**CSE 5382-001**

**Secure Programming**

**Spring 2017**

**Programming Assignment : 4**

**Finding, Exploiting, and Fixing Vulnerabilities in Web Apps**

**By: Vidhi Shah[1001150873]**

**Sneha Gundurao[1001231532]**

**List of Exploited Vulnerabilities (Covering 15 of CSE/SANS Top 25):**

1. **SQL Injection :**

* This SQL vulnerability has been exploited on Mutillidae application of Meatsplotable2.

1. ***How to exploit?***

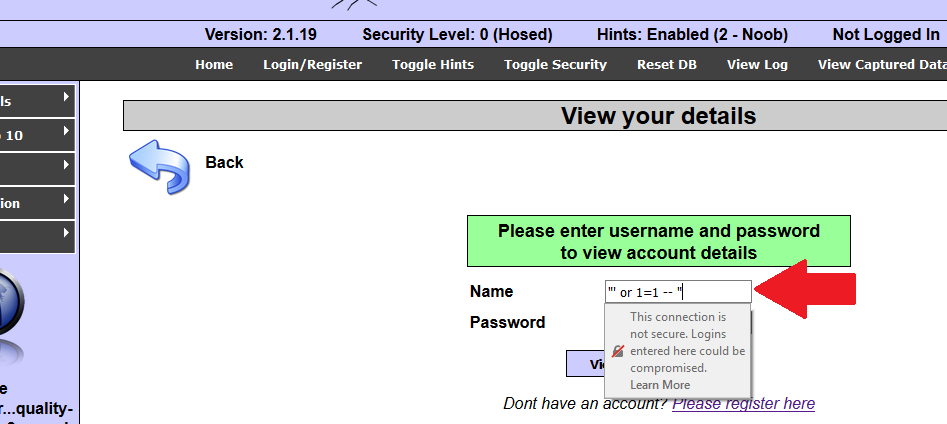
* Click on OWASP Top 10 tab on the left menu, then click A1-Injection, then click SQLi Extract Data, then click User Info. As shown in below image.



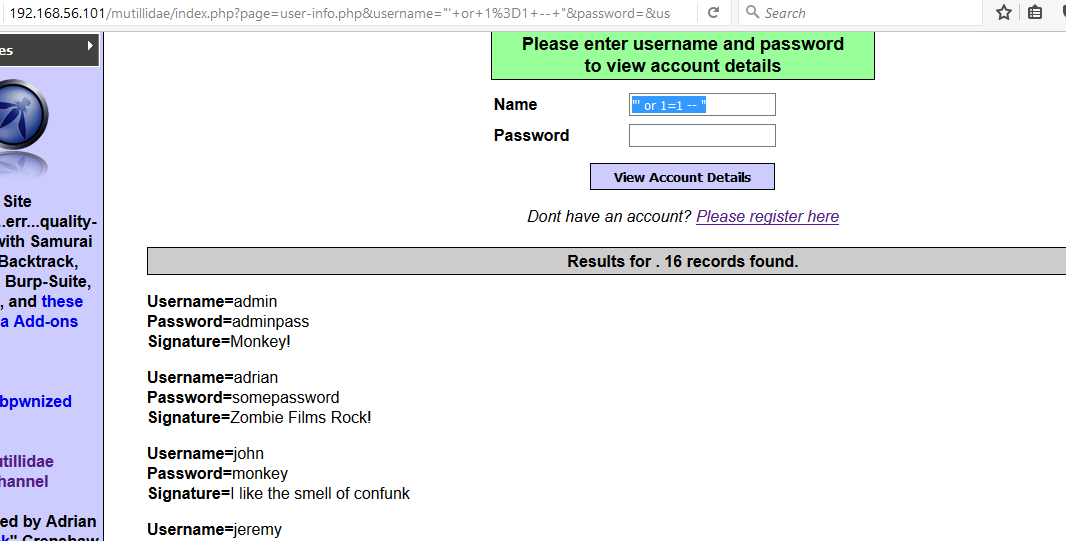
* This page, shows information regarding user upon entering username and password.
* So, we will do most normal sql injection using:

*"' or 1=1 -- "*

* It is shown in the figure below.



* Enter the command in Name field, and Click on view Account Detail, it will show all the user details which are in database.
* Below image shows, the output listing all user details.



1. ***How to fix?***

* Here, before just accepting what user inserts in the field, validation must be done before executing queries.
* Input field validation is must, when any data is coming from user input.
* SQL familiar symbols and namespaces should not be allowed.
* And even check what type of output is generated, if valid than show else fail safely.

1. ***Tools Used?***

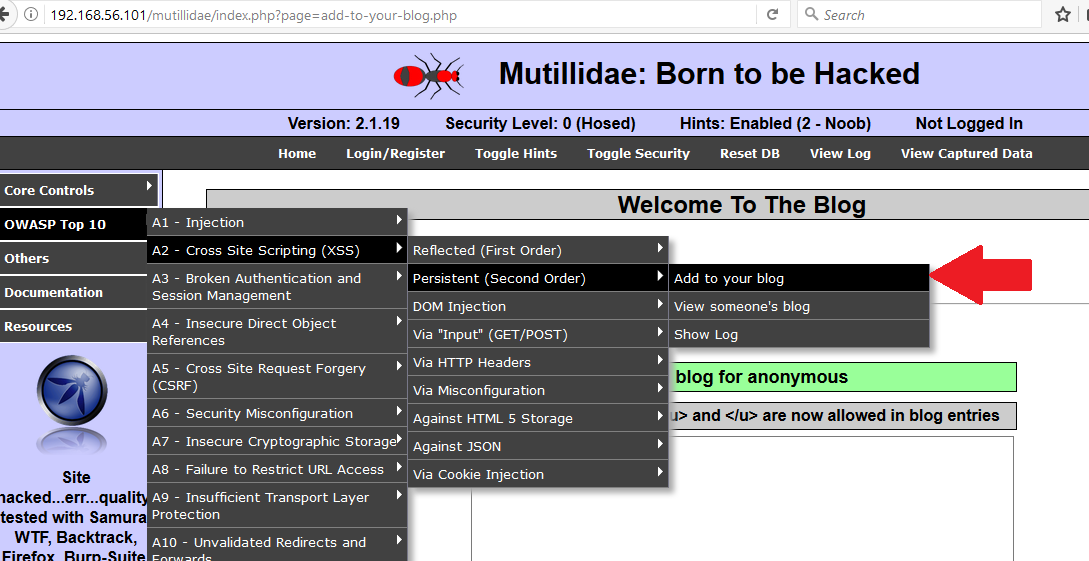
* No extra tool has been used for this exploit.

1. **Cross Site Scripting Injection:**

* This vulnerability has been exploited on Mutillidae application of Meatsplotable2.

1. ***How to exploit?***

* Click on OWASP Top 10 tab on the left menu, then click A2-Cross site scripting(xss), then click Persistent (second order), then click Add to your blog. As shown in below image.



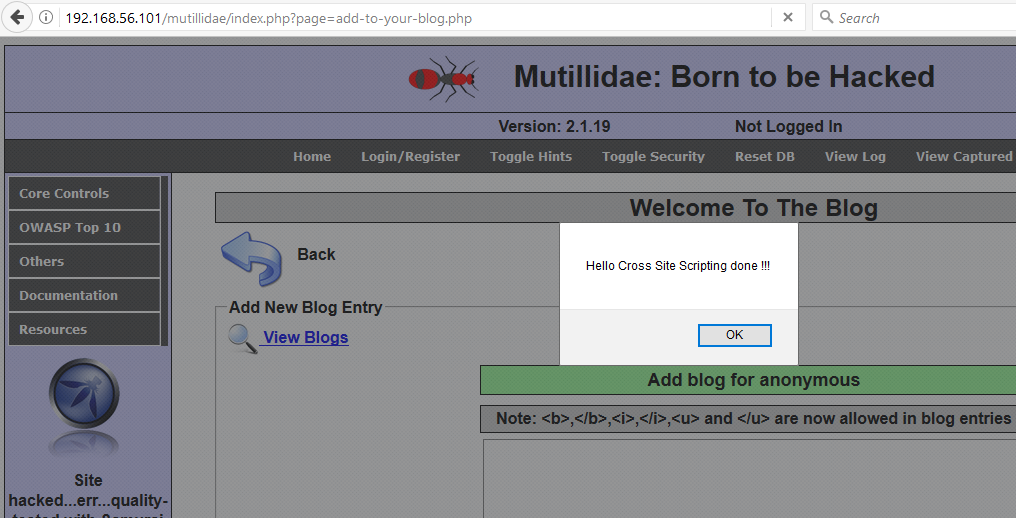
* This page, allows user to write blog and add the content on web page.
* So, we will do most normal Cross Site Scripting injection by using:

*<script>alert("Hello Cross Site Scripting done !!! ")</script>*

* It is shown in the figure below.



* Enter the above given script in the text area field and click Save Blog Entry.
* This will add the content on the text field to the web page, so it will add the scripting in the webpage which will be faced by all users.
* The output result of the above scripting is shown in below image.



1. ***How to fix?***

* Here, before just accepting what user inserts in the field, validation must be done before executing anything.
* Input field validation is must, when any data is coming from user input.
* No HTML or other Markup tags should be allowed.
* To fix/prevent, always escape html before inserting data.
* Even try escaping from the JavaScript tag/html attribute section before inserting data.

1. ***Tools Used?***

* No extra tool has been used for this exploit.

1. **Broken Authentication:**

* This vulnerability has been exploited on Mutillidae application of Meatsplotable2.

1. ***How to exploit?***

* Click on OWASP Top 10 tab on the left menu, then click A3-Broken Authentication, then click Login. As shown in below image.

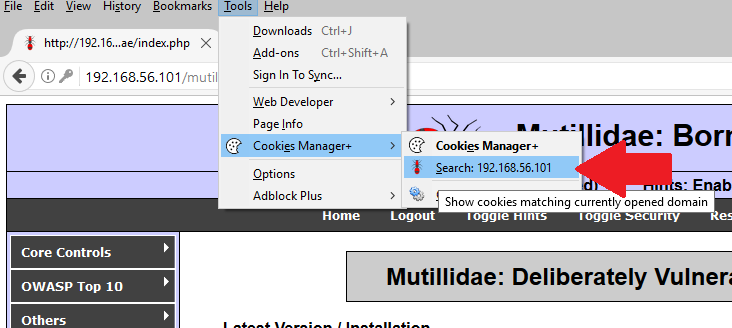


* Before login , you will need to create your account, I have created it with credentials:

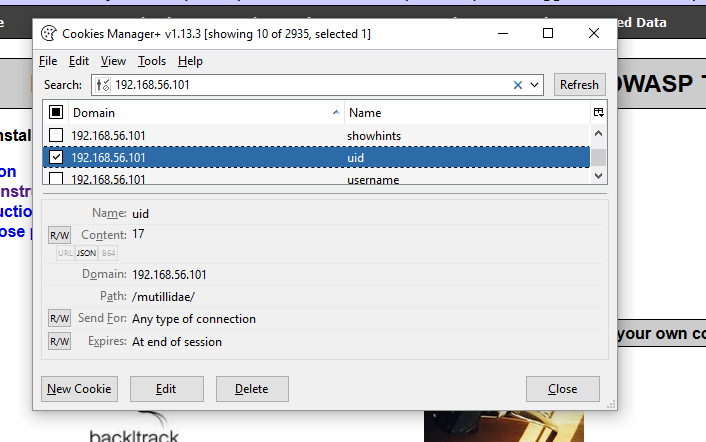
*Name: Vidhi*

*Pass: monkey.*

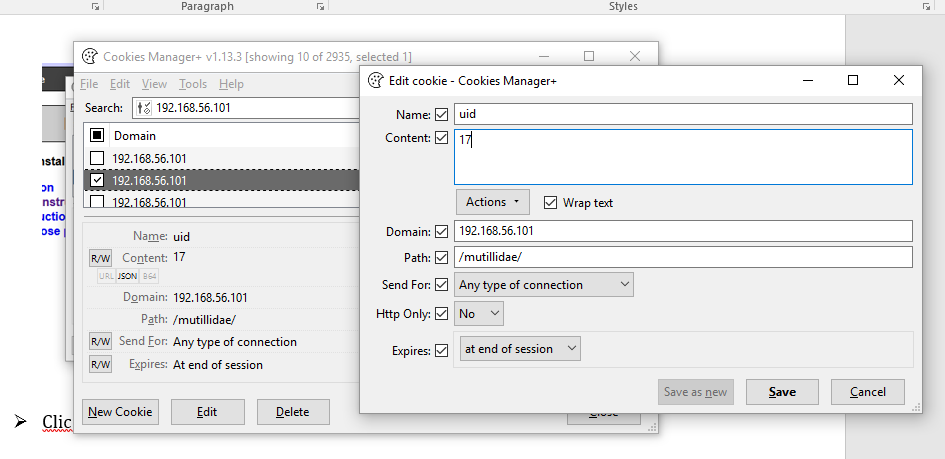
* So, now as guided you go to login page and login with your account.
* Install cookie manager add-on to your web browser. And open cookie manager from menu. As shown in below image.



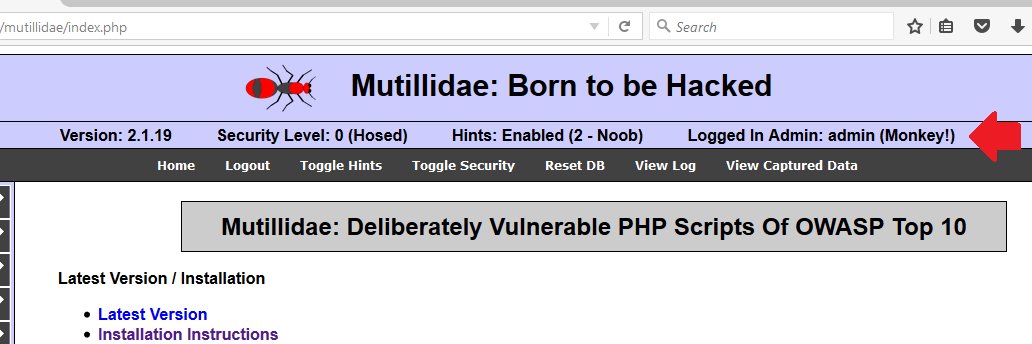
* Find cookie with name uid, this cookie has detail of the logged in user.



* Click on Edit and in the field Content, mine id is 17, I change it to 1 (its admin’s id) and click save.



* Refresh the page, you are now logged in as admin.



1. ***How to fix?***

* Create secure cookie and make essential secure data inside cookie to be encrypted.
* Some session time should be maintained, after that time new cookie should be created for the page and user.
* Important field in cookie, should not be given permission to edit.
* Don’t store user id or such critical data on cookie on user’s web browser. Try using session on server rather than using cookie on user side.

1. ***Tools Used?***

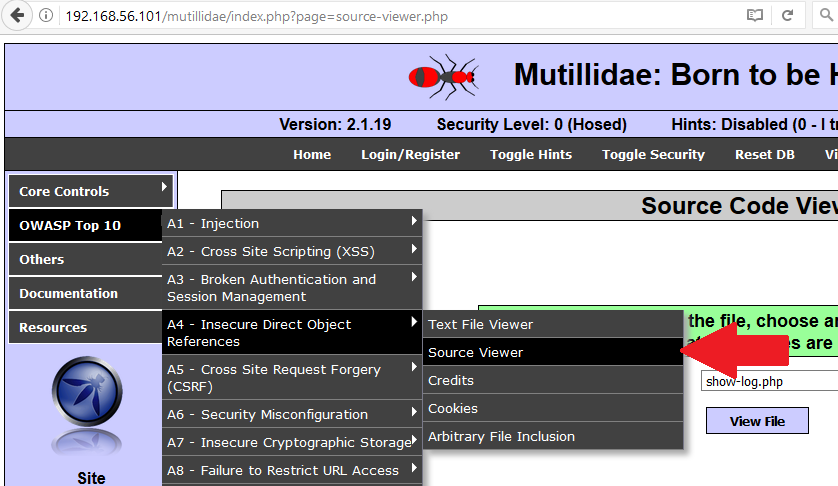
* Cookie manager add-on to edit cookie. Else, No extra tool has been used for this exploit.

1. **Insecure Direct Object Reference:**

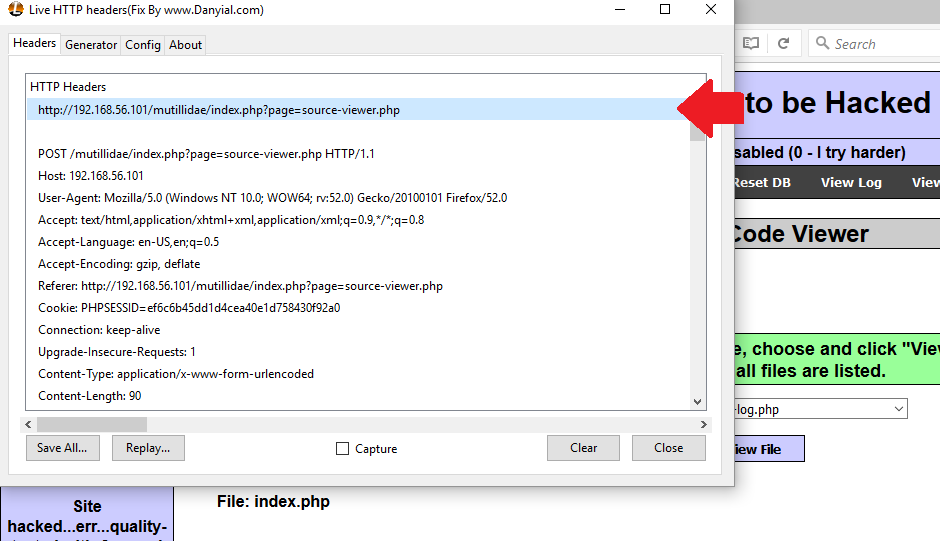
* This vulnerability has been exploited on Mutillidae application of Meatsplotable2.

1. ***How to exploit?***

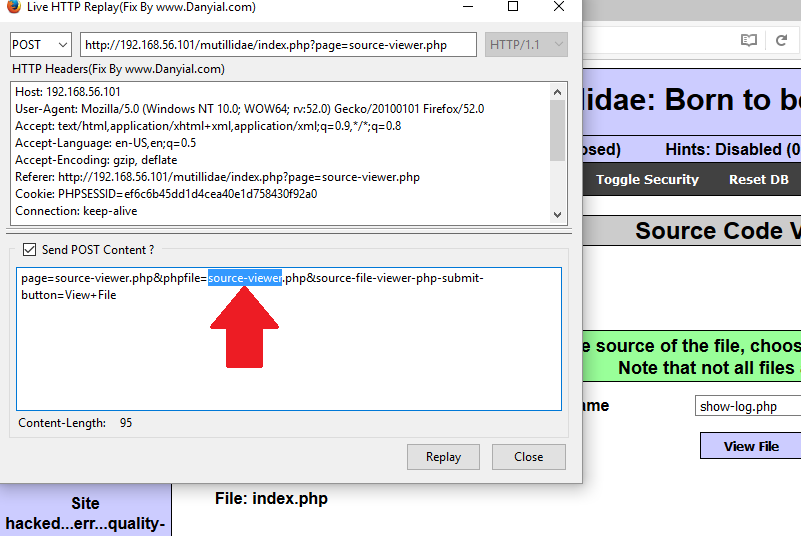
* Click on OWASP Top 10 tab on the left menu, then click A4- Insecure Direct Object Reference, then click Source Viewer. As shown in below image.



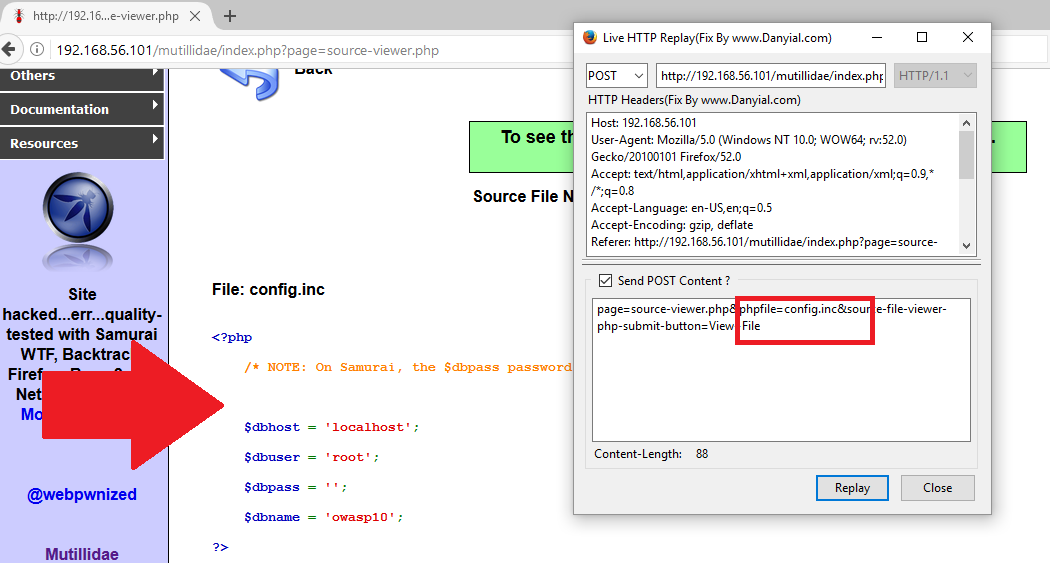
* This page lets s see the source code of the file we select from the dropdown box.
* Install add-on to edit the http header, I have used LiveHTTPHeader for Mozilla web browser.
* As shown in below image, open LiveHTTPHeader, and capture the http header request when trying to click on view file button. Click on the highlighted link in image and then click Replay.



* Now, you will be redirected to the other tab in livehttpheader. Change the post link, As shown the highlighted part, you can change it to any file name(do some brute force to know important file names e.g: config files /index files etc…)



* I have changed it to config.inc file. Which may have some important information. As shown in below image, the config.inc file has databases or any such important data which is now visible to intruder.



1. ***How to fix?***

* Limit the resources to the user according to their privileges or session.
* Check the authorization of user, before they access any resources.

1. ***Tools Used?***

* LiveHttpHeader (header capture) add-on to edit headers and resent requests. Else, No extra tool has been used for this exploit.

1. **Security Misconfiguration and Error Handling:**

* This not exactly vulnerability in web applications . So we don’t exploit this vulnerability, we just do check for existing software in application and they being properly functioning and up to date.

1. ***How to fix?***

* Check if all of the software used in web application are up to date.
* We need to check , on server where our application is running, doesn’t have any malicious web application which may affect our website.
* Remove all unneeded functions from application.
* If error or failure occurs, fail securely without showing technical details.

1. ***Tools Used?***

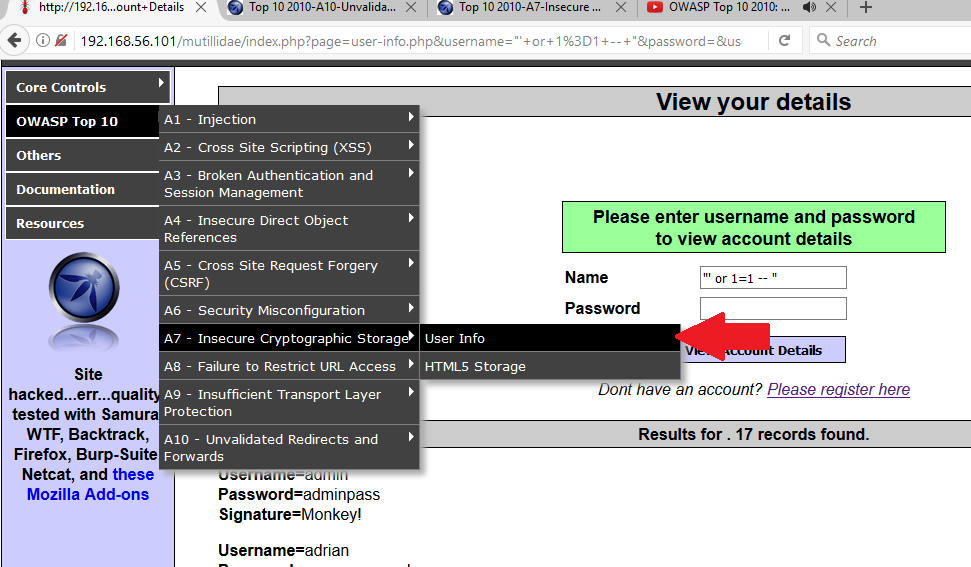
* We can use static analysis tool or scanner to scan through the software and can try finding some flows or break down.

1. **Insecure Cryptographic Storage:**

* This vulnerability has been exploited on Mutillidae application of Meatsploitable2.

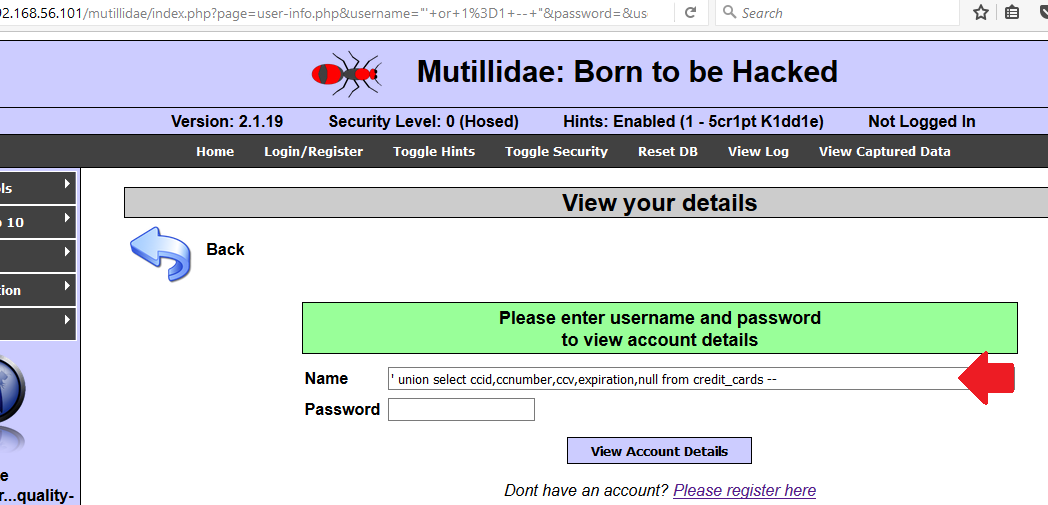
1. ***How to exploit?***

* Click on OWASP Top 10 tab on the left menu, then click A7- Insecure Cryptographic Storage, then click User Info. As shown in below image.

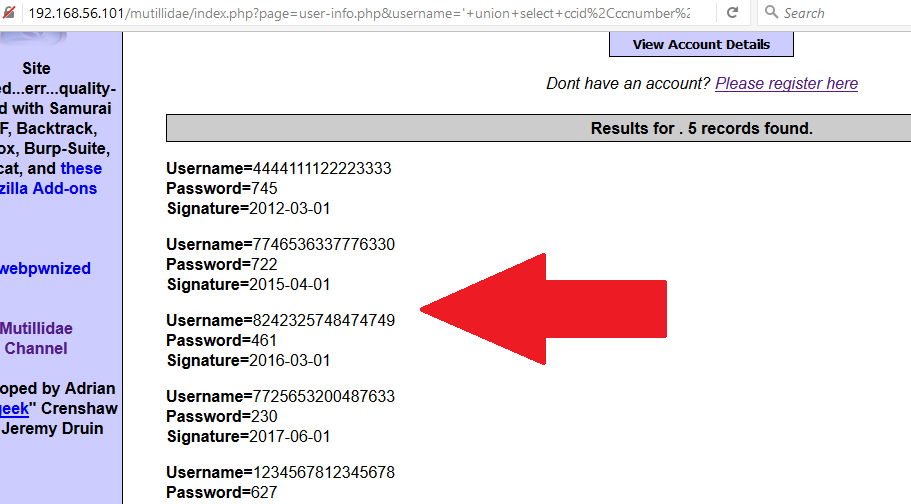


* You will be redirected to the user info page, and this page allows user to get his user details. But there are many data such as credit card information or passwords, such sensitive information should be encrypted and stored.
* We will use union sql injection to get the credit card details of the user.

*' union select ccid,ccnumber,ccv,expiration,null from credit\_cards --(leave space after --)*



* Click on View account details. And the output will be the credit card number, cc number and expiration date of all users. Output has been shown in the image below.



1. ***How to fix?***

* Check if sensitive data has been encrypted and stored.
* Provide authorization to users and check the privileges before granting access to any resource to any user.
* All backups and data which stored onsite or offsite are encrypted.
* Strong encryption algorithm with strong key are used to perform encryption.

1. ***Tools Used?***

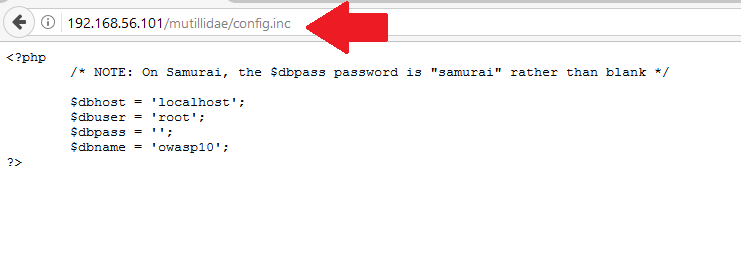
* No extra tool require to exploit this vulnerability.

1. **Failure to restrict URL Access:**

* This vulnerability has been exploited on Mutillidae application of Meatsploitable2. We can directly call config.inc file(by brute force you can easily come to know files present in directory)

1. ***How to exploit?***

* Enter the URL : <http://192.168.56.101/mutillidae/config.inc>
* This will directly open configuration file of mutillidae application. And this file has all information of database. The image is shown in figure.



1. ***How to fix?***

* Authentication and authorization is required to prevent this vulnerability.
* URL should be directed by URL routes mentioned in configuration file.
* Unspecified URL paths in configuration file must be redirected to default path.
* Every user needs to be authenticated to access or to be redirected to any page, only provide access if privileges are granted to authenticated users.

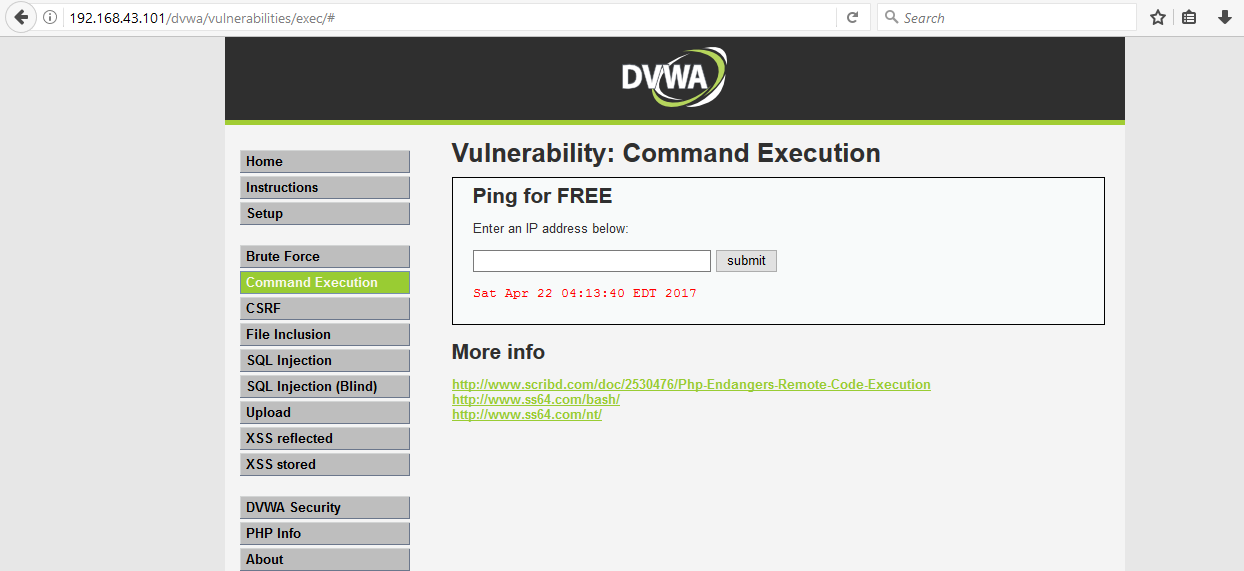
1. ***Tools Used?***

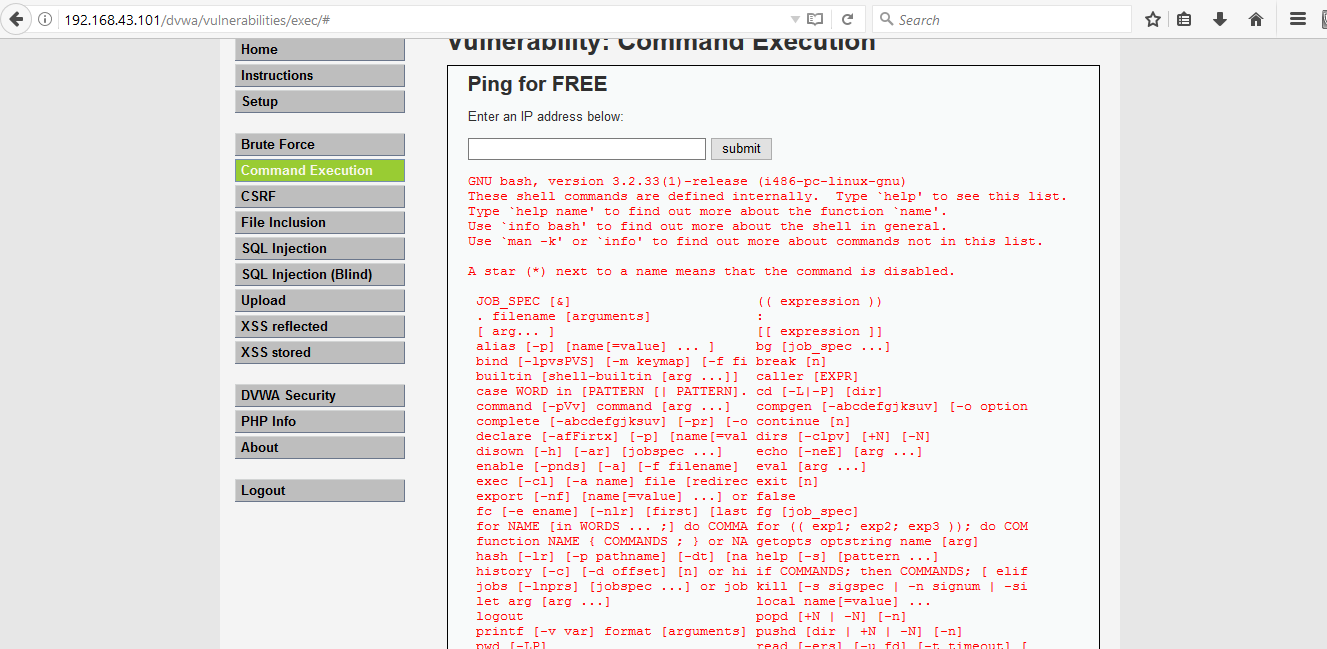
* No extra tool require to exploit this vulnerability.

1. Command execution:

This vulnerability has been exploited on Damn vulnerable web application.

1. How to exploit?
2. Go to command execution in Damn Vulnerable Web App and execute the command “ | help”. This will list all the possible commands that can be executed just like your linux machine.
3. Also type “ |date”, this will display the date.





1. How to fix??

* Input validation has to be done before accepting any input.
* The input should be in the format of an IP Address.

1. Tools used

* No tools have been used for this exploit.

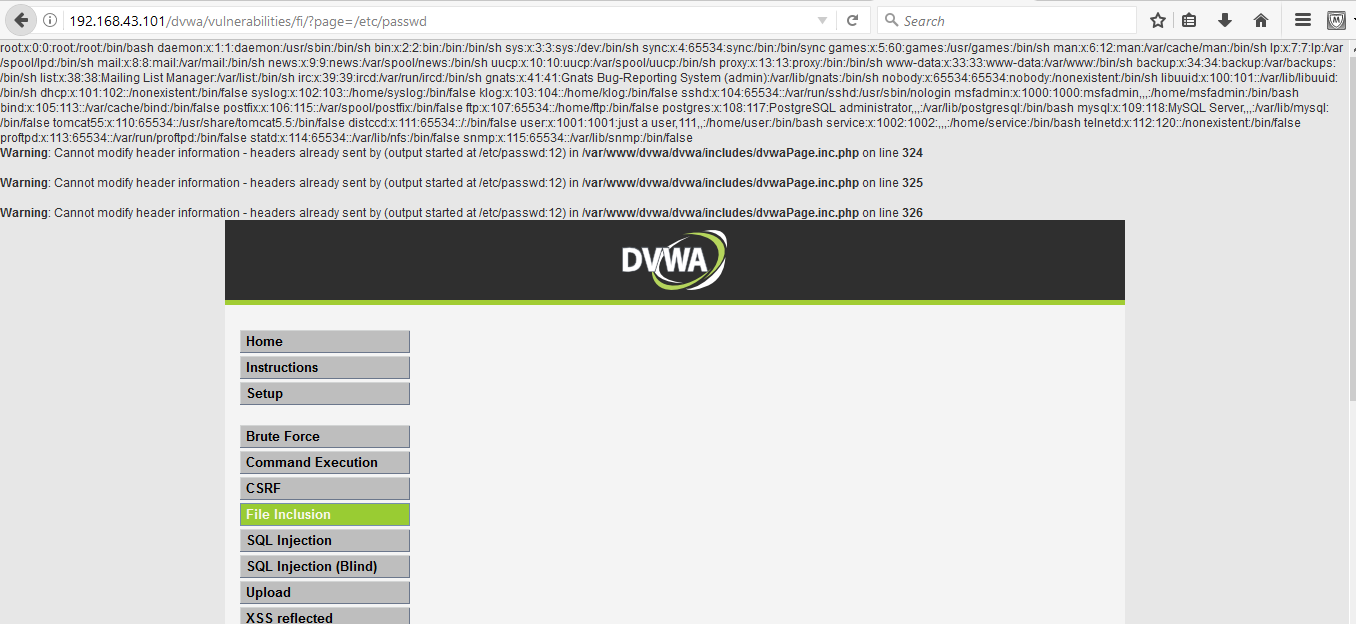
1. File inclusion

This vulnerability has been exploited on Damn vulnerable web application.

1. How to exploit?

Modify the url as : <http://192.168.43.101/dvwa/vulnerabilities/fi/?page=/etc/passwd>.

Then the contents of the file can be seen.



1. How to fix?

All the files used must be encrypted and also the url so that upon request, the contents of the file are not  displayed on the browser.

1. Tools used

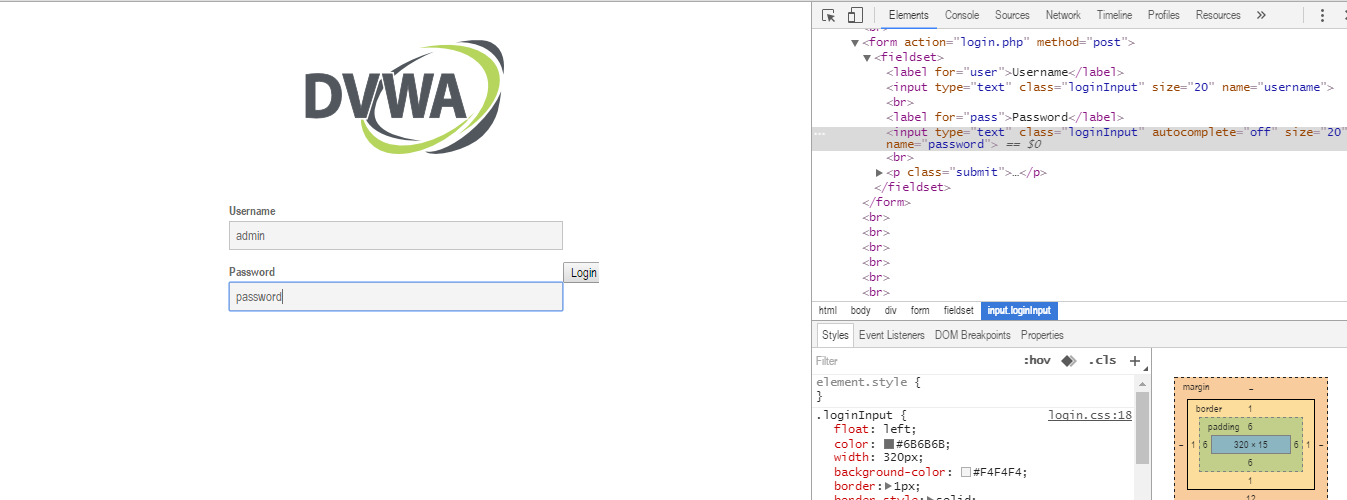
No tools have been used for this exploit.

1. Missing Encryption of Sensitive Data

This vulnerability has been exploited on Damn vulnerable web application.

1. How to exploit?

In the login screen, right click and view page source and edit it to make input type of password as text.



1. How to fix?

The username and password of the site should not be given on the login screen. Also the contents of the html should not be editable.

1. Tools used

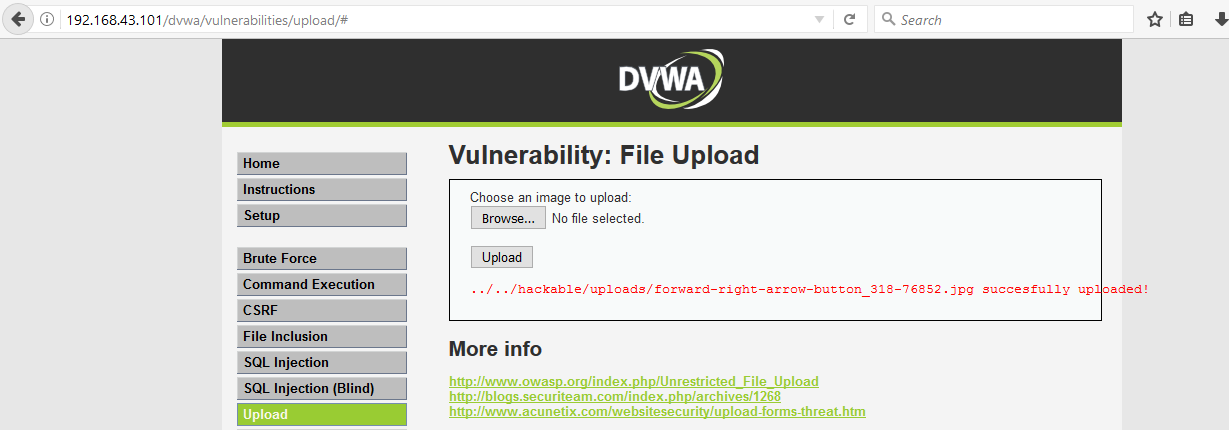
No tools were used for this exploit.

1. Unrestricted Upload of File with Dangerous Type

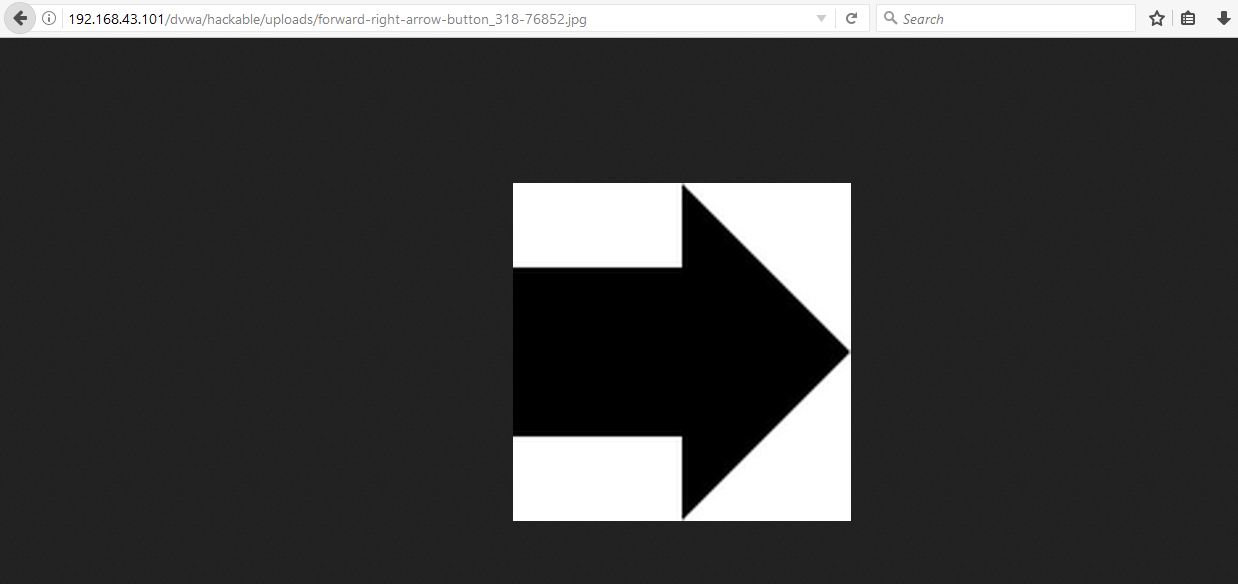
This vulnerability has been exploited on Damn vulnerable web application.

1. How to exploit?

Go to file upload and select a file and upload it. Then take the url that is displayed on the screen to get the image.



The url would be <http://192.168.43.101/dvwa/hackable/uploads/forward-right-arrow-button_318-76852.jpg>

and we get the image.

1. Tools used

No tools were used for this exploit.

1. CSRF( Cross site request forgery)

This vulnerability has been exploited on Damn vulnerable web application.

1. How to exploit?

Go to CSRF tab in DVWA. Then right click and select view page source element.

Copy the form for the password change and modify it.

The modified form would look like this.

<form action="http://192.168.43.101/dvwa/vulnerabilities/csrf/" method="GET">    New password:<br>

   <input type="password" AUTOCOMPLETE="off" name="password\_new" value="sneha"><br>

    Confirm new password: <br>

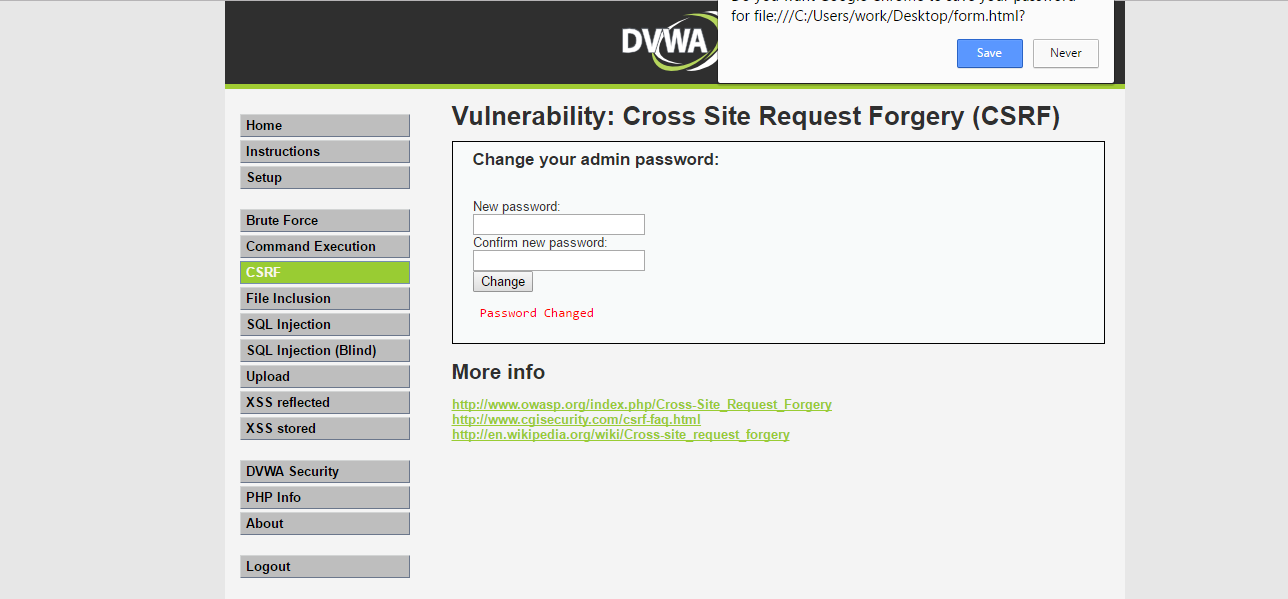
    <input type="password" AUTOCOMPLETE="off" name="password\_conf" value="sneha">

    <br>

    <input type="submit" value="Change" name="Change">

    </form>

when this form is opened in the browser, the password would be changed and it looks like this



1. How to fix?
2. Cookies and session management should be used to avoid this.
3. Tools used

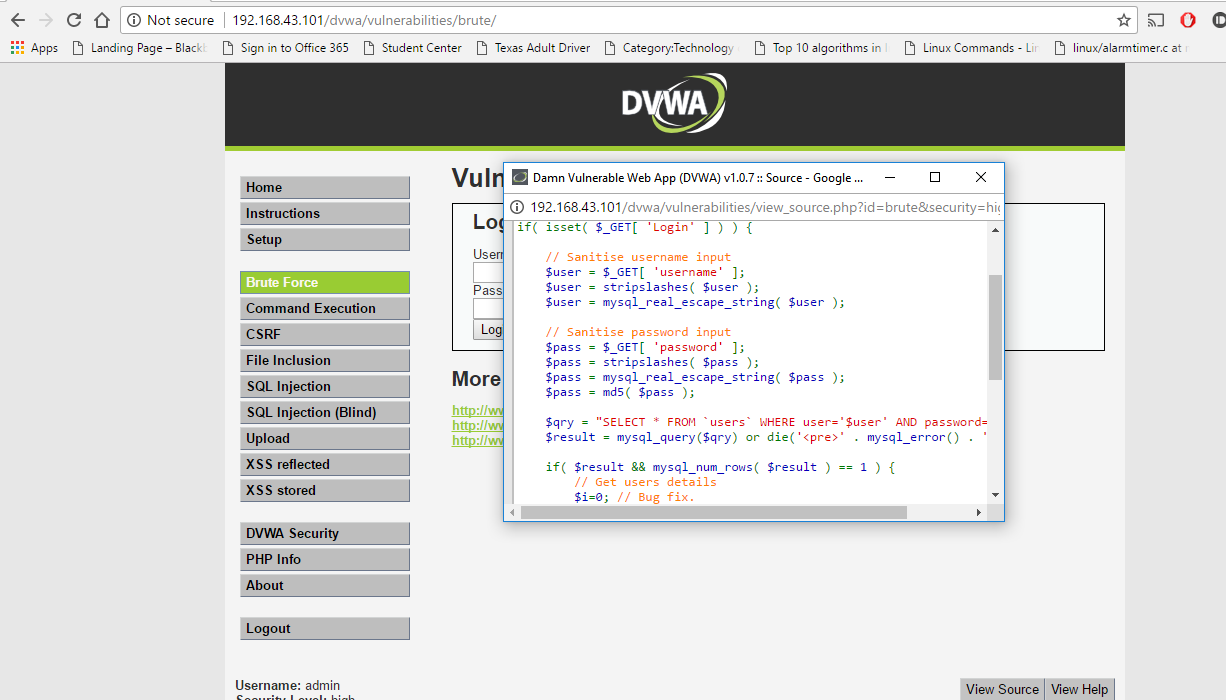
No tools were used for this exploit.

1. Download of Code Without Integrity Check

This vulnerability has been exploited on Damn vulnerable web application.

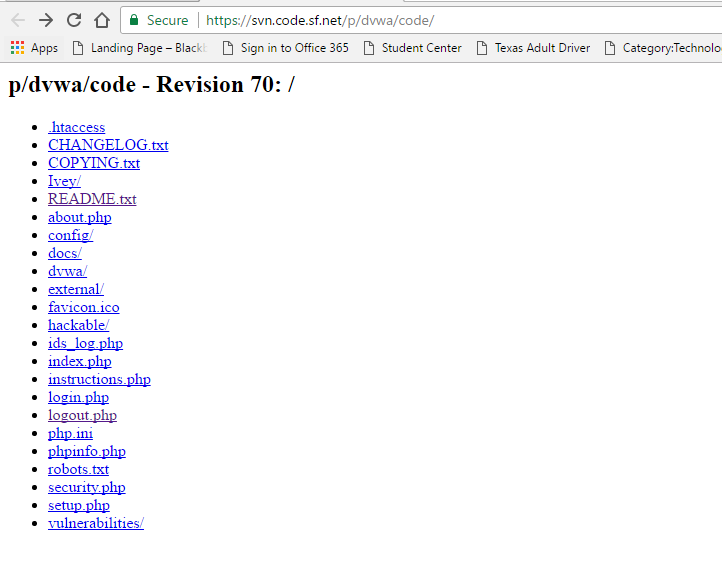
1. How to exploit?

Login to dvwa and click on view source button. The code can be seen



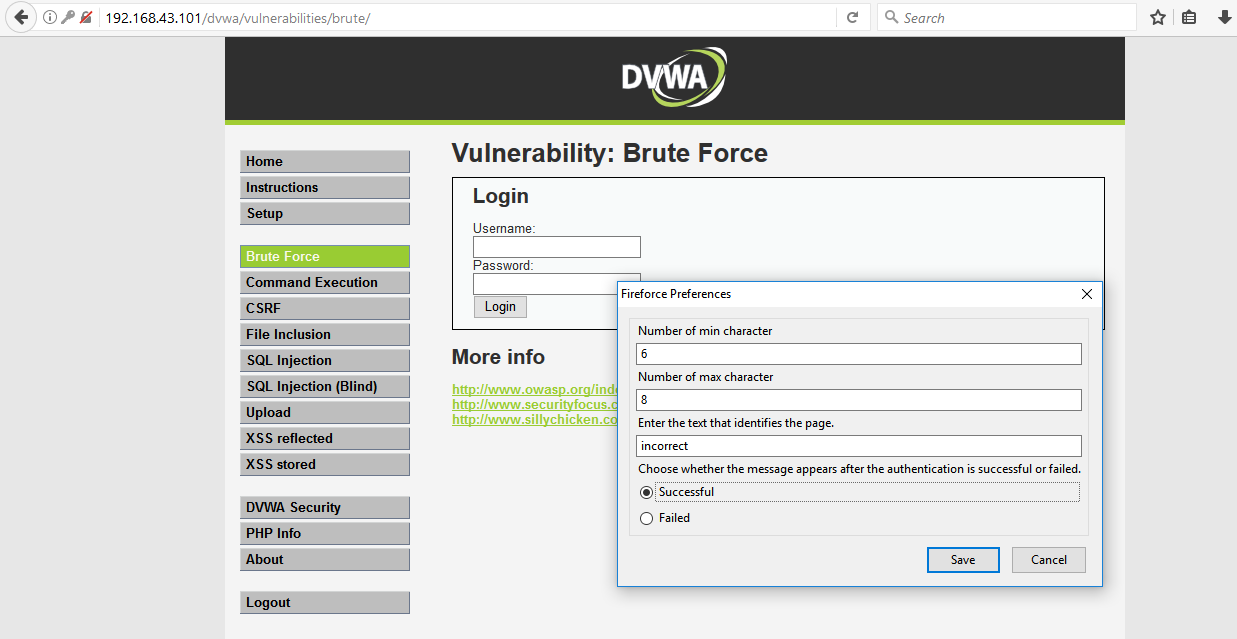
Also go to the about tab and click on the svn link. We can download the source code.

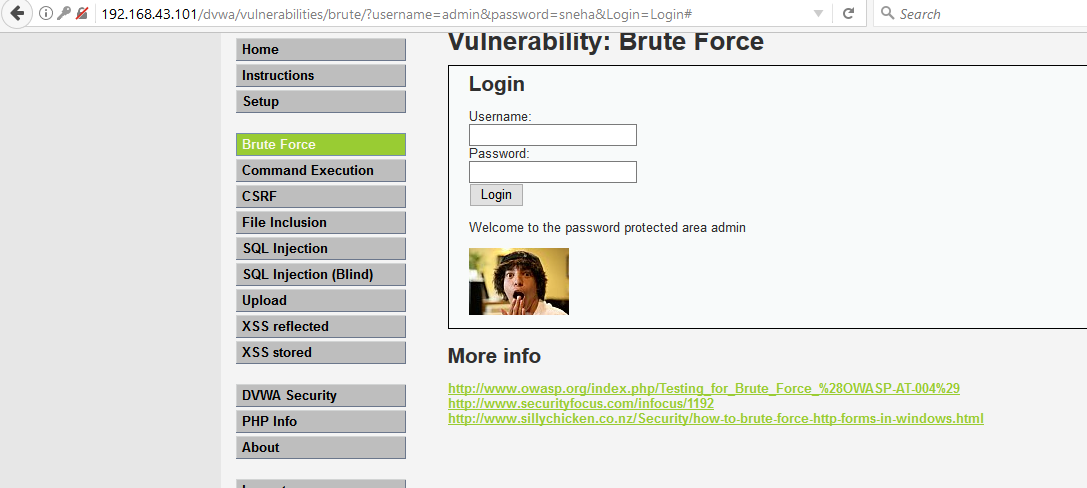
1. Tools used: No tools were used.



1. Improper Restriction of Excessive Authentication Attempts.
2. How to exploit?

Go to brute force tab of DVWA and enter the username and for password, right click and select FireForce and click generate password and give the parameters.





1. Tools used:

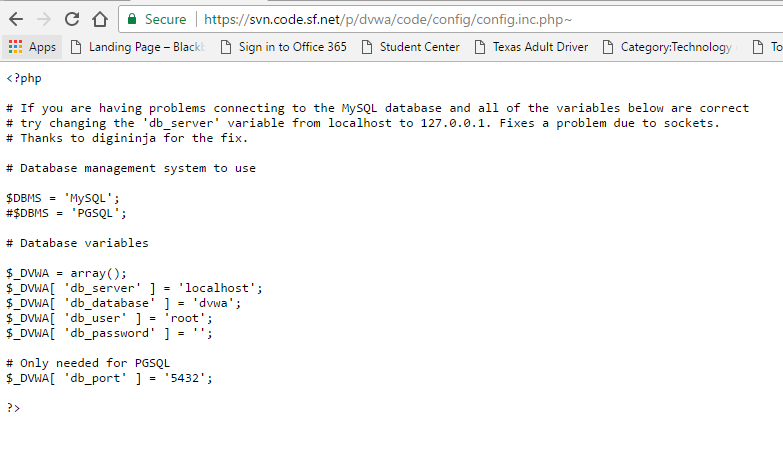
Fireforce is used which is a mozilla firefox extension.

1. Incorrect authorization

This vulnerability has been exploited on Damn vulnerable web application.

1. How to exploit?

Click on About tab in DVWA. Click on the svn link and go to config/config.php where the server name can be changed. Any user can change the server.



1. Tools used:

No tools were used for this exploit.

**References*:***

* <https://www.owasp.org/index.php/Top_10_2010-A8-Failure_to_Restrict_URL_Access>
* <https://www.owasp.org/index.php/Top_10_2010-A7-Insecure_Cryptographic_Storage>
* <https://www.owasp.org/index.php/Top_10_2010-A6-Security_Misconfiguration>
* <https://www.owasp.org/index.php/Top_10_2010-A4-Insecure_Direct_Object_References>
* <https://www.owasp.org/index.php/Top_10_2010-A3-Broken_Authentication_and_Session_Management>
* <https://www.owasp.org/index.php/Top_10_2010-A2-Cross-Site_Scripting_(XSS)>
* <https://www.owasp.org/index.php/Top_10_2010-A1-Injection>
* <https://computersecuritystudent.com/SECURITY_TOOLS/DVWA/DVWAv107/lesson1/index.html>