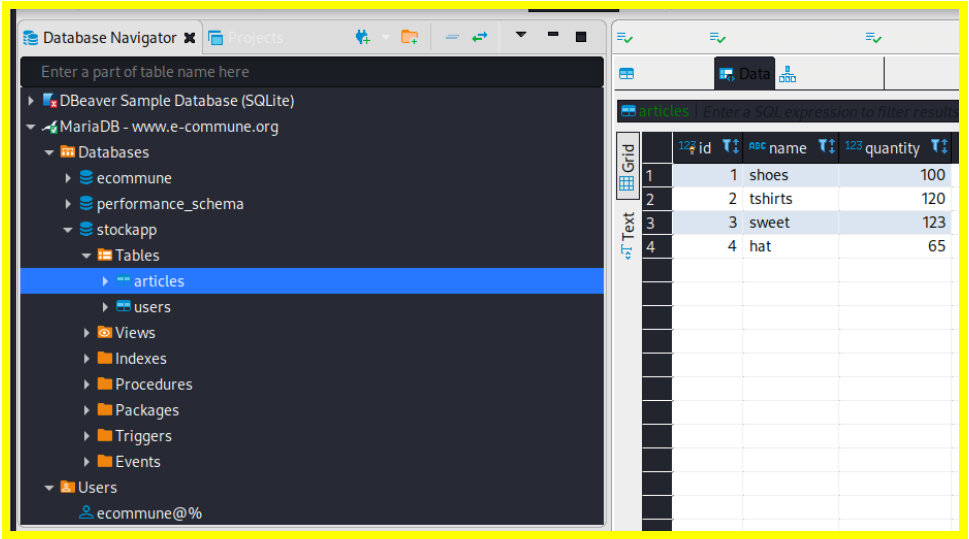


Opening the page source here shows dbuser credentials



Using the above details in DBeaver from any remote location attacker can make any changes.





Remediation:

Isolating users and implementing access management is necessary. Intended access has to be defined and only designated privileged access must be ensured.

14. Cross Site Scripting

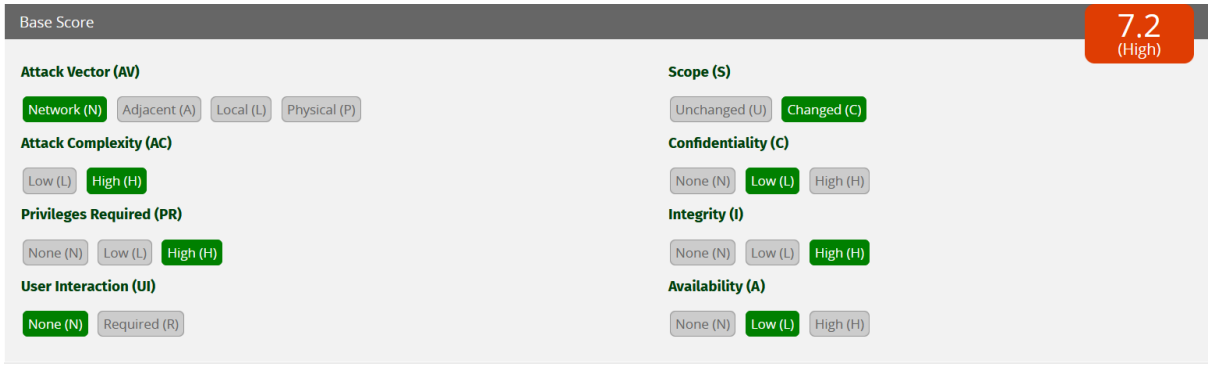
CWE ID : **CWE-79**

CVSS Score : HIGH(8.1)

Business Impact Criticality : High

Exploitation Difficulty : High

Remediation Difficulty : Easy

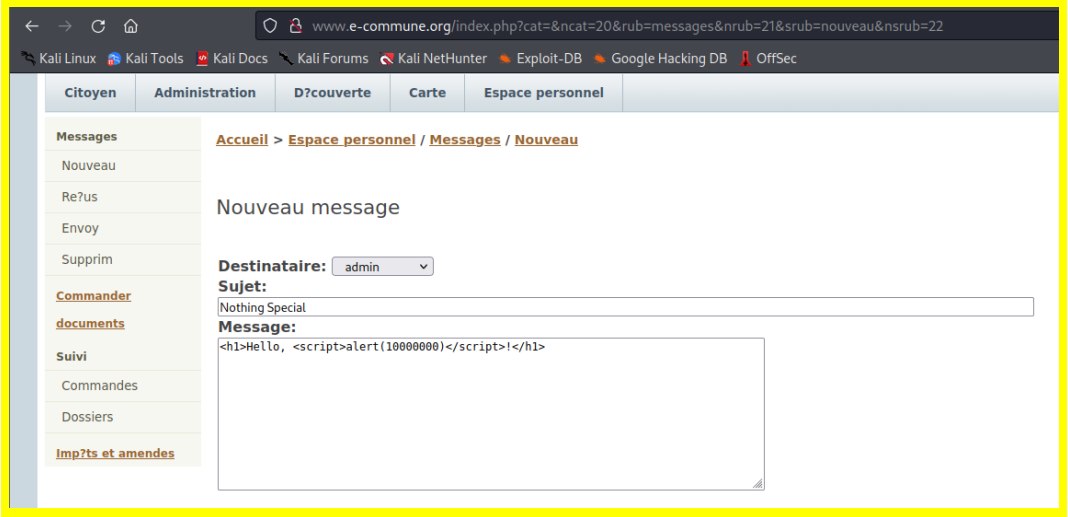


Description:

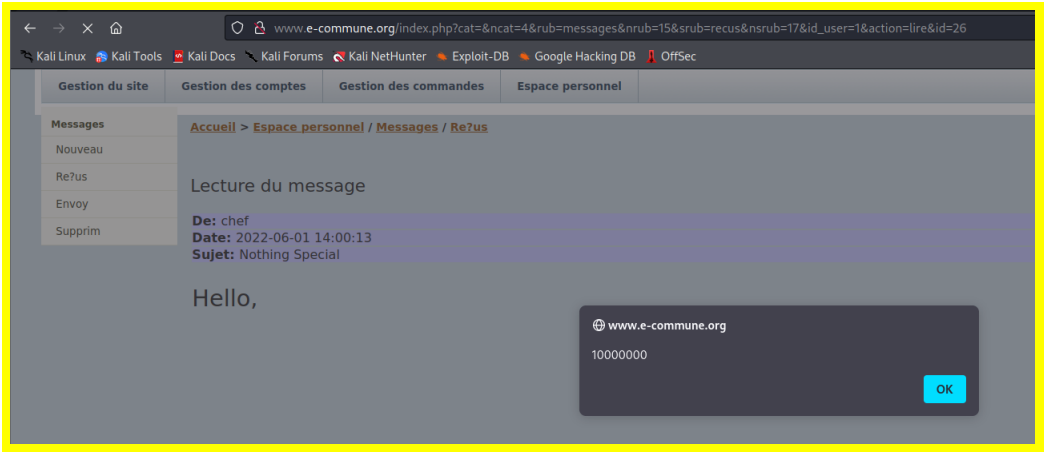
The email built into the application is poorly configured. It can be used to execute scripts with the privileges of the recipient if the recipient were to open the email. Attacker can send emails to the admin user and basically gain admin access to whole application through the scripts he sent to the admin.

Exploitation:

Open the email window and type any script as message



Above script is to display an alert for the recipient



When the recipient opens the email the script gets executed

Remediation:

Cross site scripting can be avoided with proper input validation and output encoding before transmitting data. The text entered should be in email formats XML or HTML and not anything else. There are OWASP libraries to prevent this here.

15. Cookies without HTTPonly Flagset

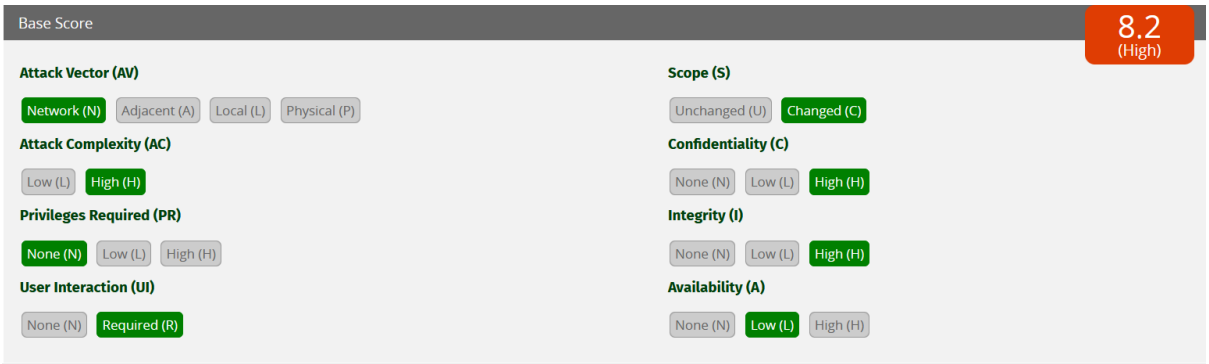
CWE ID : CWE-1004

CVSS Score : HIGH(8.2)

Business Impact Criticality : High

Exploitation Criticality : Hard

Remediation Criticality : Easy



Description:

If the HTTPonly flag isn't set on a cookie then it can be opened and read by client end. This vulnerability compromises information about the server and its users. It eases complexity for the attacker to perform cross-site scripting and injections.

Exploitation:

Using Nikto a vulnerability scanner on the domain

```
┌──(kali㉿kali)-[~]
└─$ nikto -h www.e-commune.org -p 80
- Nikto v2.1.6
---------------------------------------------------------------------------
+ Target IP:          192.168.187.176
+ Target Hostname:    www.e-commune.org
+ Target Port:        80
+ Start Time:         2022-06-01 14:37:30 (GMT-4)
---------------------------------------------------------------------------
+ Server: Apache/2.4.10 (Win32) PHP/5.4.31
+ Retrieved x-powered-by header: PHP/5.4.31
+ The anti-clickjacking X-Frame-Options header is not present.
+ The X-XSS-Protection header is not defined. This header can hint to the user agent to protect against some forms of XSS
+ The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type
+ Cookie PHPSESSID created without the httponly flag
+ RFC-1918 IP address found in the `location` header. The IP is "192.168.189.133".
+ OSVDB-630: The web server may reveal its internal or real IP in the Location header via a request to /images over HTTP/1.0. The value is "192.168.189.133".
+ PHP/5.4.31 appears to be outdated (current is at least 7.2.12). PHP 5.6.33, 7.0.27, 7.1.13, 7.2.1 may also current release for each branch.
+ Apache/2.4.10 appears to be outdated (current is at least Apache/2.4.37). Apache 2.2.34 is the EOL for the 2.x branch.
+ Web Server returns a valid response with junk HTTP methods, this may cause false positives.
+ OSVDB-877: HTTP TRACE method is active, suggesting the host is vulnerable to XST
+ OSVDB-3268: /Admin/: Directory indexing found.
+ OSVDB-3268: /admin/: Directory indexing found.
+ OSVDB-12184: /?=PHPB8B5F2A0-3C92-11d3-A3A9-4C7B08C10000: PHP reveals potentially sensitive information via certain HTTP requests that contain specific QUERY strings.
+ OSVDB-12184: /?=PHPE9568F36-D428-11d2-A769-00AA001ACF42: PHP reveals potentially sensitive information via certain HTTP requests that contain specific QUERY strings.
+ OSVDB-12184: /?=PHPE9568F34-D428-11d2-A769-00AA001ACF42: PHP reveals potentially sensitive information via certain HTTP requests that contain specific QUERY strings.
+ OSVDB-12184: /?=PHPE9568F35-D428-11d2-A769-00AA001ACF42: PHP reveals potentially sensitive information via certain HTTP requests that contain specific QUERY strings.
+ OSVDB-3092: /admin/: This might be interesting...
+ OSVDB-3268: /backup/: Directory indexing found.
+ OSVDB-3092: /backup/: This might be interesting...
+ OSVDB-3268: /img/: Directory indexing found.
+ OSVDB-3092: /img/: This might be interesting...
+ OSVDB-3268: /images/: Directory indexing found.
+ OSVDB-3092: /Admin/: This might be interesting...
+ /login.php: Admin login page/section found.
+ 8698 requests: 0 error(s) and 25 item(s) reported on remote host
+ End Time:           2022-06-01 14:38:49 (GMT-4) (79 seconds)
---------------------------------------------------------------------------
+ 1 host(s) tested
```

Observe PHPSESSID Cookie with CookieHttp vulnerability

Using burpe suite we can verify that cookie with a get request

```
Request
Pretty  Raw  Hex  \n
1  GET / HTTP/1.1
2  Host: www.e-commune.org
3  User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:91.0) Gecko/20100101 Firefox/91.0
4  Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
5  Accept-Language: en-US,en;q=0.5
6  Accept-Encoding: gzip, deflate
7  Connection: close
8  Cookie: PHPSESSID=stueqva8sad7crejf4mun570b0
9  Upgrade-Insecure-Requests: 1
10
```

We can read the cookie as follows

```
Response
Pretty  Raw  Hex  Render  \n
1  HTTP/1.1 200 OK
2  Date: Wed, 01 Jun 2022 19:58:48 GMT
3  Server: Apache/2.4.10 (Win32) PHP/5.4.31
4  X-Powered-By: PHP/5.4.31
5  Expires: Thu, 19 Nov 1981 08:52:00 GMT
6  Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
7  Pragma: no-cache
8  Content-Length: 1755
9  Connection: close
10 Content-Type: text/html
11
```

Remediation:

Cookiehttp policy must be set ON by default unless there is explicit need for client side scripts. More about it here.