CSRF (Cross-Site-Request Forgery)

* This is an Attack Method that user takes place unintended a request.

Thus, BlackHat in the situation that the user can’t acknowledge makes the unintended request through malicious script

Thus, If the user is authenticating, the site hasn’t a distinguishable method about counterfeit request of user and correct request.

CSRF targets the function and changes the server’s state. These have about purchasing or email address or password. Sometimes, vulnerable site have a CSRF. This vulnerability is called “flaw of stored cross site request forgery”

This is possible by just permitting the HTML grammar and using the IMG, IFRAME tag in the complicated CSRF.

URL: <http://localhost/dvwa/vulnerabilities/csrf/>

스크린샷이(가) 표시된 사진

자동 생성된 설명

스크린샷이(가) 표시된 사진

자동 생성된 설명

If I write up about password\_new and password\_conf on the URL field, The variables are received through written a data on the URL field.

Example: <http://localhost/dvwa/vulnerabilities/csrf/?password_new=password&password_conf=password&Change=Change>

스크린샷이(가) 표시된 사진

자동 생성된 설명

스크린샷이(가) 표시된 사진

자동 생성된 설명

As you see, although there are not inputs in the text box, password data is changed. Thus, it is important to block the CSRF Attack.

<Medium level>

스크린샷이(가) 표시된 사진

자동 생성된 설명

스크린샷이(가) 표시된 사진

자동 생성된 설명

<http://localhost/dvwa/vulnerabilities/csrf/?password_new=password&password_conf=password&Change=Change>

ereg function: ereg(“the string you want to find”, “anything”)

// uppercase, lowercase distinguish

eregi function: eregi(“the string you want to find”, “anything”)

// uppercase, lowercase doesn’t distinguish

$\_SERVER[ ‘SERVER\_NAME’ ] : Domain Information

$\_SERVER[ ‘HTTP\_REFERER’] : Previous Page address

The Web page checks where the request page comes from. The developer believes that if it matches the current domain, it must be trusted from the web application.

스크린샷이(가) 표시된 사진

자동 생성된 설명

<script>window.open(“<http://localhost/dvwa/vulnerabilities/csrf/>?password\_new=hacked&password\_conf=hacked&Change=Change#”)</script>

<img src="http://localhost/dvwa/vulnerabilities/csrf/?password\_new=hacked&password\_conf=hacked&Change=Change# onerror=alert("change");>

스크린샷이(가) 표시된 사진

자동 생성된 설명



As you see, we can know that the password is changed by XSS Attack. After this work is finished, let’s go to the login page (login.php).

스크린샷이(가) 표시된 사진

자동 생성된 설명

If you enter the password as “hacked”, you can check that the password is changed well.

<High Level>

스크린샷이(가) 표시된 사진

자동 생성된 설명

high.php

Not Found