Vulnerability Analysis Report

for www.e-commune.org

Prepared by:

Bashetty Arun Kumar

MSc 2018 - EPITA

Table of Contents

Executive Summary

This document represents the Vulnerability Analysis Report (VAR) for the website www.e-commune.org as required by the web administrator. This VAR describes the risks associated with the vulnerabilities identified during www.e-commune.org's security assessment.

There are 9 vulnerabilities found during the assessment. Some of them are critical and required immediate actions to secure the website.

We also enclose our recommendations to correct the issue at each vulnerability analysis.

Tools used for assessment:

- Firefox version 52.9.0 (64-bit)
- Wireshark version 2.6.1

Vulnerability 1

NETWORK TRAFFIC NOT ENCRYPTED

CRITICALITY INDEX

Area	Index	Notes
Risk	High	
Exploitation	Medium	
Correction	Medium	

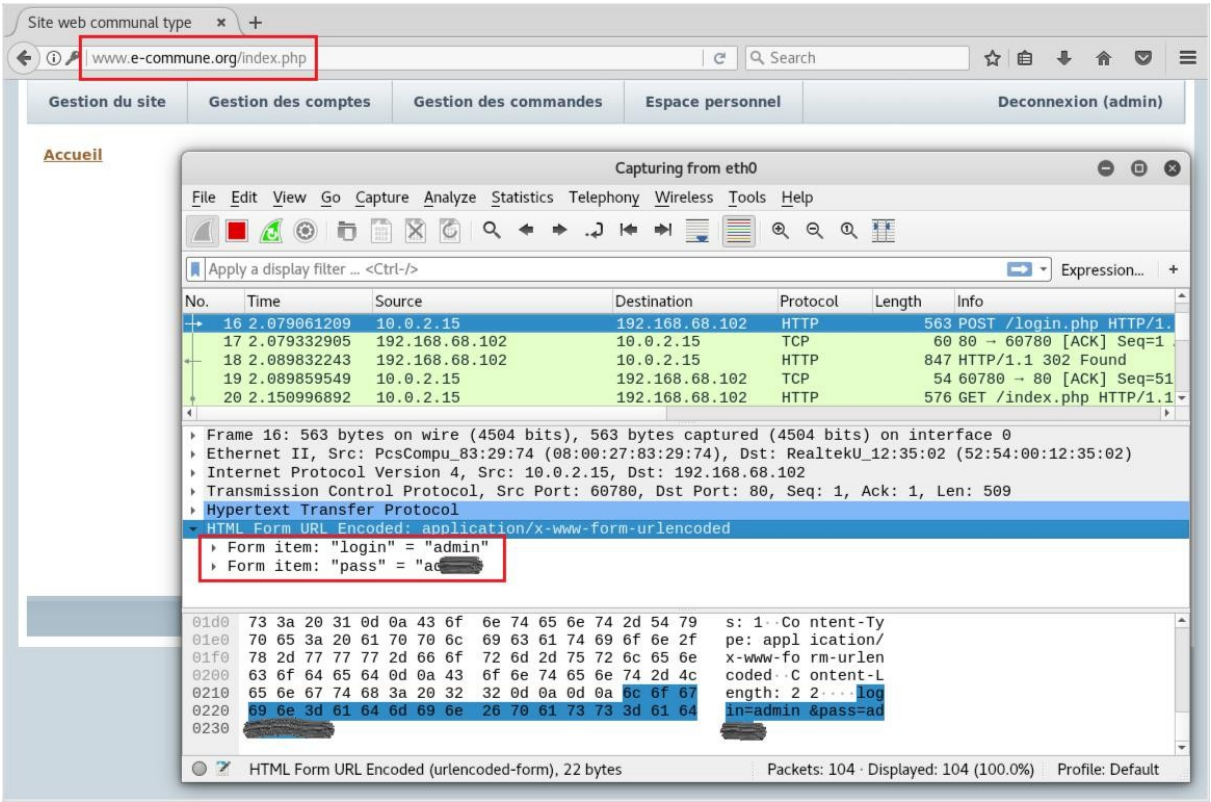
DESCRIPTION

Currently www. e-commune.org is using HTTP protocol instead of HTTPS. It means that exchanged data between user browser and e-commune.org web server is not encrypted. Sensitive data such as user's password or credit card number can be stolen when it is send from the browser to the server.

Furthermore, modern web browser can warn or prevent users to access to www. e-commune.org when unsecured HTTP protocol is being used.

EXPLOITATION

Sensitive data is visible



Recommendation

Switching to HTTPS as soon as possible. The following are the steps that you need to do to correct this vulnerability:

1. Purchase an SSL certificate and a dedicated IP address from your hosting company.
2. Install and configure the SSL certificate.
3. Perform a full back-up of your site in case you need to revert back.
4. Configure any hard internal links within your website, from HTTP to HTTPS.
5. Update any code libraries, such as JavaScript, Ajax and any third-party plugins.
6. Redirect any external links you control to HTTPS, such as directory listings.
7. Update htaccess applications, such as Apache Web Server, LiteSpeed, NGinx Config and your internet services manager function (such as Windows Web Server), to redirect HTTP traffic to HTTPS.
8. If you are using a content delivery network (CDN), update your CDN's SSL settings.
9. Implement 301 redirects on a page-by-page basis.
10. Update any links you use in marketing automation tools; such as email links.
11. Update any landing pages and paid search links.
12. Set up an HTTPS site in Google Search Console and Google Analytics.

For more information, please refer to https://www.keycdn.com/blog/http-to-https

Vulnerability 2

WEAK PASSWORDS ACCEPTED

CRITICALITY INDEX

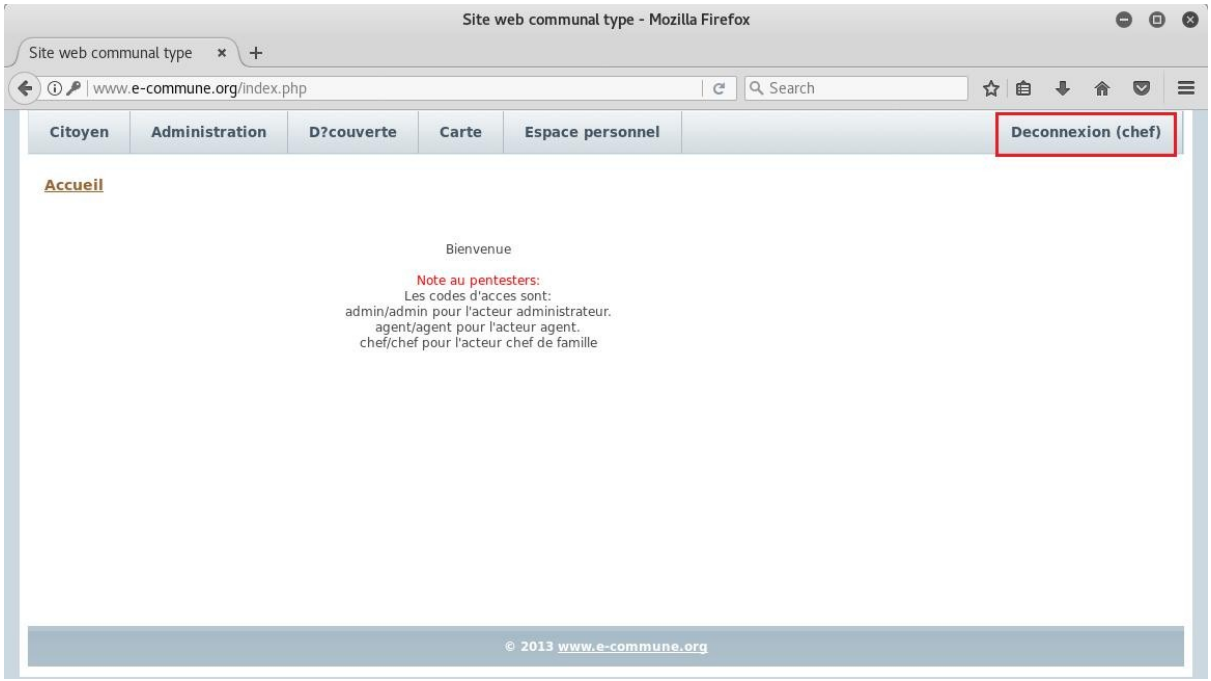
Area	Index	Notes
Risk	High	
Exploitation	Medium to Low	
Correction	Medium	

DESCRIPTION

Currently the website accepts weak user's password such as: short password (less than 5 characters), password matching with user name, password is a plain word. This vulnerability is can be exploit by hacker without any difficulty.

EXPLOITATION

User chef is logined with a weak password, 4-character long only,



RECOMMENDATION

The following are the steps that you need to do to correct this vulnerability:

1. Modify the authentication feature of the website
2. Inform current users to change their password to a strong one

Vulnerability 3

OPEN REDIRECT

CRITICALITY INDEX

Area	Index	Notes
Risk	Medium	
Exploitation	Medium	
Correction	Medium	

DESCRIPTION

Currently the website use redirect method to return back to default screen when user fail to login. This vulnerability can be exploit by hacker to do phishing attack to e-commune.org users. The victim can receive an email that looks legitimate with a link that points to a correct and expected domain such as www.e-commune.org. What the victim may not notice, is that in a middle of a long URL there are parameters that manipulate and change where the link will take them. Hacker can build a fake website and steal victim password.

EXPLOITATION

Redirect link after a fail login:

RECOMMENDATION

1. To correct this vulnerability, you need to replace the redirect link at “Retour a l’accueil” with a go back button (<button onclick="goBack()">Retour a l’accueil</button>).
2. Review all other links in the website to ensure no other open redirect to www.e-commune.org.

Vulnerability 4

DIRECTORY LISTING

CRITICALITY INDEX

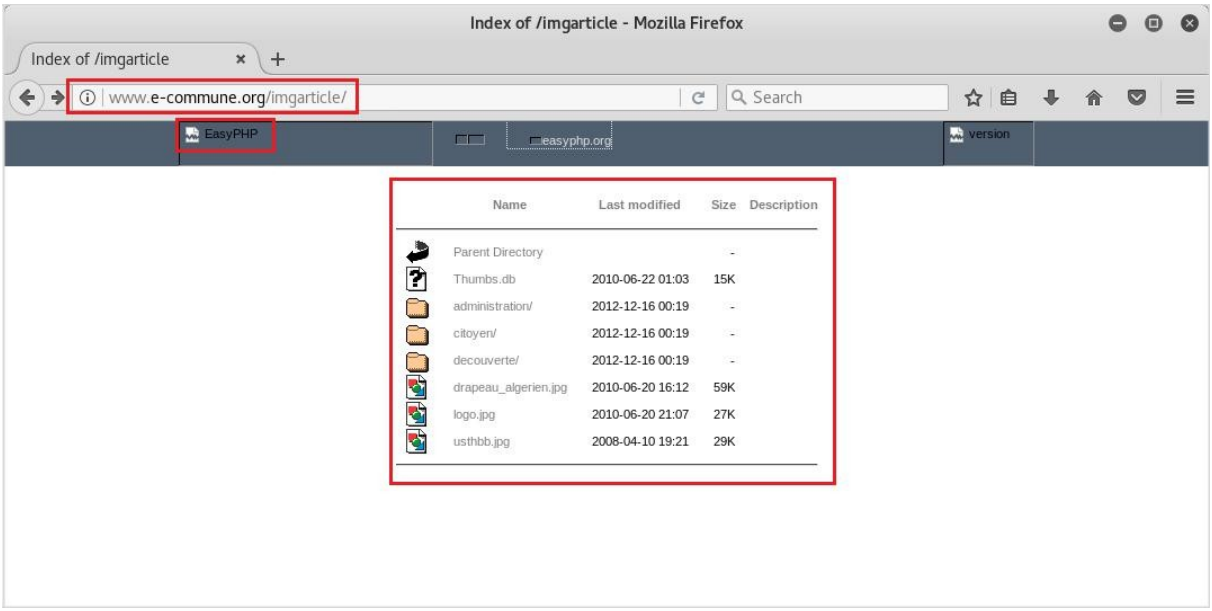
Area	Index	Notes
Risk	Medium	
Exploitation	Medium	
Correction	Medium	

DESCRIPTION

Currently the website is enable for directory listing. It means that public user can view more information that is not shown in web pages. Sensitive information such as backup files, password files, database files, FTP logs and PHP scripts if incidentally left there can be stolen by others.

EXPLOITATION

The website is enable directory listing when visiting http://www.e-commune.org/imgarticle/ with a
CMS name (EasyPHP):



RECOMMENDATION

In order to correct this vulnerability, you need to:

1. Configure your web server to prevent directory listings for all paths beneath the web root;
2. Place into each directory a default file (such as index.htm) that the web server will display instead of returning a directory listing.

Vulnerability 5

TECHNICAL INFORMATION DISCLOSURE

CRITICALITY INDEX

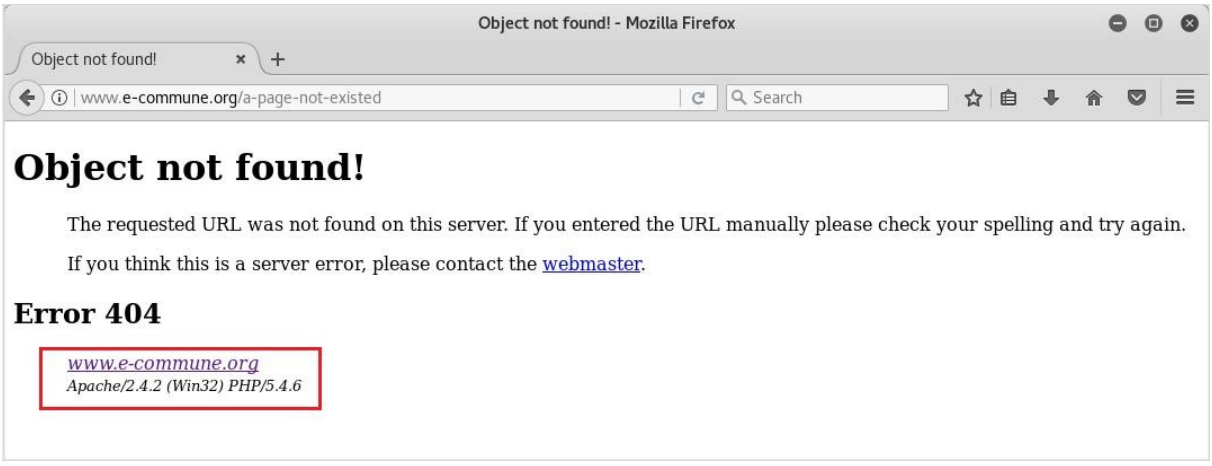
Area	Index	Notes
Risk	Medium	
Exploitation	Medium	
Correction	Medium	

DESCRIPTION

Currently the website is displaying error message with technical information that is not needed by normal users but can be benefit to hackers. With those technical information, the hackers can search for vulnerabilities currently existing in e-commune.org website (operating system, web server software, programing language) and attack the site.

EXPLOITATION

Detail technical information (Apache 2.4.2 and PHP 5.4.6) is disclosed when visiting an unknown page in www.e-commune.org. The situation can be worst together with the CMS name shown in vulnerability 4.



RECOMMENDATION

In order to correct this vulnerability, you need to:

1. Develop an error page to communicate error with users instead of showing technical detail.
2. Hide all PHP errors with an extra script:

```
error_reporting(0);
ini_set('display_errors', 0);
```

Vulnerability 6

POSSIBLE TO VERIFY LOGIN ID

CRITICALITY INDEX

Area	Index	Notes
Risk	Medium	
Exploitation	High	
Correction	Medium	

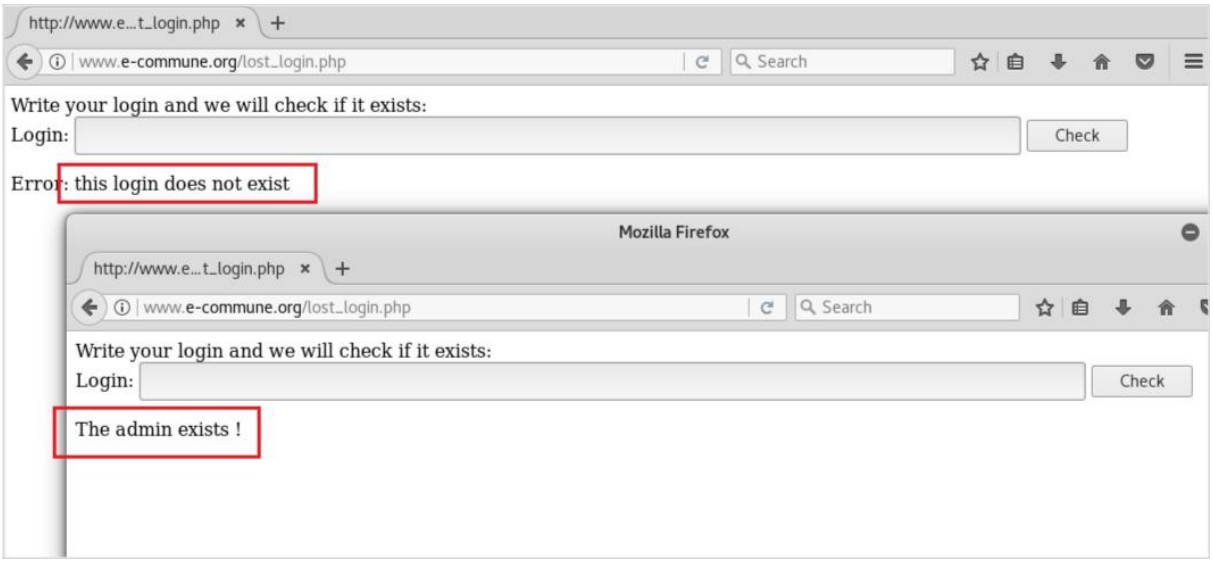
DESCRIPTION

The website currently allows public user to check if a login id exist or not. There is no limitation in number of trials. Hackers can exploit this feature to confirm some important login id such as “admin” then use the found id with other technic to hack the website. We need to remove this feature or limit the usage to administration group only.

Public user can retrieve his/her forgotten login id by type in their email address. The website then will email them their login id.

EXPLOITATION

At the “lost_login” screen, we can type any login id to test the feature, as many times as we want.



RECOMMENDATION

In order to correct this vulnerability, you need to:

1. Remove this feature from public user. Only user of administration group can view and use this feature.

2. Implement a new “login id search” by allow user to key in their email address and the website will email them their login id if any.

Vulnerability 7

LOGIN/PASSWORD BRUTE-FORCE POSSIBLE

CRITICALITY INDEX

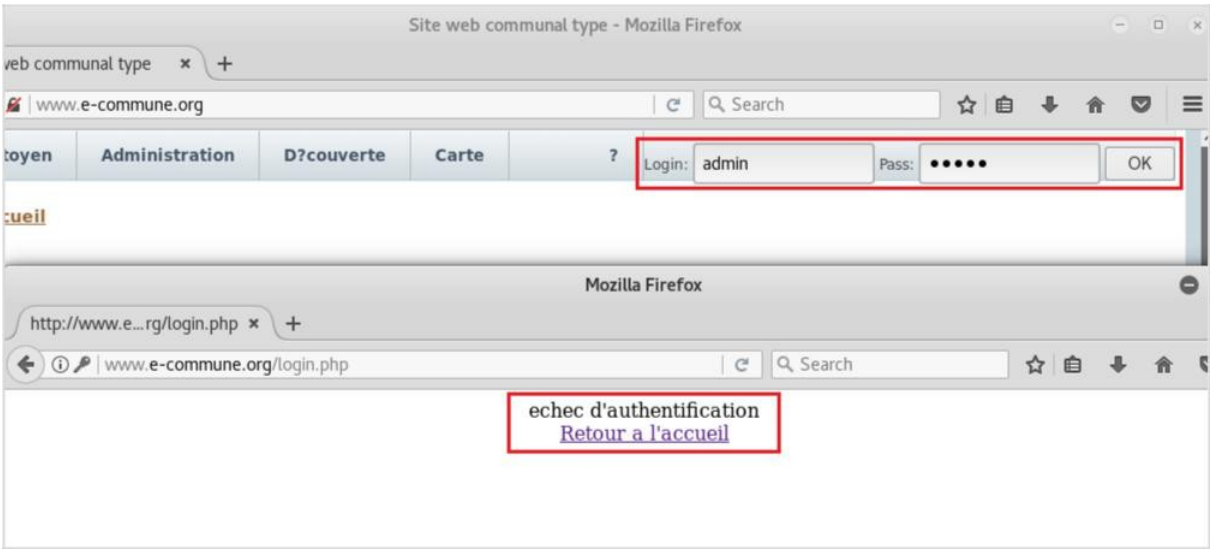
Area	Index	Notes
Risk	High	
Exploitation	Medium	
Correction	Medium	

DESCRIPTION

Currently the website allow user try to login into the system with unlimited fail attempts. Hackers can exploit this feature by running a brute-force program (try various combinations of usernames and passwords again and again until it gets in) to login into the website. It will be much easier for them when our website accepts weak password (vulnerability 2) and let them verify login id of admin (vulnerability 6).

EXPLOITATION

There is no bot prevention at the login page and no limitation for number of fail logins



RECOMMENDATION

In order to correct this vulnerability, you need to:

1. Increase password length and complexity: minimum 10 characters with combination of upper case, lower case, number and special characters.
2. Limit number of login attempts to 5 and should block that IP for 1 hour to stop further attempts being made.
3. Modifying .htaccess file to limit/allow specific IP addresses.

4. Using Captcha to prevent bots from executing automated scripts.
5. Implement two factor authentication that require user to input a pin number that the website has just shared with them separately.

Vulnerability 8

Lack of HTTPOnly and Secure Flags for Cookies

Criticality Index

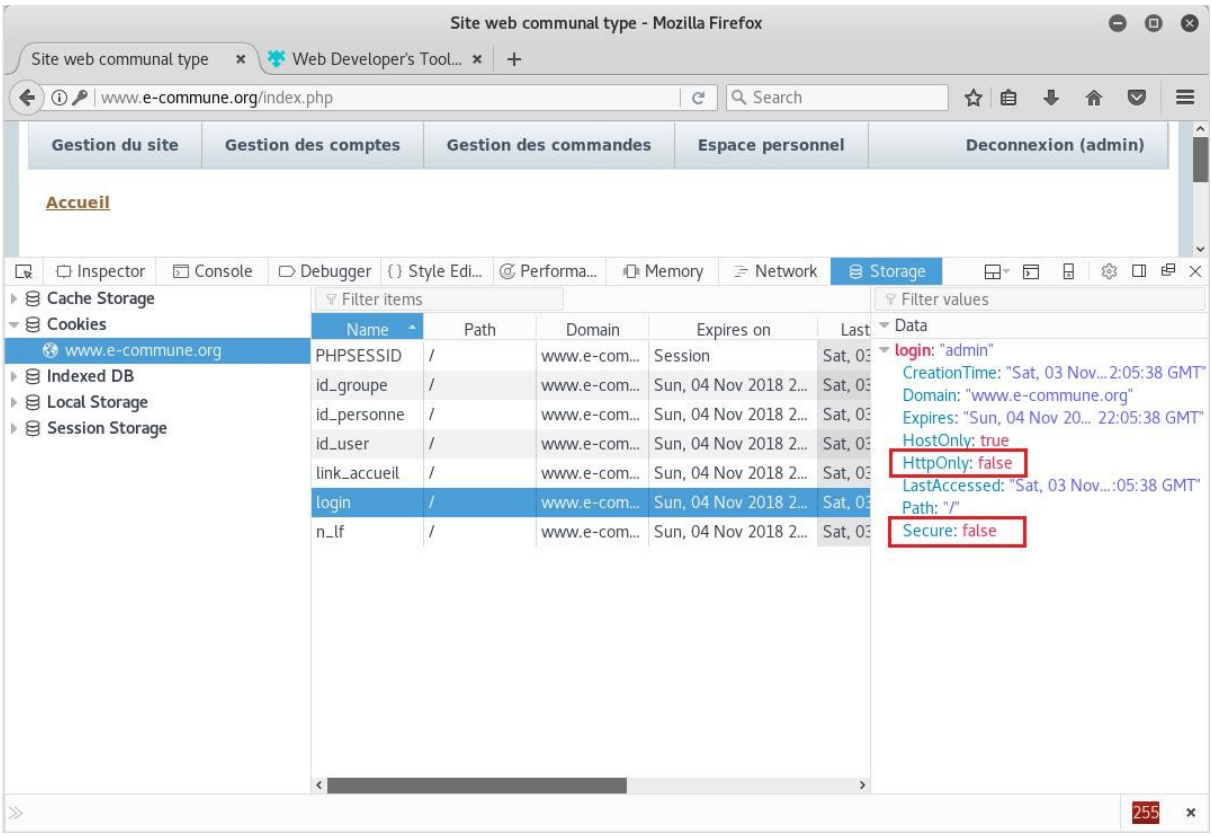
Area	Index	Notes
Risk	Medium	
Exploitation	Medium	
Correction	Medium	

Description

Currently the website allows data transferred in plain text under HTTP protocol. An attacker can send a website link to our users. When a user clicks the link and the HTTP request is generated. Since HTTP traffic is sent in plaintext, the attacker eavesdrops on the communication channel and reads the authentication cookie of the user then he can impersonate the user.

Exploitation

The vulnerability can be find under “Developer Tools” viewer, at tab Storage:



Recommendation

In order to correct this vulnerability, you need to:

1. Turn on the HttpOnly flag
2. Switch to HTTPS as described in the recommendation of vulnerability 1

Vulnerability 9

CAN READ EMAIL OF OTHER USER

CRITICALITY INDEX

Area	Index	Notes
Risk	High	
Exploitation	High	Easy to be exploited
Correction	Medium	

DESCRIPTION

The website currently does not check user authority when reading an email box. It merely bases on browser request parameter to return mail box information. A smart user can read emails of any other users by his login.

EXPLOITATION

By changing “id_user=1” in the web browser URL bar, the agent user can read email of the admin.

RECOMMENDATION

In order to correct this vulnerability, you need to:

1. Implement an authorization method where user id can be easily verified at any request and any time. JWT is an example. Refer to https://auth0.com/docs/jwt for more information.
2. Verify user’s role and return the result accordingly. Do not trust user_id from the request parameters.