Lab III - Web Security

Software Security, DV2546

Vinay Kumar Vennu
9408042910
vive16@student.bth.se

Neeraj Reddy Avutu
9411053375
neav16@student.bth.se

INTRODUCTION

The lab is based on the project OWASP Broken Web Applications (OWASPBWA) that produces a virtual machine running a variety of applications with known vulnerabilities. A virtual machine is set up by downloading the exportable image from their website and VM Ware/ VirtualBox application is used to access OWASP. This lab is divided into four tasks that briefly describe the tools/techniques used, vulnerabilities and exploits found or discovered and mainly the countermeasures that ensure web security.

TASK I

The aim of this task is to identify a recent vulnerability related with injection or any top three vulnerabilities from OWASP from the given web sources, give a brief discussion of the risks associated with it and analyze its countermeasures. National Vulnerability Database (NVD), a US government based repository is chosen to search for the vulnerability as it contains data that enables the automation of vulnerability management, security measurement, and compliance [1].

CVE-2016-6619 is the vulnerability chosen. This vulnerability was discovered in the user interface preference feature where a user can execute an SQL injection attack against the account of the control user [2].The severity of this type of vulnerability is considered to be serious and its impact is unauthorized disclosure of information, unauthorized modification and disruption of service [1].

The type of vulnerability is SQL injection, identified as CWE-89 on the CWE list (Common Weakness Enumeration). When there is no sufficient removal or quoting of SQL syntax in user-controllable inputs, the generated SQL query can cause those inputs to be interpreted as SQL instead of ordinary user data. This can be used to alter query logic to bypass security checks, or to insert additional statements that modify the back-end database, possibly including execution of system commands. SQL injection has become a common issue with database-driven web sites. This type of flaw is easily detected and easily exploited [3].

The server must have a control user account created in MySQL and configured in phpMyAdmin; installations without a control user are not vulnerable.

All 4.6.x versions (prior to 4.6.4), 4.4.x versions (prior to 4.4.15.8), and 4.0.x versions (prior to 4.0.10.17) are affected

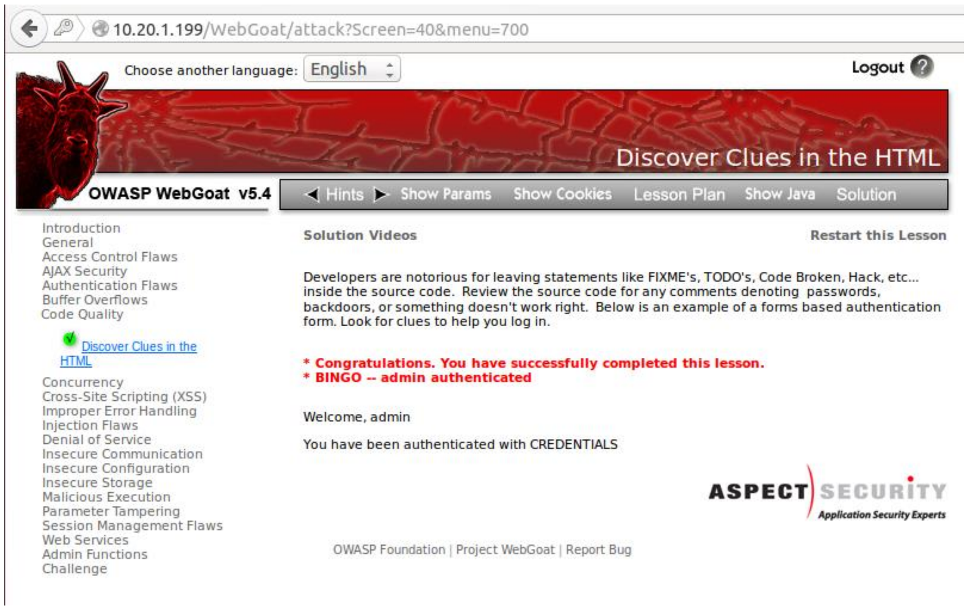
Upgrade to phpMyAdmin 4.6.4, 4.4.15.8, or 4.0.10.17 or newer or apply patch listed below.

TASK 2 : WEB GOAT

1.CODE QUALITY

Discover clues in html

The main objective is that the user must be able to bypass the authentication check. Inside the source code developers are prominent for leaving statements like FIXME's , code broken etc. The source code was reviewed for any comments denoting passwords or backdoors.While examining the html source code, the username admin and password adminpw was found in a comment. Entering the discovered credentials we could login webgoat as shown in figure 1 [4].



Developers are notorious for leaving statements like FIXME's, TODO's, Code Broken, Hack, etc... inside the source code. Review the source code for any comments denoting passwords, backdoors, or something doesn't work right. Below is an example of a forms based authentication form. Look for clues to help you log in.

* Congratulations. You have successfully completed this lesson.
* BINGO -- admin authenticated

Welcome, admin

You have been authenticated with CREDENTIALS

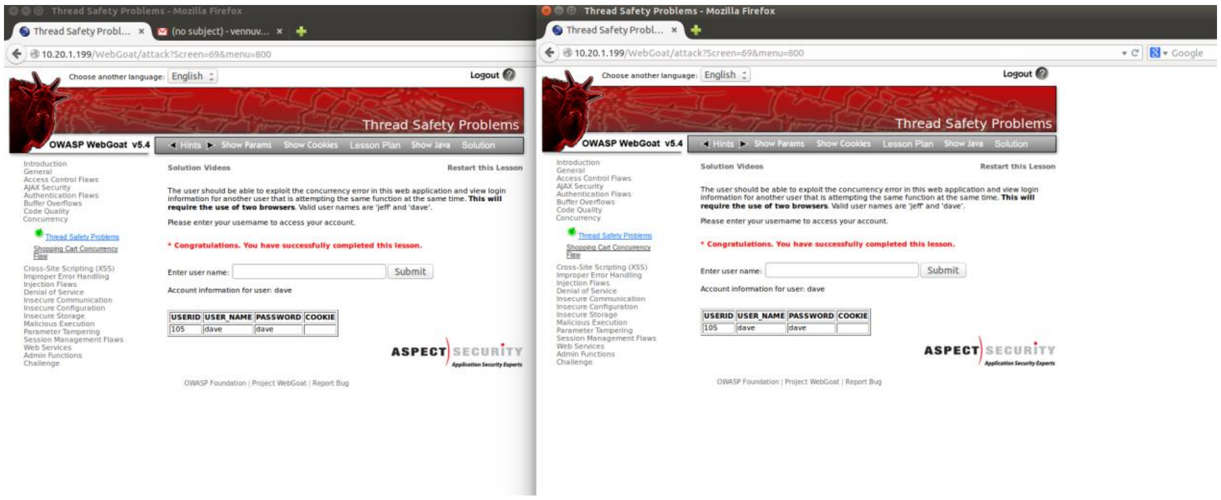
Countermeasures

When developing the source code of html page there must be certain limitations in showing the context and validations must be given to the code. Proper training of the developers prior to coding and use of automated tools for checking the quality of code can be used as countermeasures [5].

2. CONCURRENCY

Thread safety problems

The main goal of the user is to exploit the concurrency error in the web application and view login information for another user that is attempting the same function at the same time. Since web applications handle many http requests and the variables used by the developers are not thread safe, there is a possible way to exploit the concurrency bug by loading same page as another user at same time. The result obtained has same data in both the windows. The root cause could be identified when the java code is examined as it uses static variable [4].



The user should be able to exploit the concurrency error in this web application and view login information for another user that is attempting the same function at the same time. **This will require the use of two browsers**. Valid user names are 'jeff' and 'dave'.

Please enter your username to access your account.

* Congratulations. You have successfully completed this lesson.

Enter user name:

Account information for user: dave

USERID	USER_NAME	PASSWORD	COOKIE
105	dave	dave	

The user should be able to exploit the concurrency error in this web application and view login information for another user that is attempting the same function at the same time. **This will require the use of two browsers**. Valid user names are 'jeff' and 'dave'.

Please enter your username to access your account.

* Congratulations. You have successfully completed this lesson.

Enter user name:

Account information for user: dave

USERID	USER_NAME	PASSWORD	COOKIE
105	dave	dave	

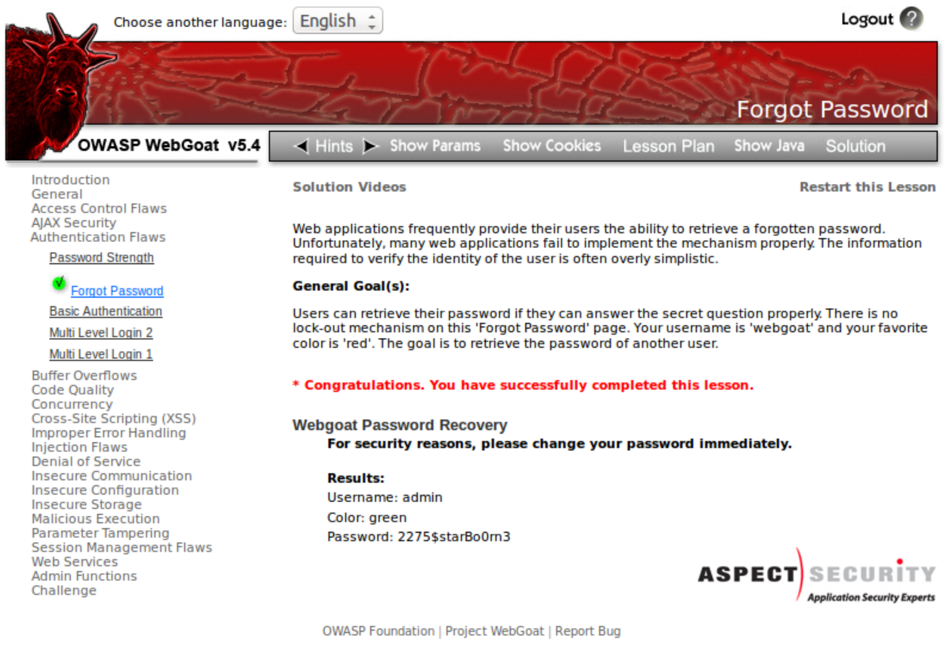
Countermeasures

Use of concurrency controlled mechanisms such as serializability, recoverability etc the test cases can be validated and implementing these avoids race conditions [5].

3. AUTHENTICATION FLAWS

Forgot password

The goal of the user is to retrieve password of another user.Users can retrieve their password when the secret question is answered properly.The main concept is that Web applications fail to implement the mechanism of providing ability to retrieve a forgotten password since the information used to assess the identity of the user is very simple. The password of the admin was retrieved by typing the color green after implementing brute force attack [4].



Web applications frequently provide their users the ability to retrieve a forgotten password. Unfortunately, many web applications fail to implement the mechanism properly. The information required to verify the identity of the user is often overly simplistic.

General Goal(s):

Users can retrieve their password if they can answer the secret question properly. There is no lock-out mechanism on this 'Forgot Password' page. Your username is 'webgoat' and your favorite color is 'red'. The goal is to retrieve the password of another user.

* Congratulations. You have successfully completed this lesson.

Webgoat Password Recovery

For security reasons, please change your password immediately.

Results:

Username: admin

Color: green

Password: 2275$starBo0rn3

Countermeasures

The security question should not be simple to guess and th number of attempts of actions we perform during brute force attack must be validated [5].

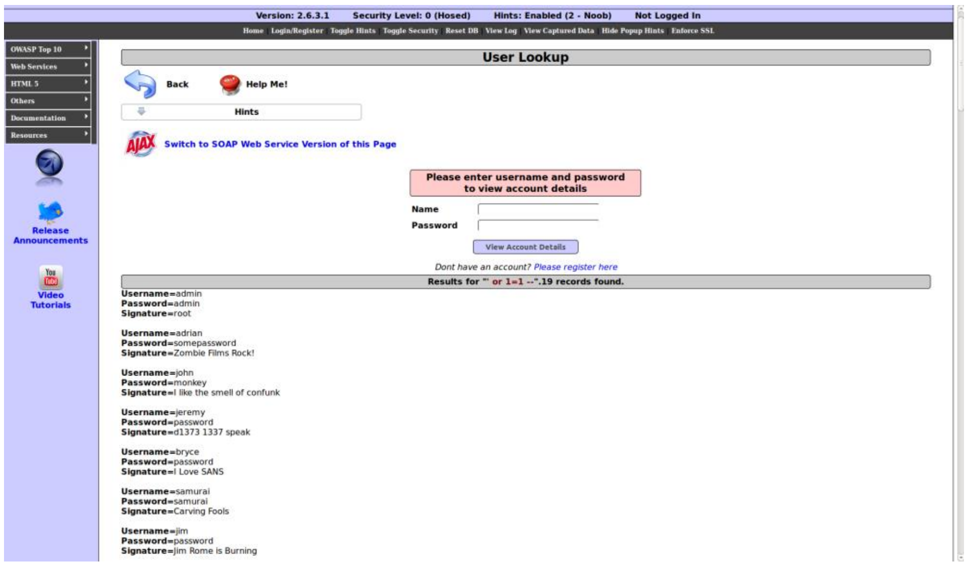
TASK 3: MUTILLIDAE

1.SQL INJECTION-SQLi-Extract data-User Info

To exploit information of all users we perform SQL injection in this vulnerability. By referring the source [5] we performed the exploitation.The characters such as single-quote, back-slash, double-hyphen, forward-slash. Injections are done on the basis of true and false values.

‘ or 1 = 1 -

This text is used for both username and password.We get the users account information by clicking view details as shown in the below figure


Version: 2.6.3.1 Security Level: 0 (Hosed) Hints: Enabled (2 - Noob) Not Logged In
Home Login/Register Toggle Hints Toggle Security Reset DB View Log View Captured Data Hide Popup Hints Enforce SSL
OWASP Top 10
Web Services
HTML 5
Others
Documentation
Resources
Release Announcements
Video Tutorials
User Lookup
Back
Help Me!
Hints
Switch to SOAP Web Service Version of this Page
View Account Details

Please enter username and password to view account details

Name

Password

Dont have an account? Please register here

Results for "' or 1=1 --".19 records found.

```
Username=admin
Password=admin
Signature=root

Username=adrian
Password=somepassword
Signature=Zombie Films Rock!

Username=john
Password=monkey
Signature=I like the smell of confunk

Username=jeremy
Password=password
Signature=d1373 1337 speak

Username=bryce
Password=password
Signature=I Love SANS

Username=samurai
Password=samurai
Signature=Carving Fools

Username=jim
Password=password
Signature=Jim Rome is Burning
```

Countermeasures [6]

1.Avoiding dynamic SQL queries

2.Whitelisting input validations

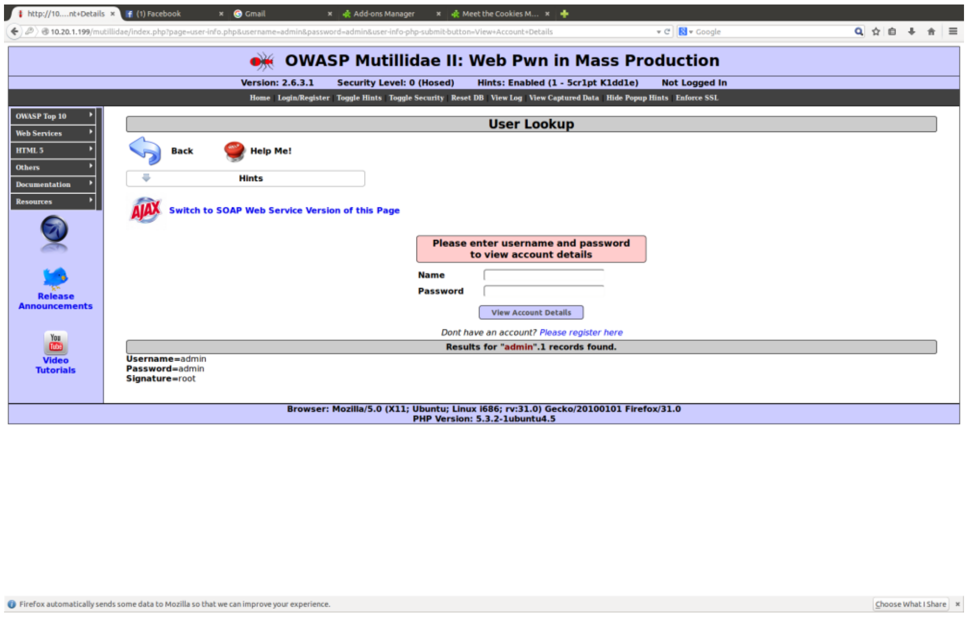
2.SQL INJECTION-SQLI-Bypass Authentication-Login

Using SQL injection we bypass authentication on the login page.This can be done using the name field to authenticate the first user found in the user table [5].

Username: '1'='1'

Password: 'or'1'='1'

We sign in as admin by entering same text in the username and password fields.


10.20.1.199/mutillidae/index.php?page=user-info.php&username=admin&password=admin&user-info-php-submit-button=View+Account+Details

OWASP Mutillidae II: Web Pwn in Mass Production

Version: 2.6.3.1 Security Level: 0 (Hosed) Hints: Enabled (1 - 5cr1pt K1dd1e) Not Logged In

User Lookup

Please enter username and password to view account details

Name:
Password:

Dont have an account? Please register here

Results for "admin".1 records found.

Username=admin
Password=admin
Signature=root

Browser: Mozilla/5.0 (X11; Ubuntu; Linux i686; rv:31.0) Gecko/20100101 Firefox/31.0
PHP Version: 5.3.2-1ubuntu4.5

Firefox automatically sends some data to Mozilla so that we can improve your experience.

Home | Login/Register | Toggle Hints | Toggle Security | Reset DB | View Log | View Captured Data | Hide Popup Hints | Enforce SSL
OWASP Top 10
Web Services
HTML 5
Others
Documentation
Resources
Back
Help Me!
Hints
Switch to SOAP Web Service Version of this Page
View Account Details
Release Announcements
Video Tutorials
Choose What I Share

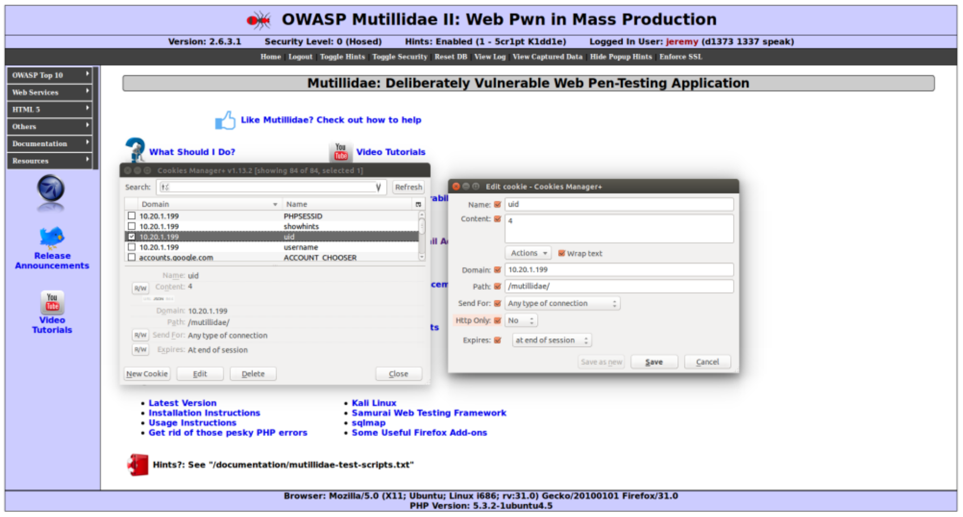
Countermeasures [6]

1.safe characters in place of special characters.

2.use of specific escape syntax.

3. Broken Authentication and session management-Authentication bypass-Via cookies

We need to register in the beginning by entering random username and password. After the installations of cookie manager + from extensions for checking the cookies (to be installed if not found in the system). By changing the value of aid value from 24 to 1, we could successfully login [5].


OWASP Mutillidae II: Web Pwn in Mass Production

Version: 2.6.3.1 Security Level: 0 (Hosed) Hints: Enabled (1 - 5cr1pt K1dd1e) Logged In User: jeremy (d1373 1337 speak)

Mutillidae: Deliberately Vulnerable Web Pen-Testing Application

Cookies Manager+ v1.13.2 [showing 84 of 84, selected 1]

Domain	Name
10.20.1.199	PHPSESSID
10.20.1.199	showhints
10.20.1.199	uid
10.20.1.199	username
accounts.google.com	ACCOUNT_CHOOSER

Name: uid
Content: 4
Domain: 10.20.1.199
Path: /mutillidae/
Send For: Any type of connection
Expires: At end of session

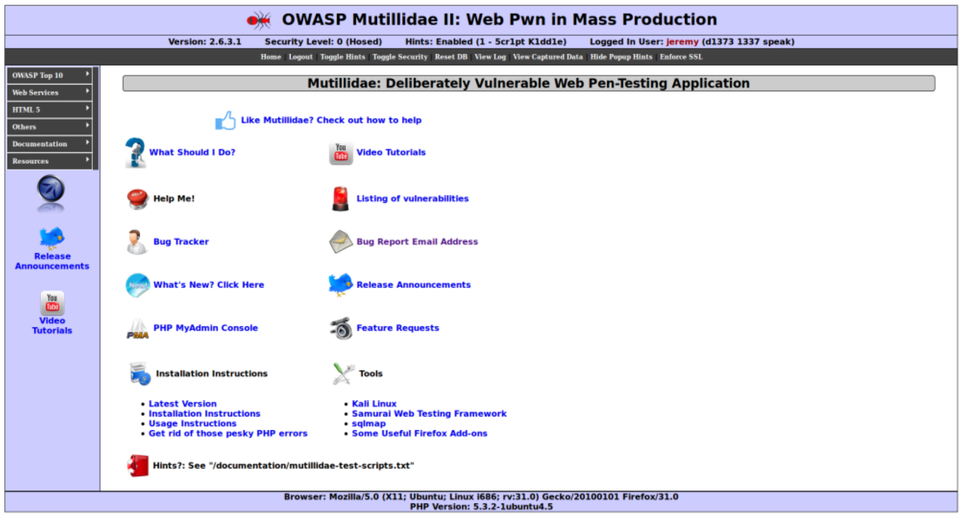
Edit cookie - Cookies Manager+

Name: uid
Content: 4
Domain: 10.20.1.199
Path: /mutillidae/
Send For: Any type of connection
Http Only: No
Expires: at end of session

Hints?: See "/documentation/mutillidae-test-scripts.txt"

Browser: Mozilla/5.0 (X11; Ubuntu; Linux i686; rv:31.0) Gecko/20100101 Firefox/31.0
PHP Version: 5.3.2-1ubuntu4.5

Home | Logout | Toggle Hints | Toggle Security | Reset DB | View Log | View Captured Data | Hide Popup Hints | Enforce SSL
OWASP Top 10
Web Services
HTML 5
Others
Documentation
Resources
Like Mutillidae? Check out how to help
What Should I Do?
Video Tutorials
Search:
Refresh
New Cookie
Edit
Delete
Close
Actions
Wrap text
Save as new
Save
Cancel
Latest Version
Installation Instructions
Usage Instructions
Get rid of those pesky PHP errors
Kali Linux
Samurai Web Testing Framework
sqlmap
Some Useful Firefox Add-ons
Release Announcements
Video Tutorials


OWASP Mutillidae II: Web Pwn in Mass Production

Version: 2.6.3.1 Security Level: 0 (Hosed) Hints: Enabled (1 - 5cr1pt K1dd1e) Logged In User: jeremy (d1373 1337 speak)

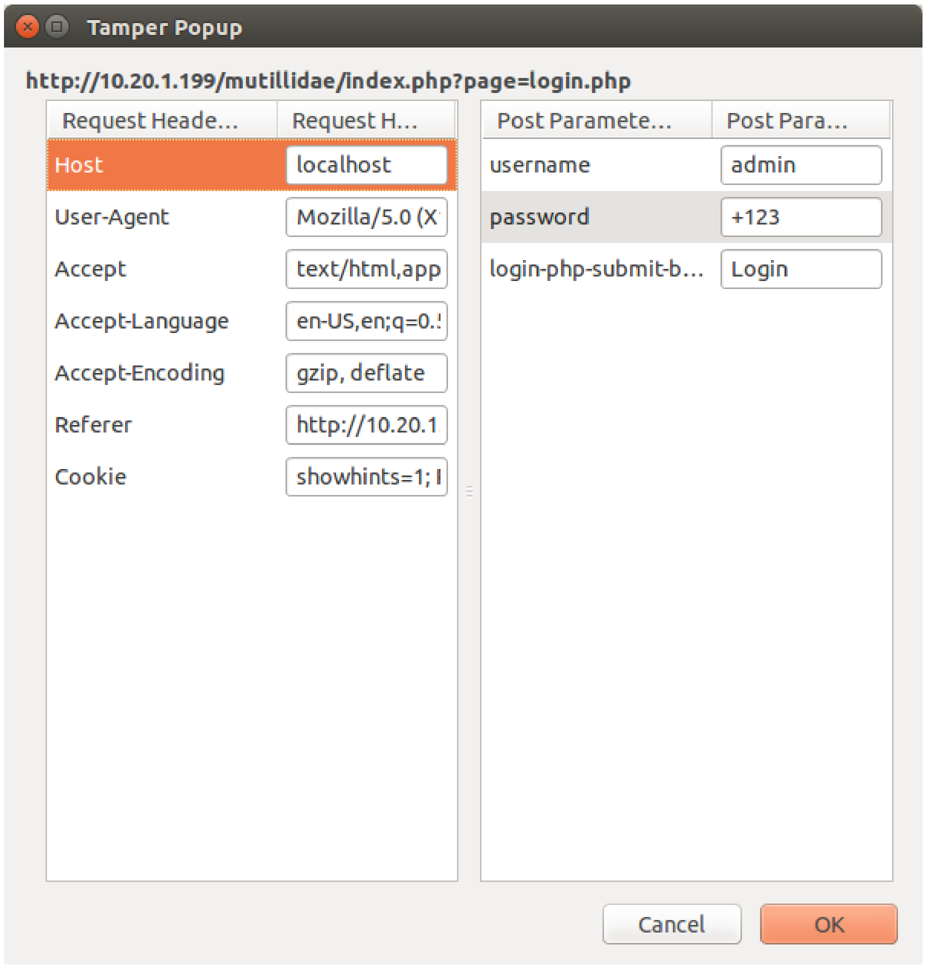
Mutillidae: Deliberately Vulnerable Web Pen-Testing Application

Hints?: See "/documentation/mutillidae-test-scripts.txt"

Browser: Mozilla/5.0 (X11; Ubuntu; Linux i686; rv:31.0) Gecko/20100101 Firefox/31.0
PHP Version: 5.3.2-1ubuntu4.5

4. Broken Authentication and session management-Login-Via SQL injection

We had done this by downloading an extension tamper data. After entering username as ‘admin’ and some random number as password, we will start tamper by selecting tamper data in tools menu. Then we will login to the page and change request beside host to localhost and the password to ‘1’=‘1. After refreshing the page, we can find that admin is logged in.

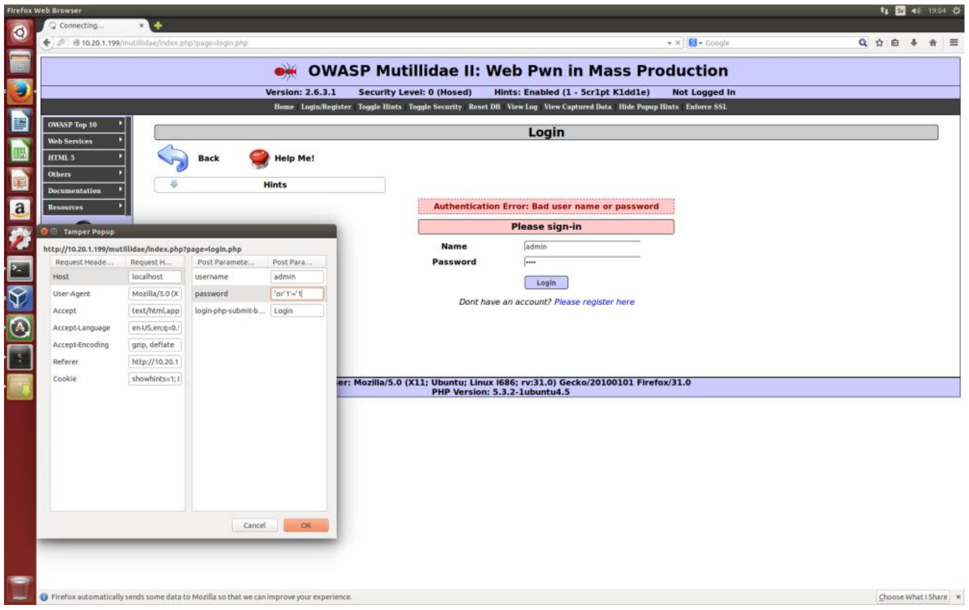


Tamper Popup

http://10.20.1.199/mutillidae/index.php?page=login.php

Request Heade...	Request H...
Host	localhost
User-Agent	Mozilla/5.0 (X
Accept	text/html,app
Accept-Language	en-US,en;q=0.!
Accept-Encoding	gzip, deflate
Referer	http://10.20.1
Cookie	showhints=1; I

Post Paramete...	Post Para...
username	admin
password	+123
login-php-submit-b...	Login



Firefox Web Browser
Connecting...
19:04
Google

10.20.1.199/mutillidae/index.php?page=login.php

OWASP Mutillidae II: Web Pwn in Mass Production

Version: 2.6.3.1 Security Level: 0 (Hosed) Hints: Enabled (1 - 5cr1pt K1dd1e) Not Logged In

Home | Login/Register | Toggle Hints | Toggle Security | Reset DB | View Log | View Captured Data | Hide Popup Hints | Enforce SSL
OWASP Top 10
Web Services
HTML 5
Others
Documentation
Resources
Login
Back
Help Me!
Hints

Authentication Error: Bad user name or password

Please sign-in

Name	admin
Password	••••

Dont have an account? Please register here

er: Mozilla/5.0 (X11; Ubuntu; Linux i686; rv:31.0) Gecko/20100101 Firefox/31.0
PHP Version: 5.3.2-1ubuntu4.5

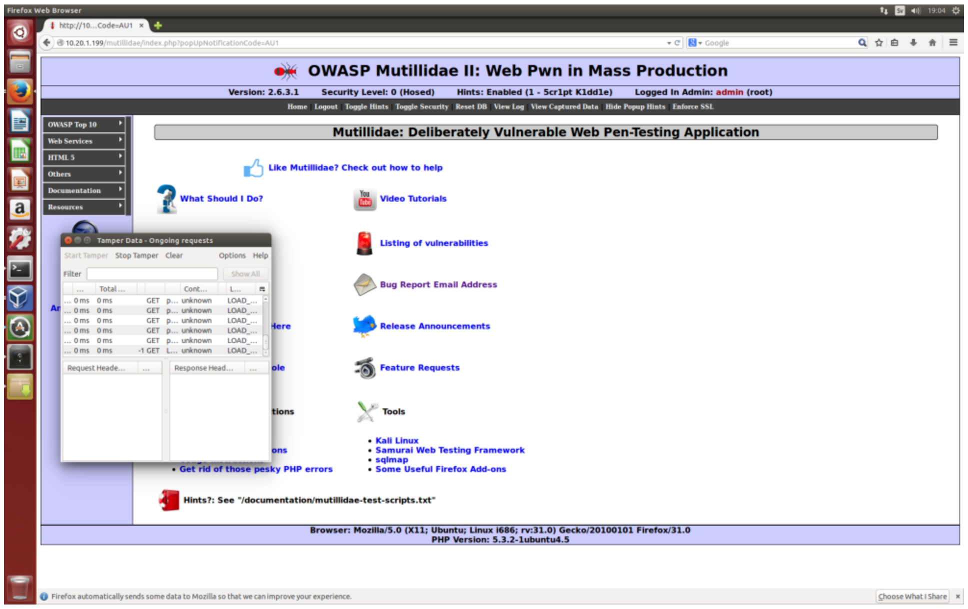
Tamper Popup

http://10.20.1.199/mutillidae/index.php?page=login.php

Request Heade...	Request H...
Host	localhost
User-Agent	Mozilla/5.0 (X
Accept	text/html,app
Accept-Language	en-US,en;q=0.!
Accept-Encoding	gzip, deflate
Referer	http://10.20.1
Cookie	showhints=1; I

Post Paramete...	Post Para...
username	admin
password	'or'1'='1
login-php-submit-b...	Login

Firefox automatically sends some data to Mozilla so that we can improve your experience.


Firefox Web Browser
19:04
http://10....Code=AU1
10.20.1.199/mutillidae/index.php?popUpNotificationCode=AU1
Google
Home | Logout | Toggle Hints | Toggle Security | Reset DB | View Log | View Captured Data | Hide Popup Hints | Enforce SSL
OWASP Top 10
Web Services
HTML 5
Others
Documentation
Resources
Like Mutillidae? Check out how to help
What Should I Do?
Video Tutorials
Listing of vulnerabilities
Bug Report Email Address
Release Announcements
Feature Requests
Tools
Kali Linux
Samurai Web Testing Framework
sqlmap
Some Useful Firefox Add-ons
Get rid of those pesky PHP errors
Start Tamper
Stop Tamper
Clear
Options
Help
Filter
Show All
Request Heade...
Response Head...
Choose What I Share

OWASP Mutillidae II: Web Pwn in Mass Production

Version: 2.6.3.1 Security Level: 0 (Hosed) Hints: Enabled (1 - 5cr1pt K1dd1e) Logged In Admin: admin (root)

Mutillidae: Deliberately Vulnerable Web Pen-Testing Application

Tamper Data - Ongoing requests

```
...     Total ...          Cont...      L...
... 0 ms  0 ms     GET  p... unknown   LOAD_...
... 0 ms  0 ms     GET  p... unknown   LOAD_...
... 0 ms  0 ms     GET  p... unknown   LOAD_...
... 0 ms  0 ms     GET  p... unknown   LOAD_...
... 0 ms  0 ms     GET  p... unknown   LOAD_...
... 0 ms  0 ms  -1 GET  L... unknown   LOAD_...
```

Hints?: See "/documentation/mutillidae-test-scripts.txt"

Browser: Mozilla/5.0 (X11; Ubuntu; Linux i686; rv:31.0) Gecko/20100101 Firefox/31.0
PHP Version: 5.3.2-1ubuntu4.5

Firefox automatically sends some data to Mozilla so that we can improve your experience.

Countermeasures for 3 and 4 [7]

1. Avoiding XSS flaws by taking certain actions.
2. On request encryption must be enabled.

5.Cross site scripting (XSS)-Reflected (First order)-DNS Lookup

The website is evaluated to find any vulnerabilities if present.

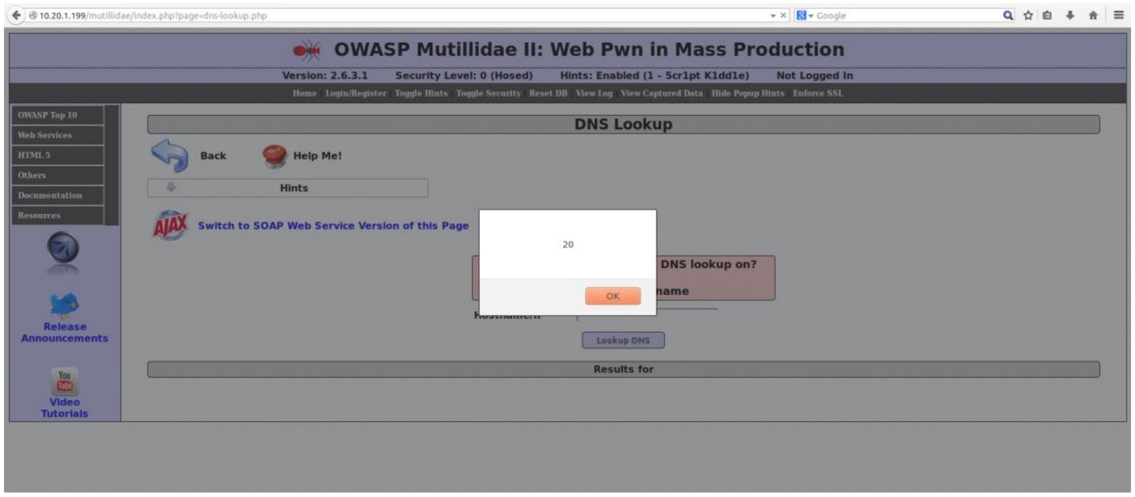
A random string is put in place of domain and the website has returned the value.This means that the website is now vulnerable to cross site scripting.The input string is executed in

ReflectedXSSExecutionPoint=”1”>;Results for asdf<;/div><pre class=”report-header”

Now input the following script :

<Script>alert(10)</script>;

This will print 20 as message as shown in the figure and confirms thata the website is XSS vulnerable.


10.20.1.199/mutillidae/index.php?page=dns-lookup.php
OWASP Mutillidae II: Web Pwn in Mass Production
Version: 2.6.3.1 Security Level: 0 (Hosed) Hints: Enabled (1 - 5cr1pt K1dd1e) Not Logged In
Home | Login/Register | Toggle Hints | Toggle Security | Reset DB | View Log | View Captured Data | Hide Popup Hints | Enforce SSL
OWASP Top 10
Web Services
HTML 5
Others
Documentation
Resources
DNS Lookup
Back
Help Me!
Hints
Switch to SOAP Web Service Version of this Page
Release Announcements
Video Tutorials
Lookup DNS
Results for
OK

20

DNS lookup on?

6.Other injection-HTML Injection –Set Background color

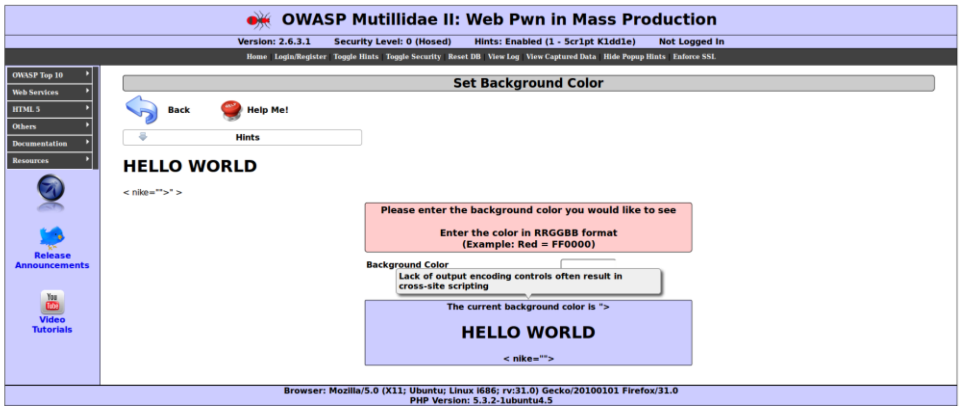
The CSS page is injected with a cross site script.The page sets background colour with the given input color. The color is set with the following style code

Style="background color:#color"

We now end the style with a "> and start our script and then end the " after colour with any tag.Our injection is like this.

"><H1>Hello world<H1/><span"

This will execute our injection code of Hello world. The result is shown in figure


OWASP Mutillidae II: Web Pwn in Mass Production
Version: 2.6.3.1 Security Level: 0 (Hosed) Hints: Enabled (1 - 5cr1pt K1dd1e) Not Logged In
Home | Login/Register | Toggle Hints | Toggle Security | Reset DB | View Log | View Captured Data | Hide Popup Hints | Enforce SSL
OWASP Top 10
Web Services
HTML 5
Others
Documentation
Resources
Set Background Color
Back
Help Me!
Hints
Release Announcements
Video Tutorials
Background Color

HELLO WORLD

< nike="">" >

Please enter the background color you would like to see

Enter the color in RRGGBB format
(Example: Red = FF0000)

Lack of output encoding controls often result in cross-site scripting

The current background color is ">

HELLO WORLD

< nike="">

Browser: Mozilla/5.0 (X11; Ubuntu; Linux i686; rv:31.0) Gecko/20100101 Firefox/31.0
PHP Version: 5.3.2-1ubuntu4.5

Countermeasures

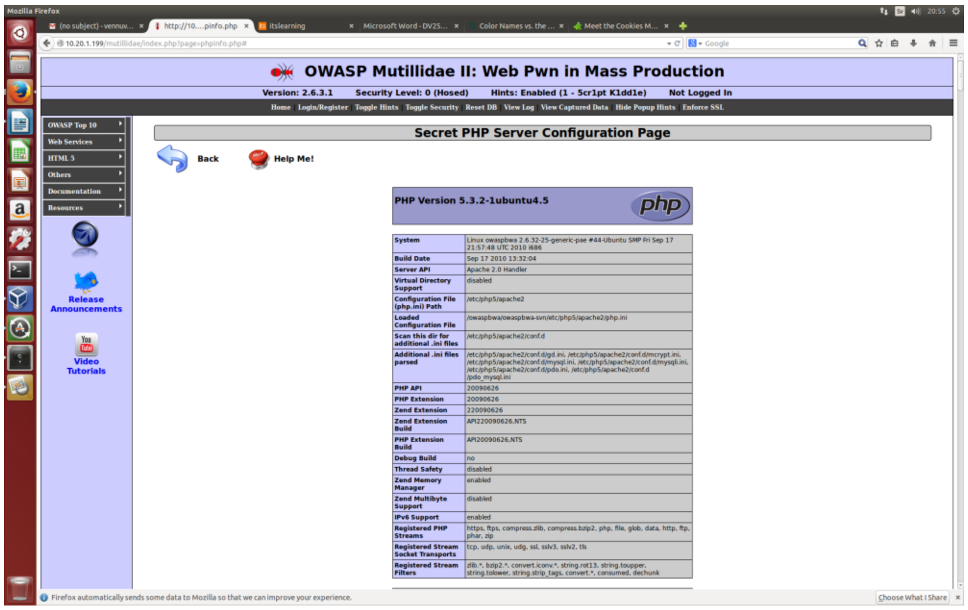
1. User input is set to minimum.
2. CSS hex encoding must be used.

7. Sensitive data exposure-information disclosure-PHP info page

Sensitive information can be exploited by viewing directly PHP info page.Hackers gain the web server information by the phpinfo() which developers use to know the version of php used.Attackers can easily bypass the possible IDS detection.

Countermeasures [8]

In the server phpinfo()must not be enclosed.Sensitive files are scanned at certain intervals of time to prevent any such threat.



OWASP Mutillidae II: Web Pwn in Mass Production
Version: 2.6.3.1
Security Level: 0 (Hosed)
Hints: Enabled (1 - 5cr1pt K1dd1e)
Not Logged In
Home | Login/Register | Toggle Hints | Toggle Security | Reset DB | View Log | View Captured Data | Hide Popup Hints | Enforce SSL
OWASP Top 10
Web Services
HTML 5
Others
Documentation
Resources
Release Announcements
Video Tutorials
Back
Help Me!

Secret PHP Server Configuration Page

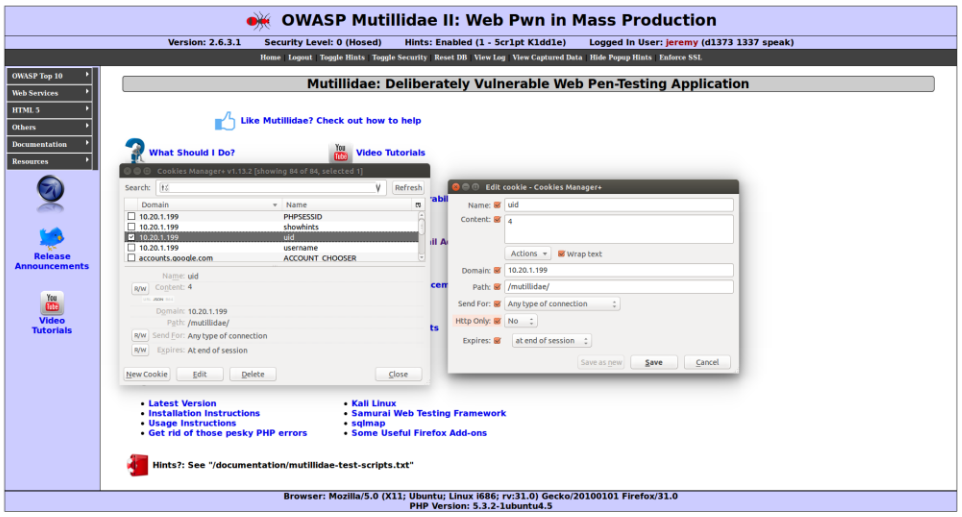
PHP Version 5.3.2-1ubuntu4.5

System	Linux owaspbwa 2.6.32-25-generic-pae #44-Ubuntu SMP Fri Sep 17 21:57:48 UTC 2010 i686
Build Date	Sep 17 2010 13:32:04
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php5/apache2
Loaded Configuration File	/owaspbwa/owaspbwa-svn/etc/php5/apache2/php.ini
Scan this dir for additional .ini files	/etc/php5/apache2/conf.d
Additional .ini files parsed	/etc/php5/apache2/conf.d/gd.ini, /etc/php5/apache2/conf.d/mcrypt.ini, /etc/php5/apache2/conf.d/mysql.ini, /etc/php5/apache2/conf.d/mysqli.ini, /etc/php5/apache2/conf.d/pdo.ini, /etc/php5/apache2/conf.d /pdo_mysql.ini
PHP API	20090626
PHP Extension	20090626
Zend Extension	220090626
Zend Extension Build	API220090626,NTS
PHP Extension Build	API20090626,NTS
Debug Build	no
Thread Safety	disabled
Zend Memory Manager	enabled
Zend Multibyte Support	disabled
IPv6 Support	enabled
Registered PHP Streams	https, ftps, compress.zlib, compress.bzip2, php, file, glob, data, http, ftp, phar, zip
Registered Stream Socket Transports	tcp, udp, unix, udg, ssl, sslv3, sslv2, tls
Registered Stream Filters	zlib.*, bzip2.*, convert.iconv.*, string.rot13, string.toupper, string.tolower, string.strip_tags, convert.*, consumed, dechunk

Firefox automatically sends some data to Mozilla so that we can improve your experience.

8. Insufficient Transport Layer Protection-Login

We have exploited login with the help of cookie manager in the vulnerability transport layer. We logged into it by using the username and password 'user' where did of the current cookie was replaced by 1 and the page was refreshed.



OWASP Mutillidae II: Web Pwn in Mass Production
Version: 2.6.3.1
Security Level: 0 (Hosed)
Hints: Enabled (1 - 5cr1pt K1dd1e)
Logged In User: jeremy (d1373 1337 speak)
Home | Logout | Toggle Hints | Toggle Security | Reset DB | View Log | View Captured Data | Hide Popup Hints | Enforce SSL
OWASP Top 10
Web Services
HTML 5
Others
Documentation
Resources
Release Announcements
Video Tutorials
Like Mutillidae? Check out how to help
What Should I Do?
Video Tutorials

Mutillidae: Deliberately Vulnerable Web Pen-Testing Application

Cookies Manager+ v1.13.2 [showing 84 of 84, selected 1]

Domain	Name
10.20.1.199	PHPSESSID
10.20.1.199	showhints
10.20.1.199	uid
10.20.1.199	username
accounts.google.com	ACCOUNT_CHOOSER

Name: uid
Content: 4
Domain: 10.20.1.199
Path: /mutillidae/
Send For: Any type of connection
Expires: At end of session

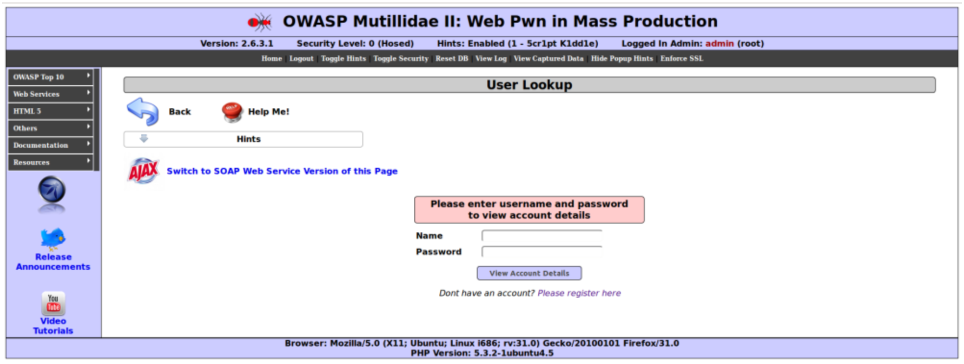
Edit cookie - Cookies Manager+

Name: uid
Content: 4
Domain: 10.20.1.199
Path: /mutillidae/
Send For: Any type of connection
Http Only: No
Expires: at end of session

- Latest Version
- Installation Instructions
- Usage Instructions
- Get rid of those pesky PHP errors
- Kali Linux
- Samurai Web Testing Framework
- sqlmap
- Some Useful Firefox Add-ons

Hints?: See "/documentation/mutillidae-test-scripts.txt"

Browser: Mozilla/5.0 (X11; Ubuntu; Linux i686; rv:31.0) Gecko/20100101 Firefox/31.0
PHP Version: 5.3.2-1ubuntu4.5


OWASP Mutillidae II: Web Pwn in Mass Production

Version: 2.6.3.1 Security Level: 0 (Hosed) Hints: Enabled (1 - 5cr1pt K1dd1e) Logged In Admin: admin (root)

Home | Logout | Toggle Hints | Toggle Security | Reset DB | View Log | View Captured Data | Hide Popup Hints | Enforce SSL
OWASP Top 10
Web Services
HTML 5
Others
Documentation
Resources
Release Announcements
Video Tutorials
Back
Help Me!
Hints
Switch to SOAP Web Service Version of this Page
View Account Details

User Lookup

Please enter username and password to view account details

Name

Password

Dont have an account? Please register here

Browser: Mozilla/5.0 (X11; Ubuntu; Linux i686; rv:31.0) Gecko/20100101 Firefox/31.0
PHP Version: 5.3.2-1ubuntu4.5

Countermeasures [9]

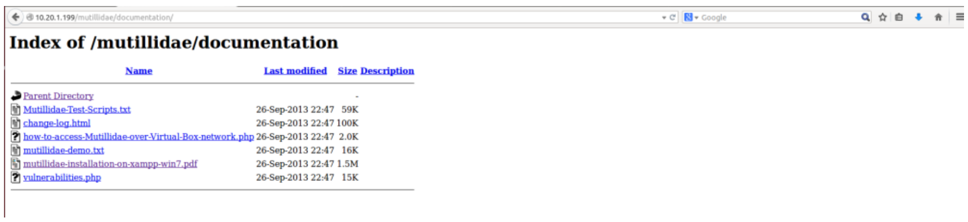
1.requires SSL for sensitive pafes whereas these pages should be redirected to SSL page for Non-SSL requests

2.Configure your SSL provider to only support strong algorithms

3.We must ensure that a valid certificate which has not expired is used which matches the domain used

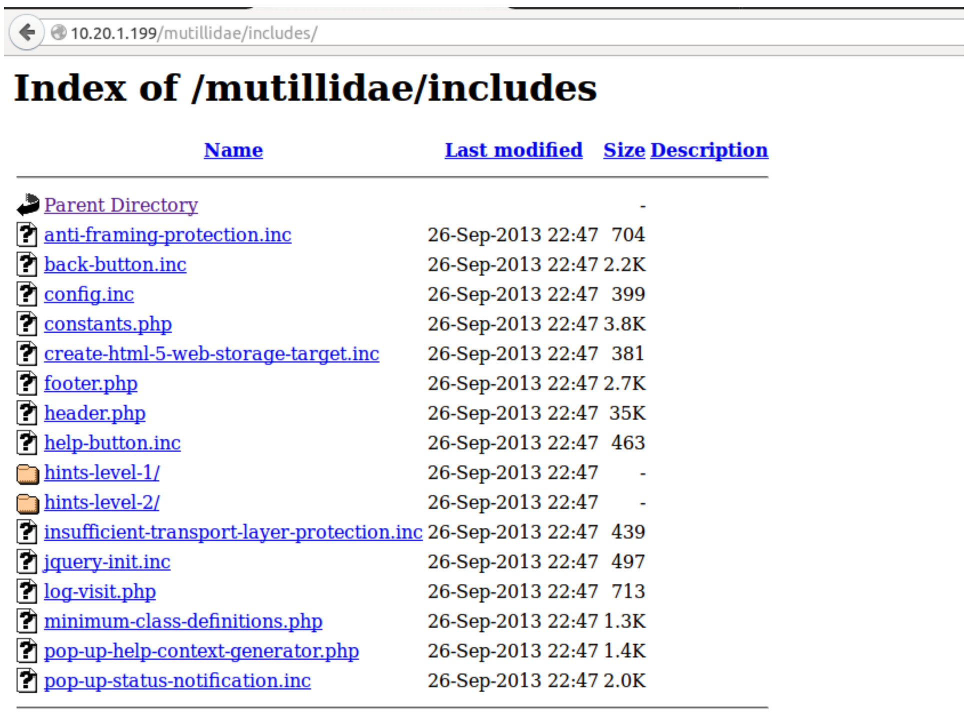
9. Security Misconfiguration-Directory Browsing

Directory browsing shows the contents of directories on the server to the user when the web server is misconfigured. Using search engines to look for pages which include "index of" in the title and reading the robot.txt file and spider the application with a tool such as Burp-site, OWASPZAP, named by search engines. While spidering inspect the folders named in robots.txt.The sensitive information in mutillidae and passwords and documentation can be seen by editing the URL as shown in the below figure [10].


10.20.1.199/mutillidae/documentation/

Index of /mutillidae/documentation

Name	Last modified	Size	Description
Parent Directory		-	
Mutillidae-Test-Scripts.txt	26-Sep-2013 22:47	59K	
change-log.html	26-Sep-2013 22:47	100K	
how-to-access-Mutillidae-over-Virtual-Box-network.php	26-Sep-2013 22:47	2.0K	
mutillidae-demo.txt	26-Sep-2013 22:47	16K	
mutillidae-installation-on-xampp-win7.pdf	26-Sep-2013 22:47	1.5M	
vulnerabilities.php	26-Sep-2013 22:47	15K	



10.20.1.199/mutillidae/includes/

Index of /mutillidae/includes

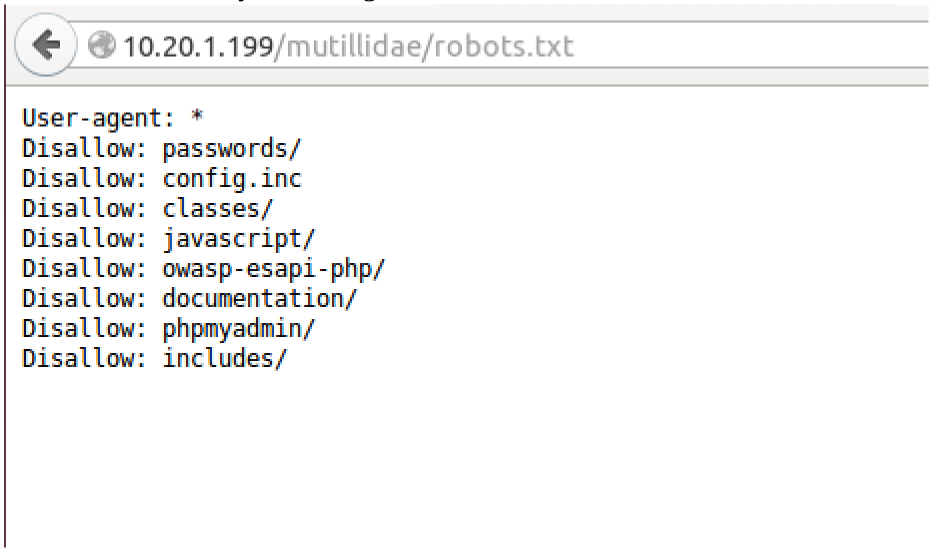
Name	Last modified	Size	Description
Parent Directory		-	
anti-framing-protection.inc	26-Sep-2013 22:47	704	
back-button.inc	26-Sep-2013 22:47	2.2K	
config.inc	26-Sep-2013 22:47	399	
constants.php	26-Sep-2013 22:47	3.8K	
create-html-5-web-storage-target.inc	26-Sep-2013 22:47	381	
footer.php	26-Sep-2013 22:47	2.7K	
header.php	26-Sep-2013 22:47	35K	
help-button.inc	26-Sep-2013 22:47	463	
hints-level-1/	26-Sep-2013 22:47	-	
hints-level-2/	26-Sep-2013 22:47	-	
insufficient-transport-layer-protection.inc	26-Sep-2013 22:47	439	
jquery-init.inc	26-Sep-2013 22:47	497	
log-visit.php	26-Sep-2013 22:47	713	
minimum-class-definitions.php	26-Sep-2013 22:47	1.3K	
pop-up-help-context-generator.php	26-Sep-2013 22:47	1.4K	
pop-up-status-notification.inc	26-Sep-2013 22:47	2.0K	

Countermeasures

A strong application architecture implementation and server configuration to prevent any unauthorized access [10].

10. Security Misconfiguration-Information Disclosure-robots.txt

Robot.txt file is used by sites to prevent web crawlers from indexing site content.Robots.txt is a plain text file which can be read by site visitors. In some cases, the robots.txt file will point to sensitive pages or directories. If a sensitive file is placed in robots.txt without proper authorization controls protecting the file, site visitors may discover the contents and browse to the files [http://10.20.1.199/mutillidae/index.php?page=robots-txt.php]. To exploit we need to follow the paths in the robot.txt file by browsing it in mutillidae.Robots.txt is located at localhost/mutillidae/robots.txt.



```
User-agent: *
Disallow: passwords/
Disallow: config.inc
Disallow: classes/
Disallow: javascript/
Disallow: owasp-esapi-php/
Disallow: documentation/
Disallow: phpmyadmin/
Disallow: includes/
```

Countermeasures

These files must not be filled with sensitive information and attackers who hack this can be discovered by google honeypot [].

REFLEXION REPORT

S NO:	TIME	KNOWLEDGE
Task1	**2 hours**	**SQL vulnerability in cms software and overcoming vulnerabilities**
Task2	**5 hours**	**Concurrency, denial of service ,forgot password attack**
Task3	**15 hours**	**Complex task where top 10 OWASP vulnerabilities were learned .**

REFERENCES

[1] https://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2016-6619

[2] https://www.phpmyadmin.net/security/PMASA-2016-42/

[3]http://cwe.mitre.org/data/definitions/89.html

[4] http://10.20.1.199/WebGoat/source?solution=true

[5] https://www.owasp.org/index.php/Main_Page

[6] https://www.owasp.org/index.php/Top_10_2010-A1-Injection

[7] https://www.owasp.org/index.php/Top_10_2010-A3

[8] http://yehg.net/lab/pr0js/papers/Disclosure%20Vulnerability%20PHPINFO.pdf

[9] https://www.owasp.org/index.php/Top_10_2010-A9-Insufficient_Transport_Layer_Protection

[10] https://www.owasp.org/index.php/Top_10_2010-A6-Security_Misconfiguration