

ASSIGNMENT 7

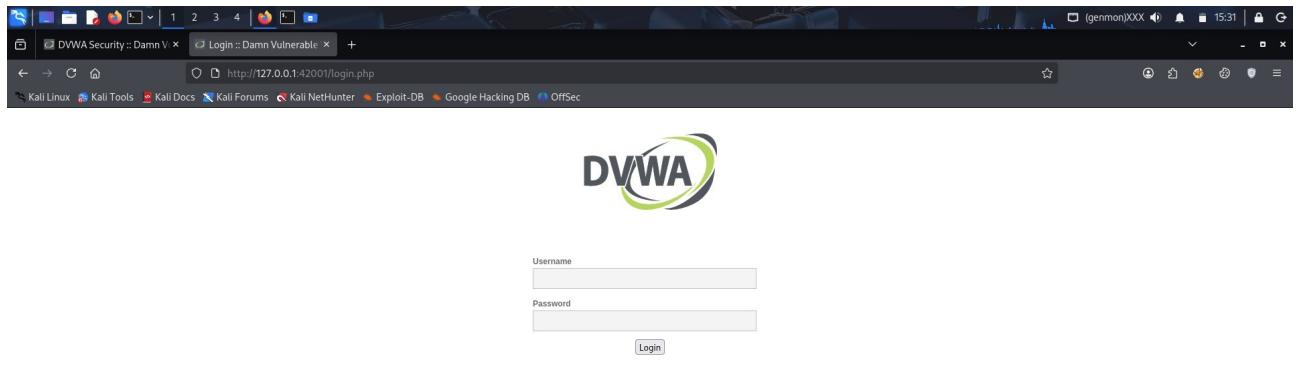
DVWA Installation + Vulnerability Discovery

Step 1 - DVWA Installation

Installed DVWA on Kali using apt and started service.

Commands:

```
sudo apt update  
sudo apt install dvwa  
sudo dvwa-start
```



DVWA Security :: Damn V x Welcome :: Damn Vulnerable Web Application http://127.0.0.1:42001/index.php

Welcome to Damn Vulnerable Web Application!

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

The aim of DVWA is to practice some of the most common web vulnerabilities, with various levels of difficulty, with a simple straightforward interface.

General Instructions

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

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

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

WARNING!

Damn Vulnerable Web Application is damn vulnerable! Do not upload it to your hosting provider's public html folder or any Internet facing servers, as they will be compromised. It is recommend using a virtual machine (such as [VirtualBox](#) or [VMware](#)), which is set to NAT networking mode. Inside a guest machine, you can download and install [LAMP2](#) for the web server and database.

Disclaimer

We do not take responsibility for the way in which any one uses this application (DVWA). 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 person/s who uploaded and installed it.

More Training Resources

DVWA aims to cover the most commonly seen vulnerabilities found in today's web applications. However there are plenty of other issues with web applications. Should you wish to explore any additional attack vectors, or want more difficult challenges, you may wish to look into the following other projects:

- [Mutillidae](#)
- [OWASP Vulnerable Web Applications Directory](#)

Step 2 - Security Levels

Changed DVWA security to Low → Medium → High and tested again.

DVWA Security :: Damn V x DVWA Security :: Damn V x http://127.0.0.1:42001/security.php

DVWA Security

Security Level

Security level is currently: **low**.

You can set the security level to low, medium, high or impossible. The security level changes the vulnerability level of DVWA:

1. Low - This security level is completely vulnerable and has no security measures at all. It's use is to be as an example of how web applications can be exploited through bad coding practices and to serve as a platform to teach basic web application exploitation techniques.
2. Medium - This setting is mainly to give an example to the user of **bad security practices**, where the developer has tried but failed to secure an application. It also acts as a challenge to users to refine their exploitation techniques.
3. High - This setting is an extension to the medium difficulty, with a mixture of **harder or alternative bad practices** to attempt to secure the code. The vulnerability may not allow the same extent of the exploitation similar in various Capture The Flags (CTFs) competitions.
4. Impossible - This level should **not** be **secure against all vulnerabilities**. It is used to compare the vulnerable source code with the secure source code.

Prior to DVWA v1.9, this level was known as **high**.

Username: admin
Security Level: low

Step 3 - Vulnerability Tests

1. SQL Injection

Performed basic SQL injection to bypass login and extract DB info.

Payloads:

1' OR '1'='1

The screenshot shows a Firefox browser window with the DVWA Security application open. The URL is `http://127.0.0.1:8001/vulnerabilities/sql/?id=1+OR+1%3D1&Submit=Submit#`. The DVWA logo is at the top. On the left is a sidebar menu with various security vulnerabilities listed. The 'SQL Injection' option is highlighted. The main content area is titled 'Vulnerability: SQL Injection'. It shows a user input field with the value '1' OR '1'='1'. Below it, several database rows are displayed, each with an ID, first name, and surname. The rows are:

ID	First name	Surname
1 OR 1='1	admin	admin
1 OR 1='1	Gordon	Brown
1 OR 1='1	Hack	Me
1 OR 1='1	Pablo	Picasso
1 OR 1='1	Bob	Smith

Below the table, there's a 'More Information' section with links to external resources about SQL injection. At the bottom of the page, it says 'Username: admin' and 'Security Level: low'.

2. Cross-Site Scripting (XSS)

Injected JavaScript to confirm reflected and stored XSS.

Payloads:

<h1>vikram<h1>

The screenshot shows the DVWA Reflected XSS page. In the top navigation bar, there are tabs for Home, Instructions, Setup / Reset DB, Brute Force, Command Injection, CSRF, File Inclusion, File Upload, Insecure CAPTCHA, SQL Injection, SQL Injection (Blind), Weak Session IDs, XSS (DOM), XSS (Reflected) (which is highlighted in green), XSS (Stored), CSP Bypass, JavaScript, Authorisation Bypass, Open HTTP Redirect, Cryptography, DVWA Security, PHP Info, and About. The Logout button is at the bottom of the sidebar. The main content area has a title "Vulnerability: Reflected Cross Site Scripting (XSS)". A form asks "What's your name?" with a value of "<h1>vikram</h1>". Below the form, the text "Hello vikram" is displayed in red. A "More Information" section lists several links related to XSS. At the bottom, it says "Username: admin" and "Security Level: low". There are "View Source" and "View Help" buttons.

```
<script>alert(1)</script>
```

The screenshot shows a browser window with the same URL as the previous screenshot. A modal dialog box is centered on the screen with the text "127.0.0.1:42001" and "1" below it. An "OK" button is at the bottom right of the dialog. The status bar at the bottom left shows "Read 127.0.0.1".

3. Command Injection

Executed system commands via vulnerable parameter.

Payloads:

127.0.0.1; ls

The screenshot shows a browser window for DVWA (Damn Vulnerable Web Application) version 1.0.4. The URL is <http://127.0.0.1:42001/vulnerabilities/exec/>. The page title is "Vulnerability: Command Injection". On the left, there's a sidebar menu with various exploit categories like Home, Instructions, Setup / Reset DB, Brute Force, Command Injection (which is highlighted in green), CSRF, File Inclusion, File Upload, Insecure CAPTCHA, SQL Injection, SQL Injection (Blind), Weak Session IDs, XSS (DOM), XSS (Reflected), XSS (Stored), CSP Bypass, JavaScript, Authorisation Bypass, Open HTTP Redirect, Cryptography, DVWA Security, PHP Info, About, and Logout. The main content area has a form titled "Ping a device" with a text input field containing "127.0.0.1; ls" and a "Submit" button. Below the form, the output of the ping command is displayed:
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=0.033 ms
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.043 ms
64 bytes from 127.0.0.1: icmp_seq=3 ttl=64 time=0.043 ms
64 bytes from 127.0.0.1: icmp_seq=4 ttl=64 time=0.059 ms
... 127.0.0.1 ping statistics ...
4 packets transmitted, 4 received, 0% packet loss, time 3057ms
rtt min/avg/max/mdev = 0.033/0.042/0.050/0.006 ms
help
index.php
source

4. File Upload Vulnerability

Uploaded test PHP file by bypassing file extension filters.

Payload:

```
<?php echo "test"; ?>
```

Rename as: shell.php.jpg

The screenshot shows the DVWA (Damn Vulnerable Web Application) interface. On the left, a sidebar lists various security vulnerabilities: Home, Instructions, Setup / Reset DB, Brute Force, Command Injection, CSRF, File Inclusion, **File Upload**, Insecure CAPTCHA, SQL Injection, SQL Injection (Blind), Weak Session IDs, XSS (DOM), XSS (Reflected), XSS (Stored), CSP Bypass, JavaScript, Authorisation Bypass, Open HTTP Redirect, Cryptography, DVWA Security, PHP Info, About, and Logout. The 'File Upload' option is currently selected. The main content area is titled 'Vulnerability: File Upload' and contains a form with a 'Browse...' button and an 'Upload' button. A message at the bottom says 'Your image was not uploaded.' Below this, a section titled 'More Information' provides links to external resources. A separate browser window in the foreground shows the source code of the upload script, which includes logic for moving uploaded files to a specific directory and echoing the result.

5. IDOR (Insecure Direct Object Reference)

Accessed other user info by modifying ID value.(performed on THM)

URL Test:

?id=10

The screenshot shows a browser developer tools interface with the 'Response' tab selected. The response is in JSON format, representing a user profile. The user_id is explicitly set to 10, while other fields like username, email, and address are populated from a different user's profile. This demonstrates how modifying the ID parameter can lead to retrieving information from another user's account.

```

{
  "user_id": 10,
  "username": "niels",
  "email": "niels@webmail.thm",
  "firstname": "Niels",
  "lastname": "Tester",
  "id_number": "NIELS-001",
  "address1": "42 chill Street",
  "address2": "Apt 1",
  "city": "TryTown",
  "state": "THM",
  "postal_code": "42424",
  "country": "Netherlands",
  "children": (5)[{...}, {...}, {...}, {...}, {...}]
}

```

6. Security Misconfigurations / Headers

Checked missing security headers using curl.

Command:

```
curl -I http://localhost/dvwa
```

bhai@KALI: ~

```
[bhai@KALI: ~] $ curl -I http://127.0.0.1:42001/
HTTP/1.1 302 Found
Server: nginx/1.28.0
Date: Thu, 13 Dec 2025 10:58:45 GMT
Content-Type: text/html; charset=UTF-8
Connection: keep-alive
Set-Cookie: security-impossible; path=/; HttpOnly
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Set-Cookie: PHPSESSID=e5ecb77b3df7166de9b4cbf9a56fcebc; expires=Fri, 12 Dec 2025 10:58:45 GMT; Max-Age=86400; path=/; domain=127.0.0.1; HttpOnly; SameSite=Strict
Location: login.php

[bhai@KALI: ~] $ curl -I http://127.0.0.1:42001/
HTTP/1.1 302 Found
Server: nginx/1.28.0
Date: Thu, 13 Dec 2025 11:33:15 GMT
Content-Type: text/html; charset=UTF-8
Connection: keep-alive
Set-Cookie: security-impossible; path=/; HttpOnly
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Set-Cookie: PHPSESSID:44e0fbe9cb20ff99836751363d13853; expires=Fri, 12 Dec 2025 11:33:15 GMT; Max-Age=86400; path=/; domain=127.0.0.1; HttpOnly; SameSite=Strict
Location: login.php

[bhai@KALI: ~] $
```

7. Sensitive Info in Source / JS

Viewed page source to identify exposed info.

Vulnerability: Reflected XSS - Scripting (XSS)

What's your name? Submit

Hello <script>alert(1)</script>

More Information

- <https://owasp.org/www-community/attacks/xss/>
- <https://owasp.org/www-community/xss-filter-evasion-cheatsheet>
- https://en.wikipedia.org/w/index.php?title=Cross-site_scripting&oldid=11700000

Element Inspector (main.css:374)

```
element { font: 100% Arial, sans-serif; vertical-align: middle; }
```

Element Inspector (main.css:147)

```
div#main { font-size: 1px; }
```

Element Inspector (main.css:725)

```
div#container { font-size: 1px; }
```

Element Inspector (main.css:1)

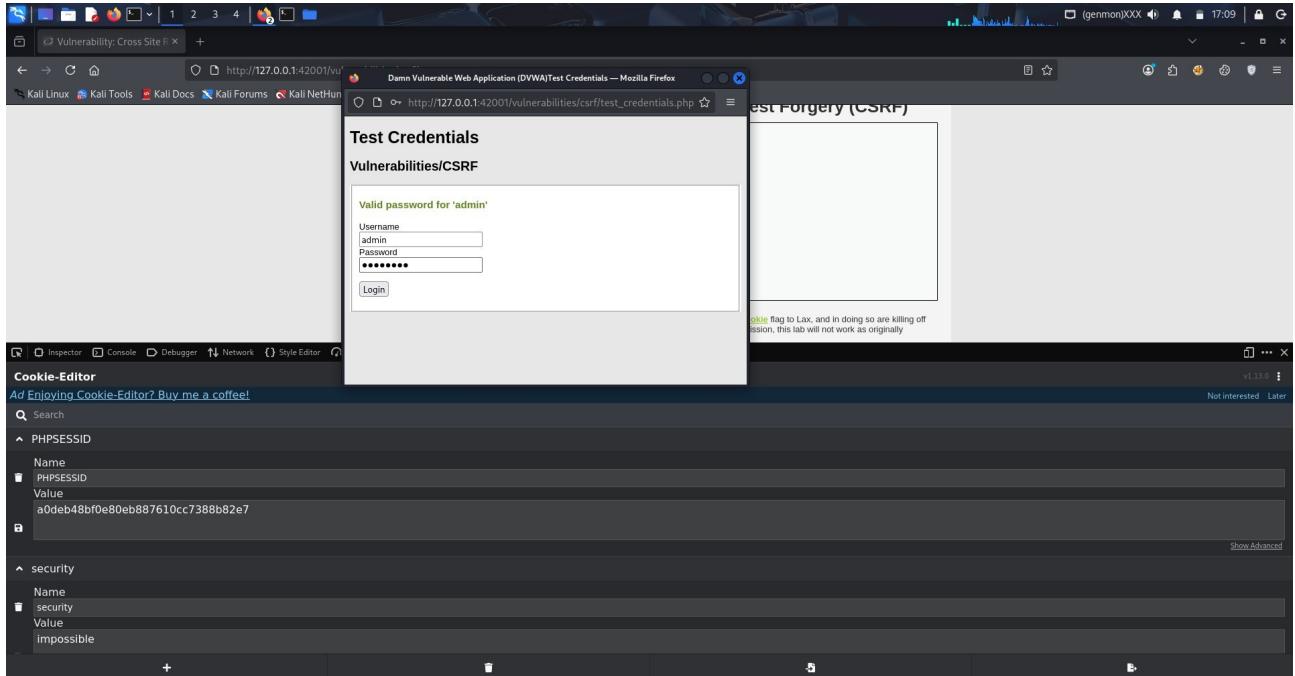
```
body { color: #212121; font-family: "Open Sans", Arial, Helvetica, sans-serif; }
```

Box Model Properties

margin	0
border	1
padding	40.466715px
width	428px
height	2px
border-radius	0px

8. Session Management Issues

Verified insecure cookies and missing flags.



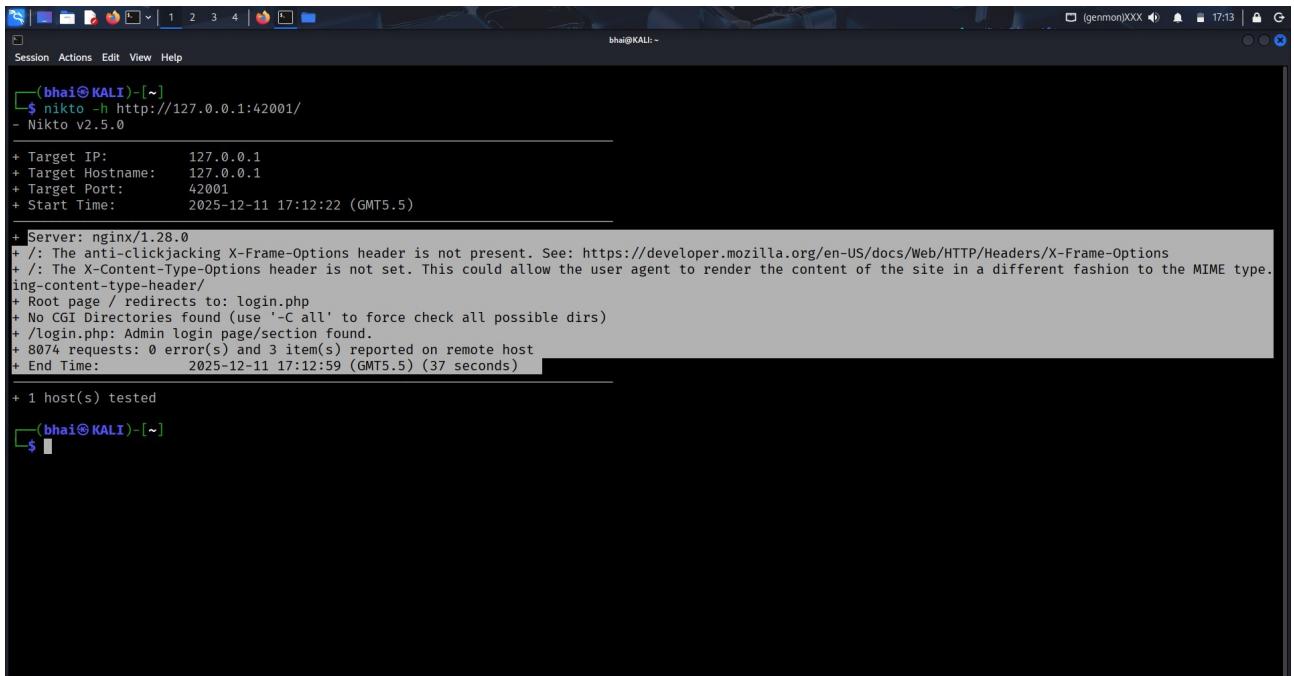
Step 4 - Light Automation

Nikto Scan

Scanned server for misconfigurations.

Command:

```
nikto -h http://localhost/dvwa
```



```
bhai@KALI:~$ nikto -h http://127.0.0.1:42001/
- Nikto v2.5.0

+ Target IP:      127.0.0.1
+ Target Hostname: 127.0.0.1
+ Target Port:    42001
+ Start Time:    2025-12-11 17:12:22 (GMT5.5)

+ Server: nginx/1.28.0
+ /: The anti-clickjacking X-Frame-Options header is not present. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options
+ /: The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type.
+ Content-type-header/
+ Root page / redirects to: login.php
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ /login.php: Admin login page/section found.
+ 8074 requests: 0 error(s) and 3 item(s) reported on remote host
+ End Time:       2025-12-11 17:12:59 (GMT5.5) (37 seconds)

+ 1 host(s) tested

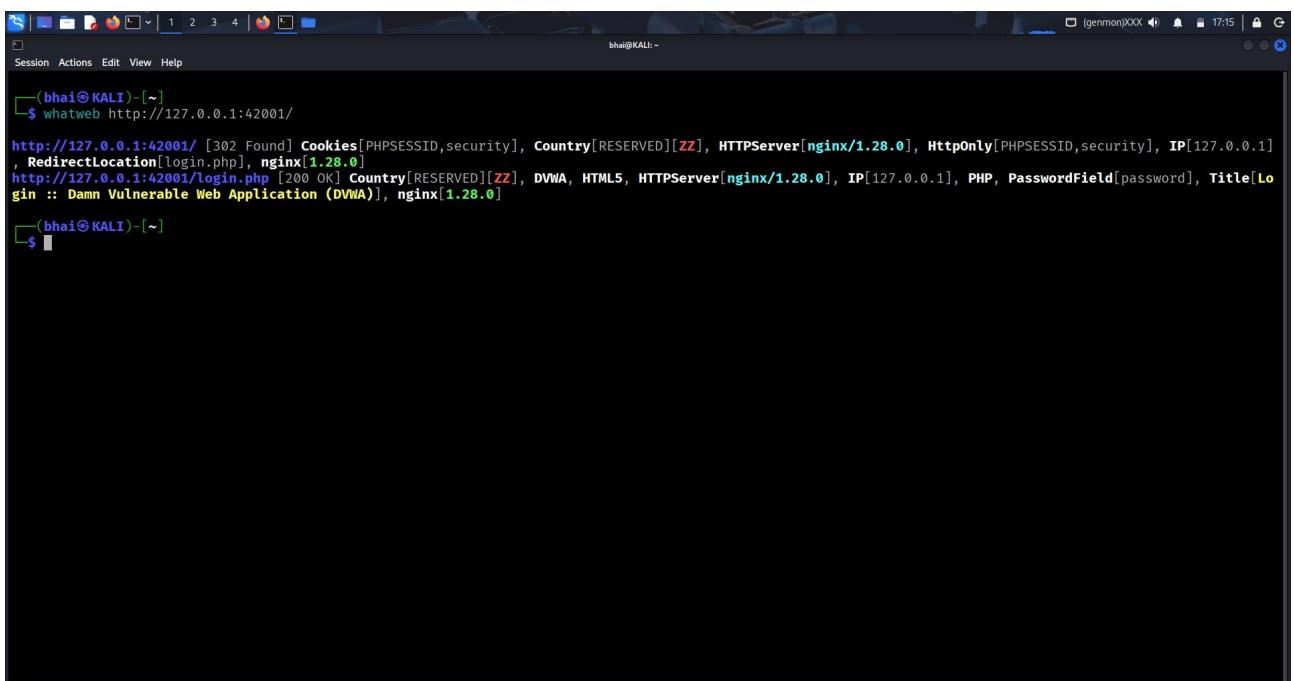
bhai@KALI:~$
```

WhatWeb Fingerprinting

Identified technologies running on DVWA.

Command:

```
whatweb http://localhost/dvwa
```



```
bhai@KALI:~$ whatweb http://127.0.0.1:42001/
http://127.0.0.1:42001/ [302 Found] Cookies[PHPSESSID,security], Country[RESERVED][ZZ], HTTPServer[nginx/1.28.0], HttpOnly[PHPSESSID,security], IP[127.0.0.1]
, RedirectLocation[login.php], nginx[1.28.0]
http://127.0.0.1:42001/login.php [200 OK] Country[RESERVED][ZZ], DVWA, HTML5, HTTPServer[nginx/1.28.0], IP[127.0.0.1], PHP, PasswordField[password], Title[Log
in :: Damn Vulnerable Web Application (DVWA)], nginx[1.28.0]

bhai@KALI:~$
```

DIRB Directory Scan

Enumerated hidden directories and files.

Command:

```
dirb http://localhost/dvwa
```

The screenshot shows a terminal window on a Kali Linux desktop environment. The terminal title is 'bhei@KALI: ~'. The command entered is 'dirb http://127.0.0.1:42001/'. The output of the scan is displayed below:

```
DIRB v2.22
By The Dark Raver

START_TIME: Thu Dec 11 17:16:51 2025
URL_BASE: http://127.0.0.1:42001/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

GENERATED WORDS: 4612

--- Scanning URL: http://127.0.0.1:42001/ ---
→ DIRECTORY: http://127.0.0.1:42001/config/
→ DIRECTORY: http://127.0.0.1:42001/database/
→ DIRECTORY: http://127.0.0.1:42001/docs/
→ DIRECTORY: http://127.0.0.1:42001/external/
+ http://127.0.0.1:42001/favicon.ico (CODE:200|SIZE:1406)
+ http://127.0.0.1:42001/index.php (CODE:302|SIZE:0)
+ http://127.0.0.1:42001/php.ini (CODE:200|SIZE:154)
+ http://127.0.0.1:42001/phpinfo.php (CODE:302|SIZE:0)
+ http://127.0.0.1:42001/robots.txt (CODE:200|SIZE:25)

--- Entering directory: http://127.0.0.1:42001/config/ ---
--- Entering directory: http://127.0.0.1:42001/database/ ---

--- Entering directory: http://127.0.0.1:42001/docs/ ---
+ http://127.0.0.1:42001/docs/copyright (CODE:200|SIZE:1085)
→ DIRECTORY: http://127.0.0.1:42001/docs/graphics/

--- Entering directory: http://127.0.0.1:42001/external/ ---
```

Nuclei Scan

Ran templates for common misconfigurations.

Command:

```
nuclei -u http://localhost/dvwa
```

```
bhai@KALI:~$ nuclei -u http://127.0.0.1:42001/
   _/\_ \_/\_ / \_/\_ \_/\_
  / \_/\_ \_/\_ \_/\_ \_/\_ \
 / \_/\_ \_/\_ \_/\_ \_/\_ \
v3.4.10
projectdiscovery.io

[INF] Your current nuclei-templates v10.3.4 are outdated. Latest is v10.3.5
[WRN] Found 1 templates with syntax error (use -validate flag for further examination)
[INF] Current nuclei version: v3.4.10 (outdated)
[INF] Current nuclei-templates version: v10.3.4 (outdated)
[INF] New templates added in latest release: 0
[INF] Templates loaded for current scan: 8855
[INF] Executing 8853 signed templates from projectdiscovery/nuclei-templates
[WRN] Loading 2 unsigned templates for scan. Use with caution.
[INF] Targets loaded for current scan: 1
[INF] Templates clustered: 1853 (Reduced 1740 Requests)
[INF] Using Interactsh Server: oast.live
[dwba-default-login] [http] [critical] http://127.0.0.1:42001/index.php [password="password",username="admin"]
[cookies-without-secure] [javascript] [info] 127.0.0.1:42001 ["security","PHPSESSID"]
[external-service-interaction] [http] [info] http://127.0.0.1:42001/
[external-service-interaction] [http] [info] http://127.0.0.1:42001/
[waf-detect:nginxgeneric] [http] [info] http://127.0.0.1:42001/
[INF] Scan completed in 1m. 5 matches found.

(bhai@KALI)-[~]
$
```