PenTesting

Group User Blog

2016

Ishant Upadhyay

Kalina Petrova

Jiaqi Ni

Mengchuan Liu

Dolores Lin, Jianfei Feng

[Jianfei Feng](https://www.facebook.com/jianfei.feng.94?fref=pb&hc_location=profile_browser)

1/1/2016

Contents

[Pen-testing a web application 3](#_Toc464420800)

[Introduction 3](#_Toc464420801)

[ Portscan server 4](#_Toc464420802)

[ Forced Browsing 5](#_Toc464420803)

[ Spider web application 7](#_Toc464420804)

[ Testing for vulnerabilities 8](#_Toc464420805)

[1. Testing for SQL injection 8](#_Toc464420806)

[2. Testing for XSS 9](#_Toc464420807)

[3. Testing CSRF / XSRF 10](#_Toc464420808)

[4. Testing for File Inclusion 12](#_Toc464420809)

[5. File Upload 13](#_Toc464420810)

[Conclusion 15](#_Toc464420811)

# Pen-testing a web application

# Introduction

In order to develop a secure web application, it must go through various testing. In order to know how secure the application is Pen-testing is done. Various tests are done in order to get the vulnerabilities of the application. Some of them are mentioned in this report.

Different tests were performed for the vulnerabilities of the application. Some of them are mentioned below:

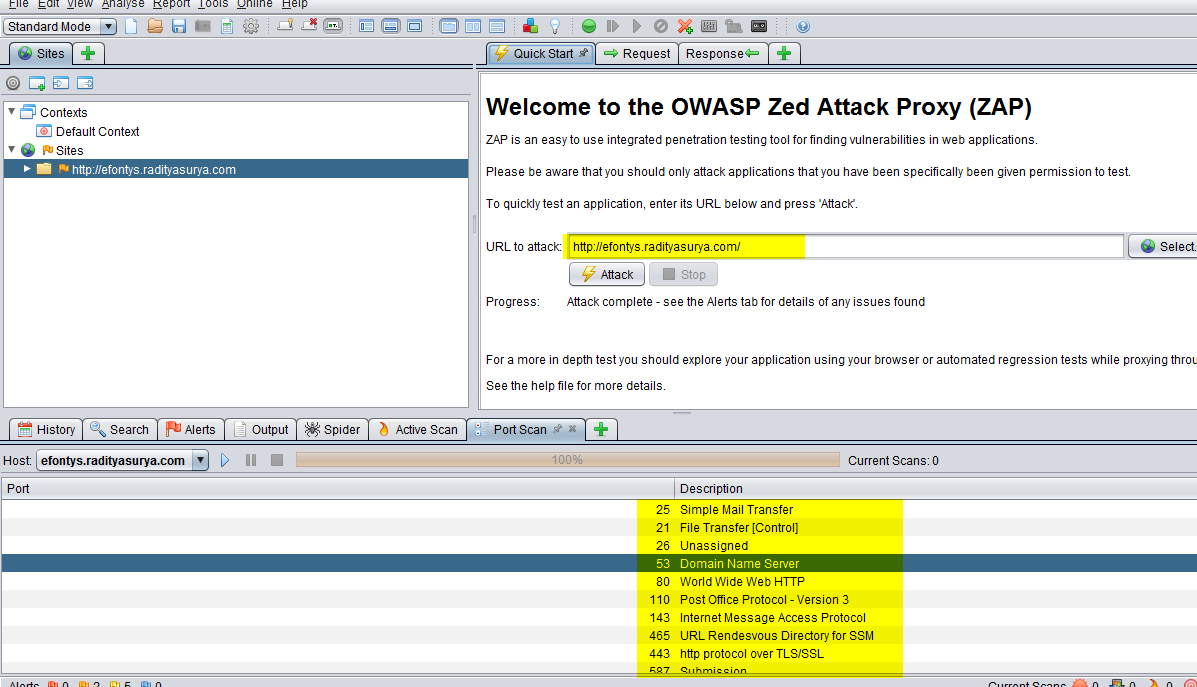
## Portscan server

Port scan are usually used by hacker to find out the open port.

**Tools used**: OWASP ZAP Port Scanner

**Action:** The url of the website http://efontys.radityasurya.com/ is added to the tool and checked for open port.

**Result:** We were able to see all the open port.



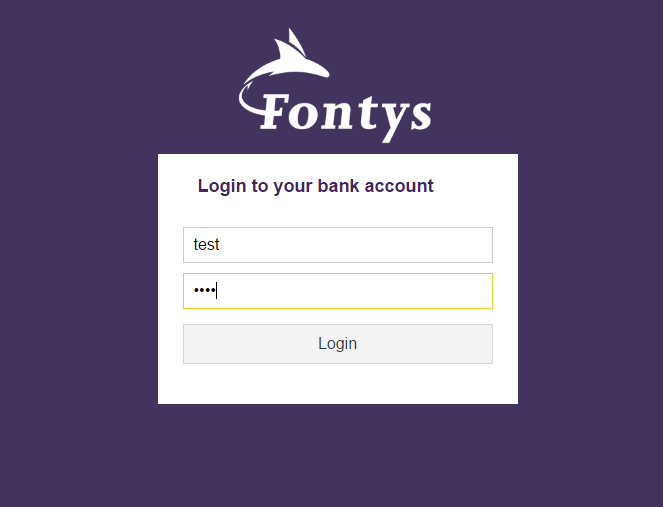
**Conclusion:** You can use some tools to prevent the port scanner.

## Forced Browsing

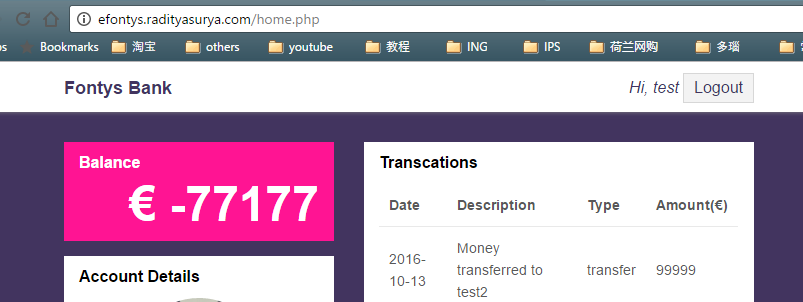
Forced browsing is an attack where the aim is to enumerate and access resources that are not referenced by the application, but are still accessible. Forced browsing is a simple browser attack that attempts to circumvent these controls by requesting authenticated areas of the application directly, without providing valid credentials, or by requesting pages beyond the access level of the logged-in user. If permissions on these pages have not been configured correctly, the pages will be displayed to the unauthorised user.

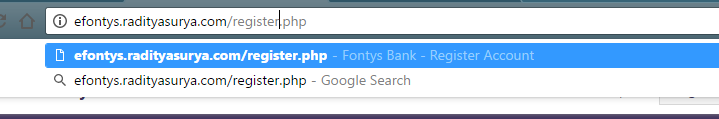
**Action:** The user is logged in with his/her normal account. And changed the url .

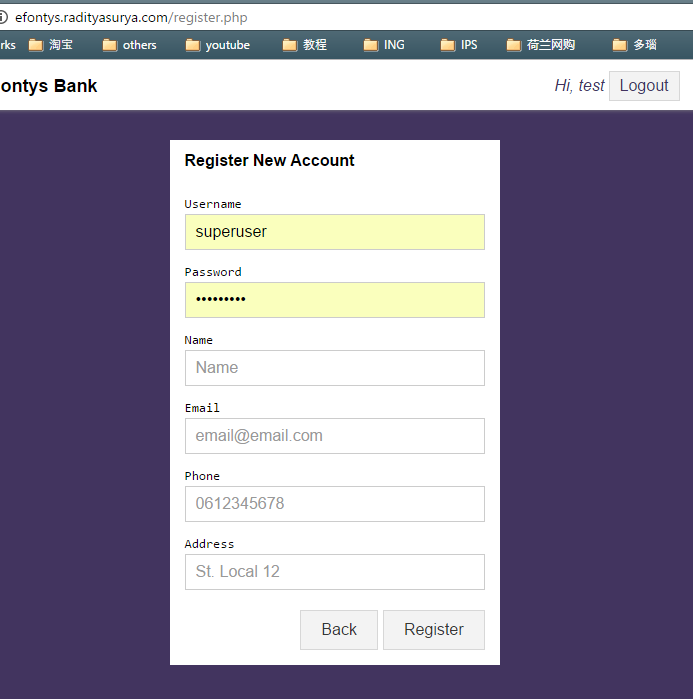
**Result:** We were able to login to register page.



Change the url from “home.php” to “register.php”







**Conclusion**: Now you can add a new account, which you should not be possible to since you are only a normal user not a super user.

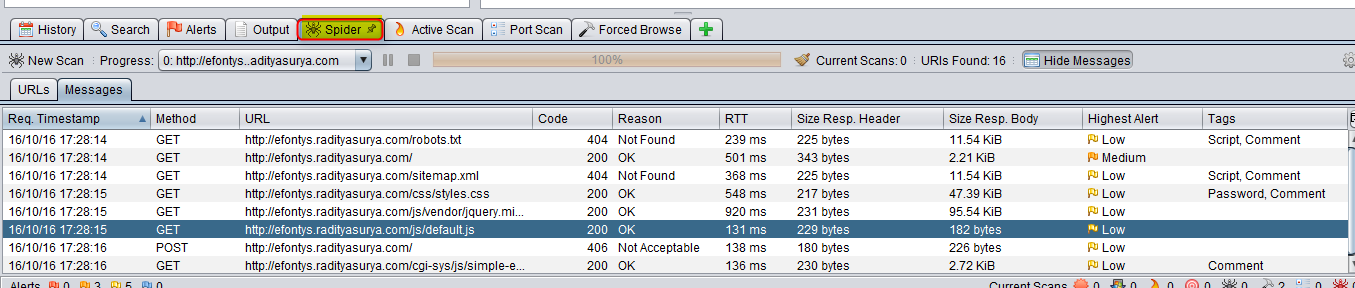
## Spider web application

Spider web application means crawling through the number of pages within a website which are not available for public users.

**Tool used**: OWASP ZAP Spider

**Action:** The url of the website http://efontys.radityasurya.com/ is added to the tool and checked for pages.

**Result:** We can crawl through number of pages.



## Testing for vulnerabilities

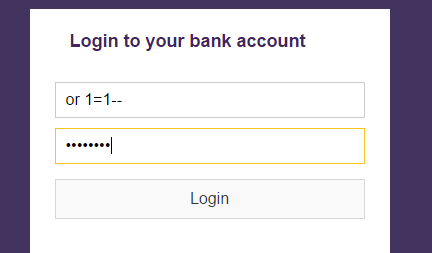
There are different types of injection attack we would use some of them are listed below:

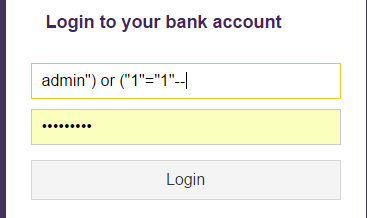
## Testing for SQL injection

A SQL injection attack consists of insertion or "injection" of a SQL query via the input data from the client to the application. A successful SQL injection exploit can read sensitive data from the database, modify database data (Insert/Update/Delete), execute administration operations on the database (such as shutdown the DBMS), recover the content of a given file present on the DBMS file system and in some cases issue commands to the operating system

**Action**: Different type of sql injection were enterd.

**Result:** Cannot access using sql injection.





**Conclusion**: The sql injection is well prevented.

## Testing for XSS

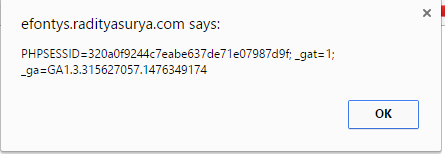
A cross-site scripting vulnerability allows the injection of malicious client-side scripts on a website, which is then served to users (client’s web browser)

**Stored XSS:**

**Action**: Tired to input XSS script transfer recipient name for cookies information.

**Syntax:** Hi <script>alert(document.cookie)</script>

**Result:** We got message cookie informations.

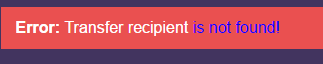


**Conclusion**: We can do XSS cookie stealing.

**Reflected XSS**:

**Action:** We input <font color=”blue”>

**Result:** We can see that the font change from white to blue.



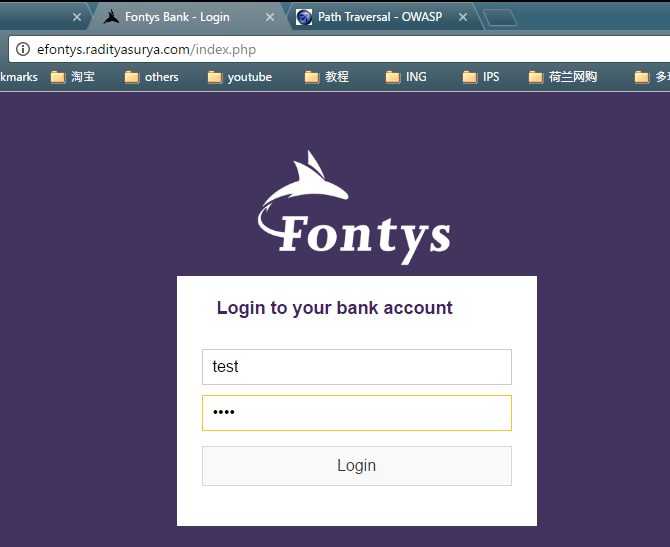
**Conclusion**: We can input script to change color.

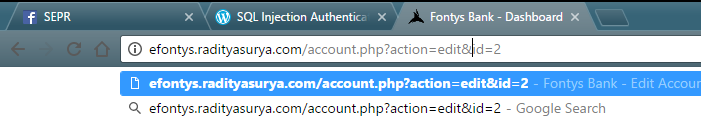
## Testing CSRF / XSRF

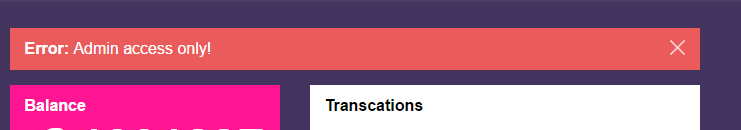
Cross-Site Request Forgery (CSRF) is an attack that tricks the victim into loading a page that contains a malicious request to perform an undesired function on the victim's behalf, like change the victim's e-mail address, home address, or password, or purchase something.

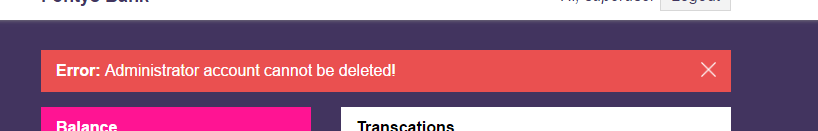
**Action:** We try to modify some account’s information by a normal user, which should only be possible by super account. Login with a normal user and the change the url.

**Result:** Error message was shown.









**Conclusion:** CSRF is also well prevented.

## Testing for File Inclusion

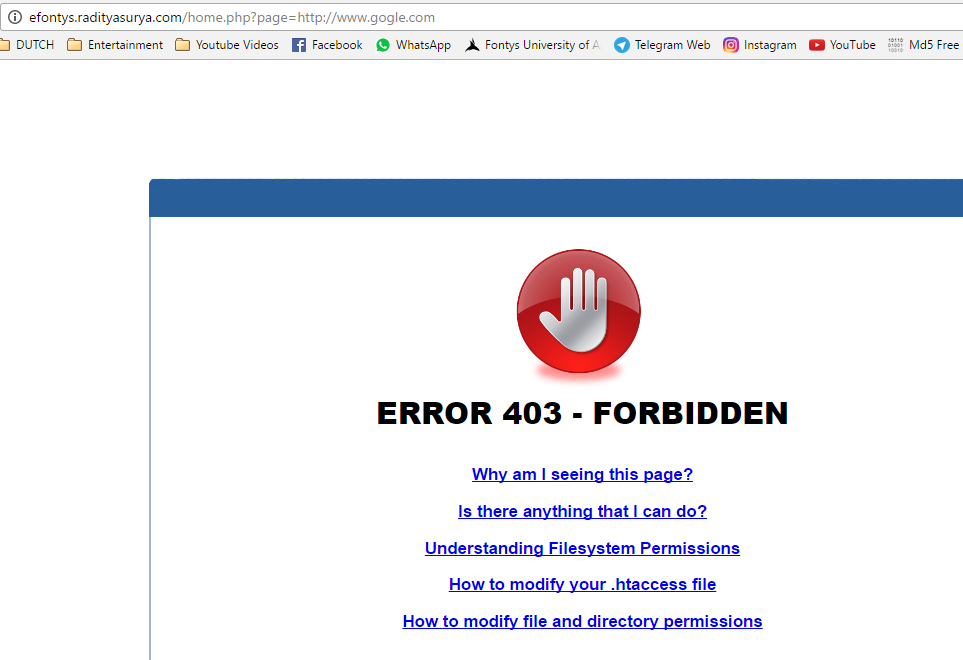
File inclusion vulnerability is a type of vulnerability most often found on websites. It allows an attacker to include a file, usually through a script on the web server. The vulnerability occurs due to the use of user-supplied input without proper validation.

RFI

RFI is the vulnerability found in web site that allows attackers to include remote file on the web server.

**Action:** We tried to change the url and add another url to the old url.

**Result:** Cannot execute another page.



**Conclusion:** Cannot add the Remote file inclusion.

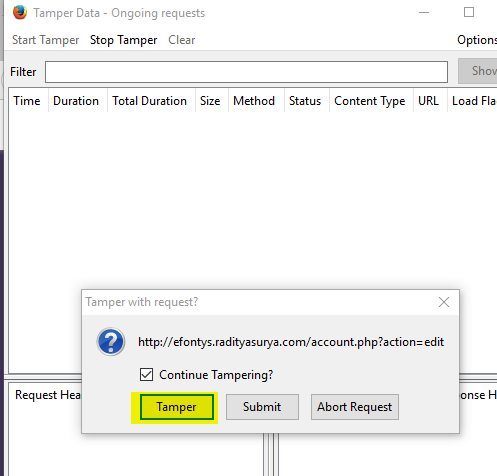
## File Upload

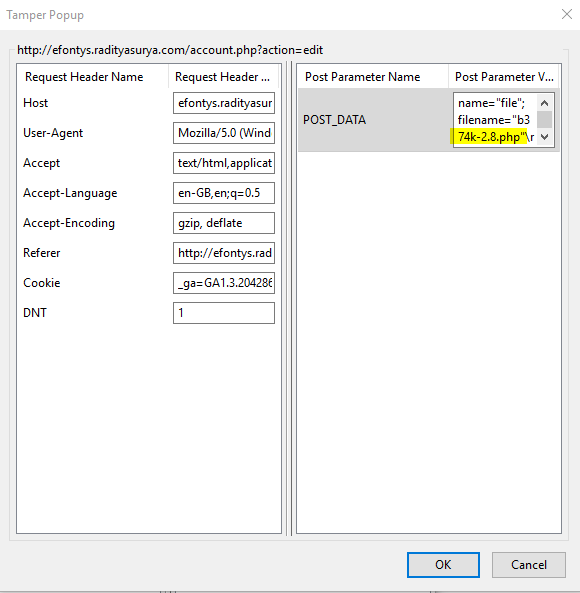
Uploading the malicious file such as shell.php is one the most attack which the attacker will do. We can upload php file by tampering the file extension and upload it the website and can get the whole information about the website database, etc.

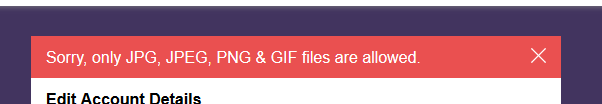
**Tool use**: Tamper Data add on for Mozilla FireFox.

**Action**: We try to upload the shell.php file to the upload image by using tamper data tool.

**Result:** We get error saying the extension only image files are allowed.







**Conclusion**: We cannot insert the php file. It is well secured.

# Conclusion

Overall, the application is secured. Most of the inputs are sanitized, so we cannot input any data except the required format.