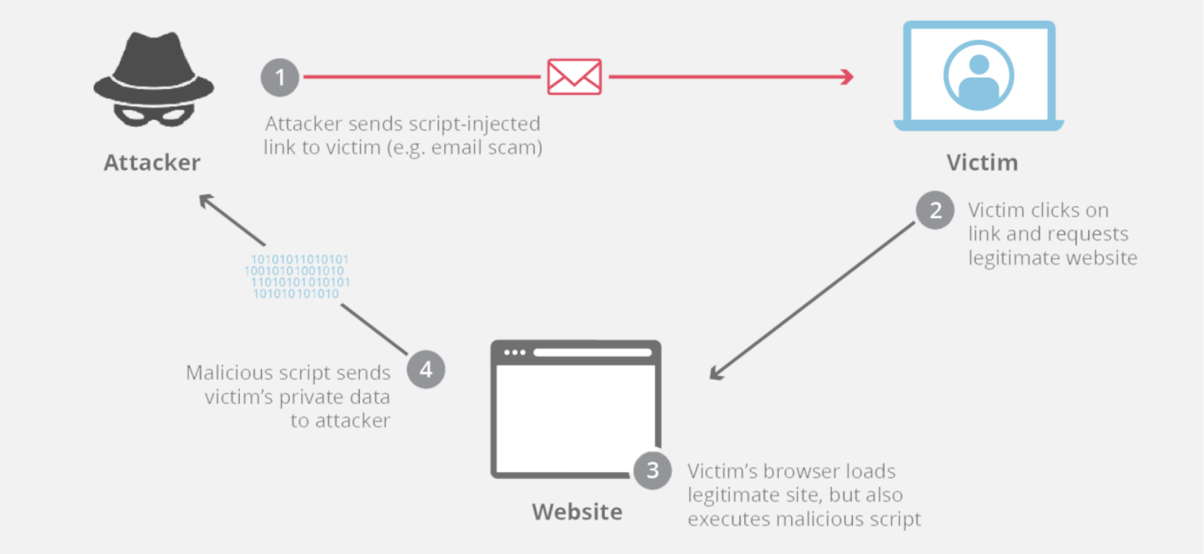
# Challenge 2: Web Server Vulnerabilities


1
Attacker sends script-injected link to victim (e.g. email scam)
Attacker
Victim
2
Victim clicks on link and requests legitimate website
10101011010101
10010101001010
11010101010101
101010101010
4
Malicious script sends victim's private data to attacker
3
Victim's browser loads legitimate site, but also executes malicious script
Website

In this part, you must find vulnerabilities on an HTTP server. Misconfiguration of a web server can allow for the listing of files contained in directories on the server. You can use any of the tools you learned in earlier labs to perform reconnaissance to find the vulnerable directories.

In this challenge, you will locate the flag file in a vulnerable directory on a web server.

## Step 1: Preliminary setup

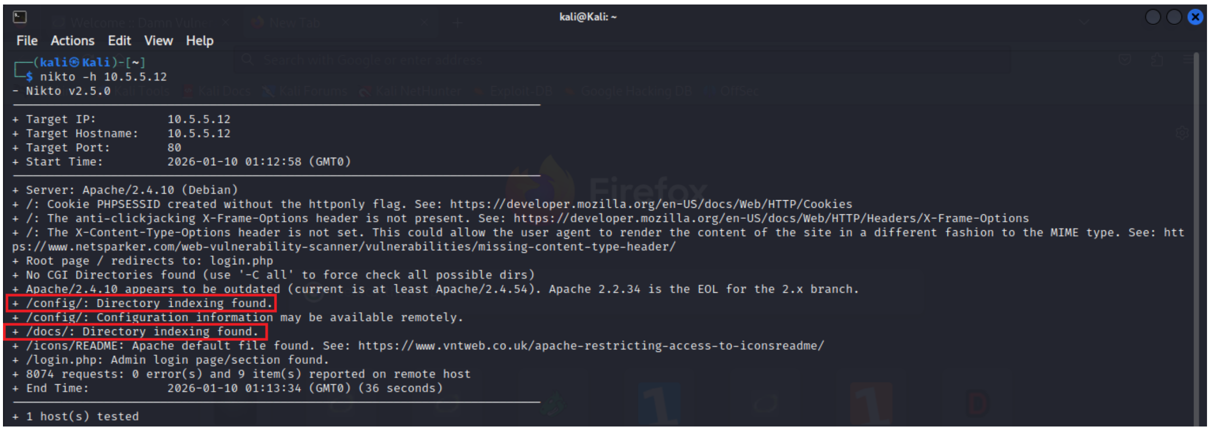
a. If not already, log into the server at 10.5.5.12 with the **admin / password** credentials.

b. Set the application security level to low.

## Step 2: From the results of your reconnaissance, determine which directories are viewable using a web browser and URL manipulation.

Perform reconnaissance on the server to find directories where indexing was found.

**Command: nikto -h 10.5.5.12**



```
┌──(kali㉿Kali)-[~]
└─$ nikto -h 10.5.5.12
- Nikto v2.5.0
---------------------------------------------------------------------------
+ Target IP:          10.5.5.12
+ Target Hostname:    10.5.5.12
+ Target Port:        80
+ Start Time:         2026-01-10 01:12:58 (GMT0)
---------------------------------------------------------------------------
+ Server: Apache/2.4.10 (Debian)
+ /: Cookie PHPSESSID created without the httponly flag. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/Cookies
+ /: 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. See: htt
ps://www.netsparker.com/web-vulnerability-scanner/vulnerabilities/missing-content-type-header/
+ Root page / redirects to: login.php
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ Apache/2.4.10 appears to be outdated (current is at least Apache/2.4.54). Apache 2.2.34 is the EOL for the 2.x branch.
+ /config/: Directory indexing found.
+ /config/: Configuration information may be available remotely.
+ /docs/: Directory indexing found.
+ /icons/README: Apache default file found. See: https://www.vntweb.co.uk/apache-restricting-access-to-iconsreadme/
+ /login.php: Admin login page/section found.
+ 8074 requests: 0 error(s) and 9 item(s) reported on remote host
+ End Time:           2026-01-10 01:13:34 (GMT0) (36 seconds)
---------------------------------------------------------------------------
+ 1 host(s) tested
```

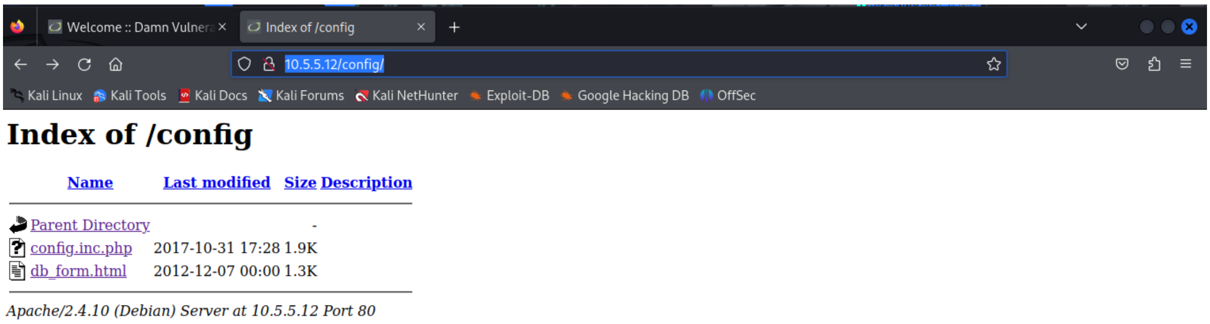
Which directories can be accessed through a web browser to list the files and subdirectories that they contain?

**/config/ and /docs/ can be accessed through a web browser to list the files and subdirectories that they contain.**

## Step 3: View the files contained in each directory to find the file containing the flag.

Create a URL in the web browser to access the viewable subdirectories. Find the file with the code for Challenge 2 located in one of the subdirectories.

## http://10.5.5.12/config/

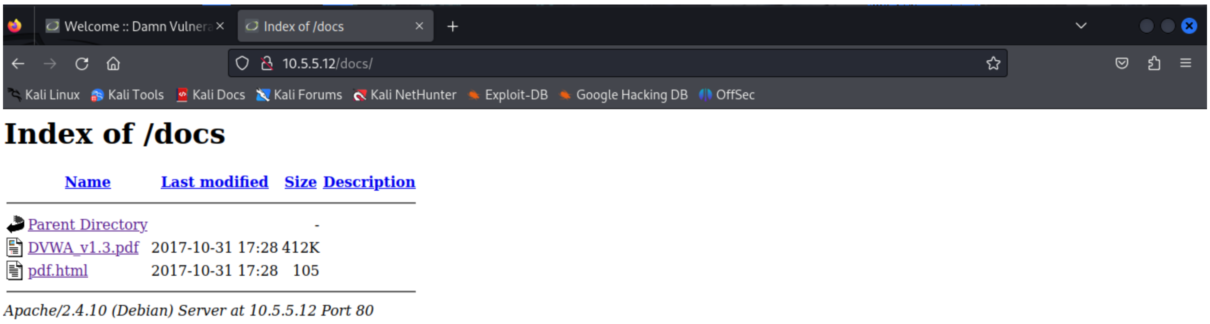


# Index of /config

| Name | Last modified | Size | Description |
|---|---|---|---|
| Parent Directory |  | - |  |
| config.inc.php | 2017-10-31 17:28 | 1.9K |  |
| db_form.html | 2012-12-07 00:00 | 1.3K |  |

*Apache/2.4.10 (Debian) Server at 10.5.5.12 Port 80*

## http://10.5.5.12/docs/

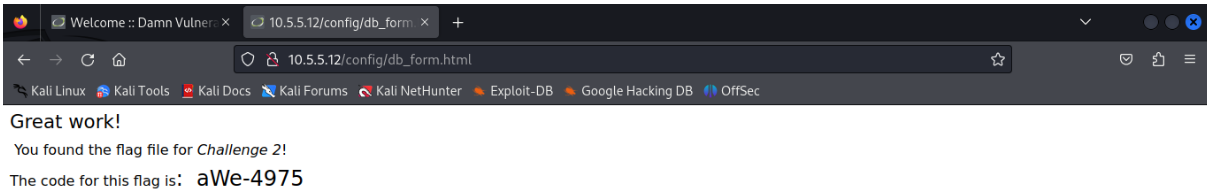


# Index of /docs

| Name | Last modified | Size | Description |
|---|---|---|---|
| Parent Directory |  | - |  |
| DVWA_v1.3.pdf | 2017-10-31 17:28 | 412K |  |
| pdf.html | 2017-10-31 17:28 | 105 |  |

*Apache/2.4.10 (Debian) Server at 10.5.5.12 Port 80*

## http://10.5.5.12/config/db_form.html



Great work!

You found the flag file for *Challenge 2*!

The code for this flag is: aWe-4975

In which two subdirectories can you look for the file?

**You look for the file in the /config/ and /docs/ and sub-directories.**

What is the filename with the Challenge 2 code?

**The filename with the Challenge 2 code is db_form.html**

Which subdirectory held the file?

**The /config/ subdirectory held the file.**

What is the message contained in the flag file? Enter the code that you find in the file.

**The message contained in the flag file is aWe-4975.**

**Step 4: Research and propose directory listing exploit remediation.**

**Missing Content-Type Header**

missing Content-Type header which means that this website could be at risk of a MIME-sniffing attacks.

What are two remediation methods for preventing directory listing exploits?

**The two remediation methods for preventing directory listing exploits are**

1. **When serving resources, make sure you send the content-type header to appropriately match the type of the resource being served. For example, if you are serving an HTML page, you should send the HTTP header:Content-Type: text/html**

2. **Add the X-Content-Type-Options header with a value of "nosniff" to inform the browser to trust what the site has sent is the appropriate content-type, and to not attempt "sniffing" the real content-type.X-Content-Type-Options: nosniff**