Final Capstone Activity

Objectives

For this Final Capstone Activity, you will conduct a complete penetration test starting with reconnaissance and then launching exploits against vulnerabilities that you have discovered. Finally, you will propose remediation for the exploits.

This assessment is in the form of a cybersecurity capture the flag exercise. You will use your ethical hacking skills to locate files that contain flag values. You will then report the flag values that you found as part of the assessment.

In this simulation of an ethical hacking engagement, you will use tools to exploit vulnerabilities that you discover in order to reach a goal. This can entail a trial-and-error approach that requires persistence and may include a degree of struggle. For your own skill development, working through this struggle can be productive. If you are completely stuck, ask your instructor for assistance.

- **Challenge 1** – Use SQL injection to find a flag file.
- **Challenge 2** – Use web server vulnerabilities to investigate directories and find a flag file.
- **Challenge 3** – Exploit open Samba shares to access a flag file.
- **Challenge 4** – Analyze a Wireshark capture file to find the location of a file containing flag information.

Background / Scenario

You have been hired to conduct a penetration test for a customer. At the conclusion of the test, the customer has requested a complete report that includes any vulnerabilities discovered, successful exploits, and remediation steps to protect vulnerable systems. You have access to hosts on the 10.5.5.0 and 192.168.0.0/24 networks.

Required Resources

- Kali VM customized for the Ethical Hacker course

Instructions

Challenge 1: SQL Injection

Total points: 25

In this part, you must discover user account information on a server and crack the password of **Bob Smith's** account. You will then locate the file with Challenge 1 code and use **Bob Smith's** account credentials to open the file at 192.168.0.10 to view its contents.

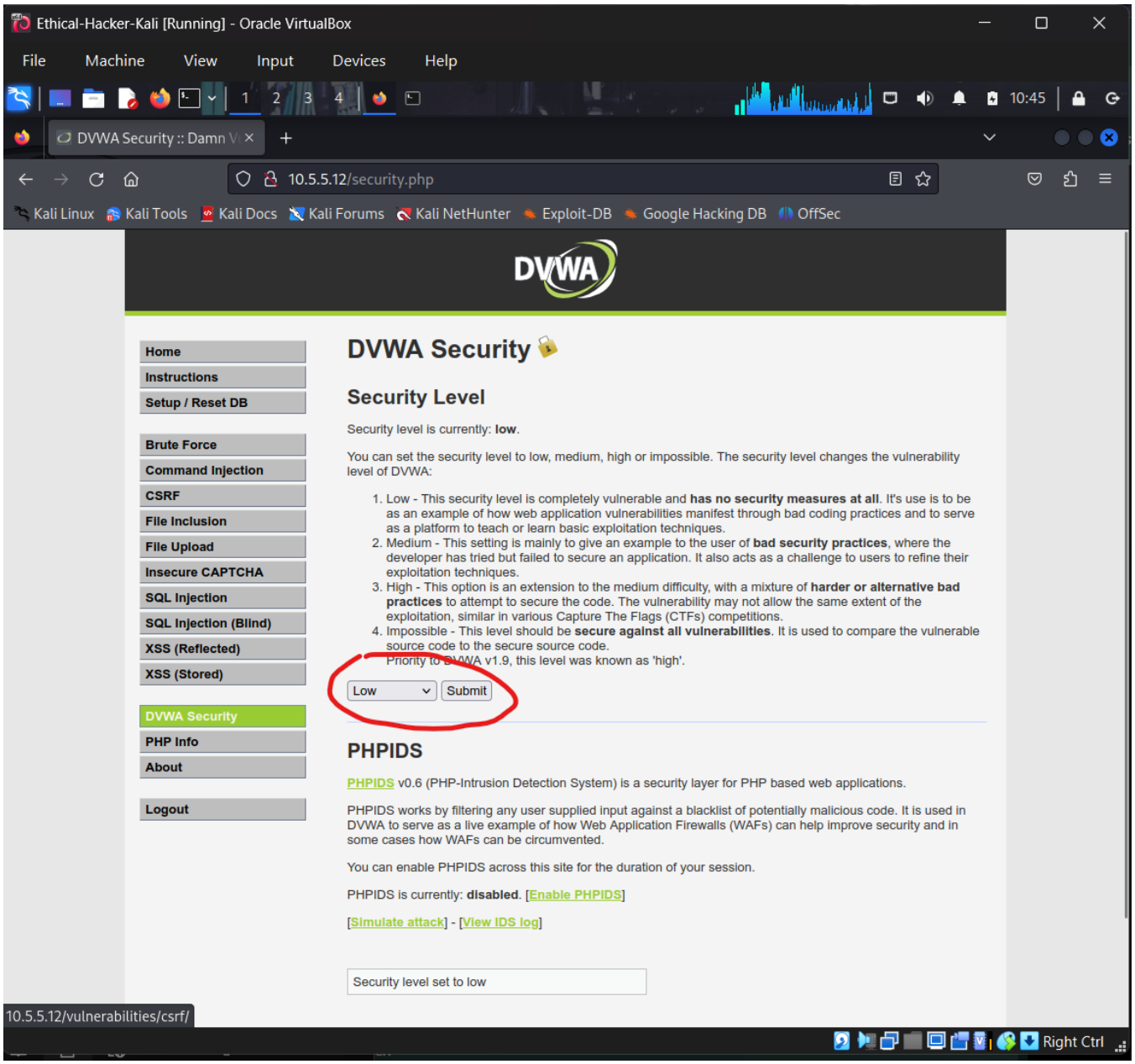
Step 1: Preliminary setup

a. Open a browser and go to the website at 10.5.5.12.

Note: If you have problems reaching the website, remove the https:// prefix from the IP address in the browser address field.

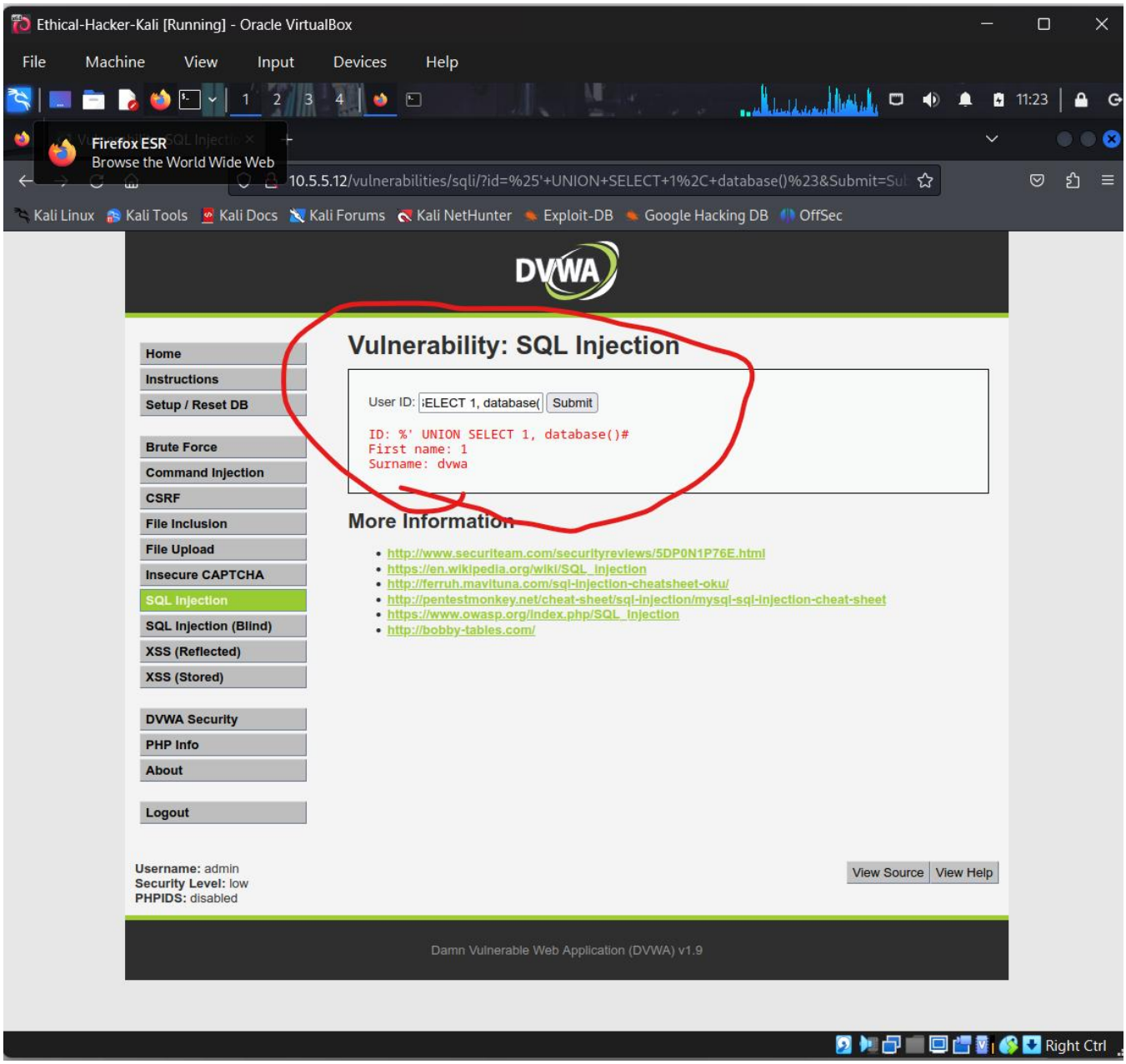
b. Login with the credentials **admin / password**.

c. Set the DVWA security level to **low** and click **Submit**.

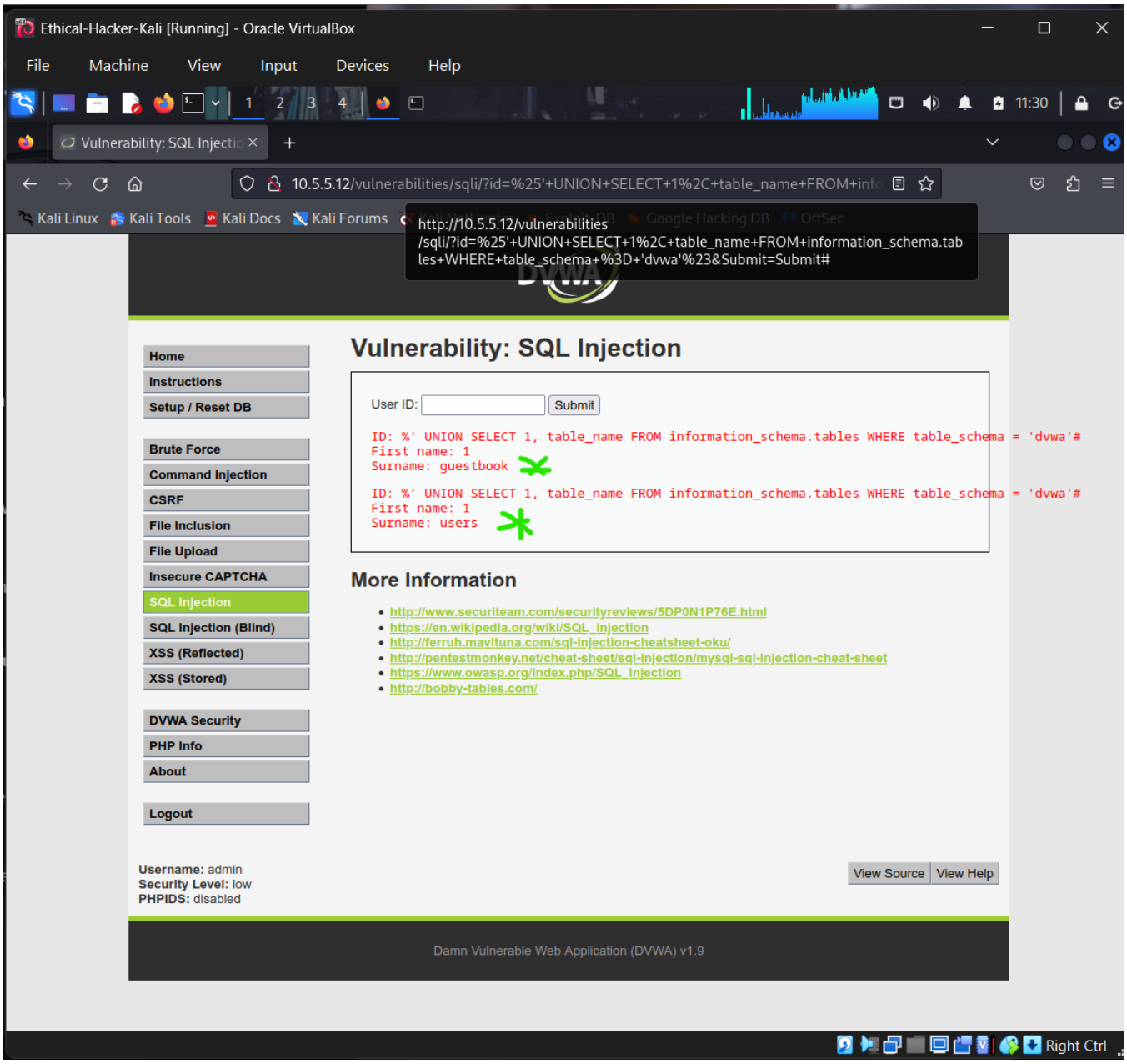


Step 2: Retrieve the user credentials for the Bob Smith's account.

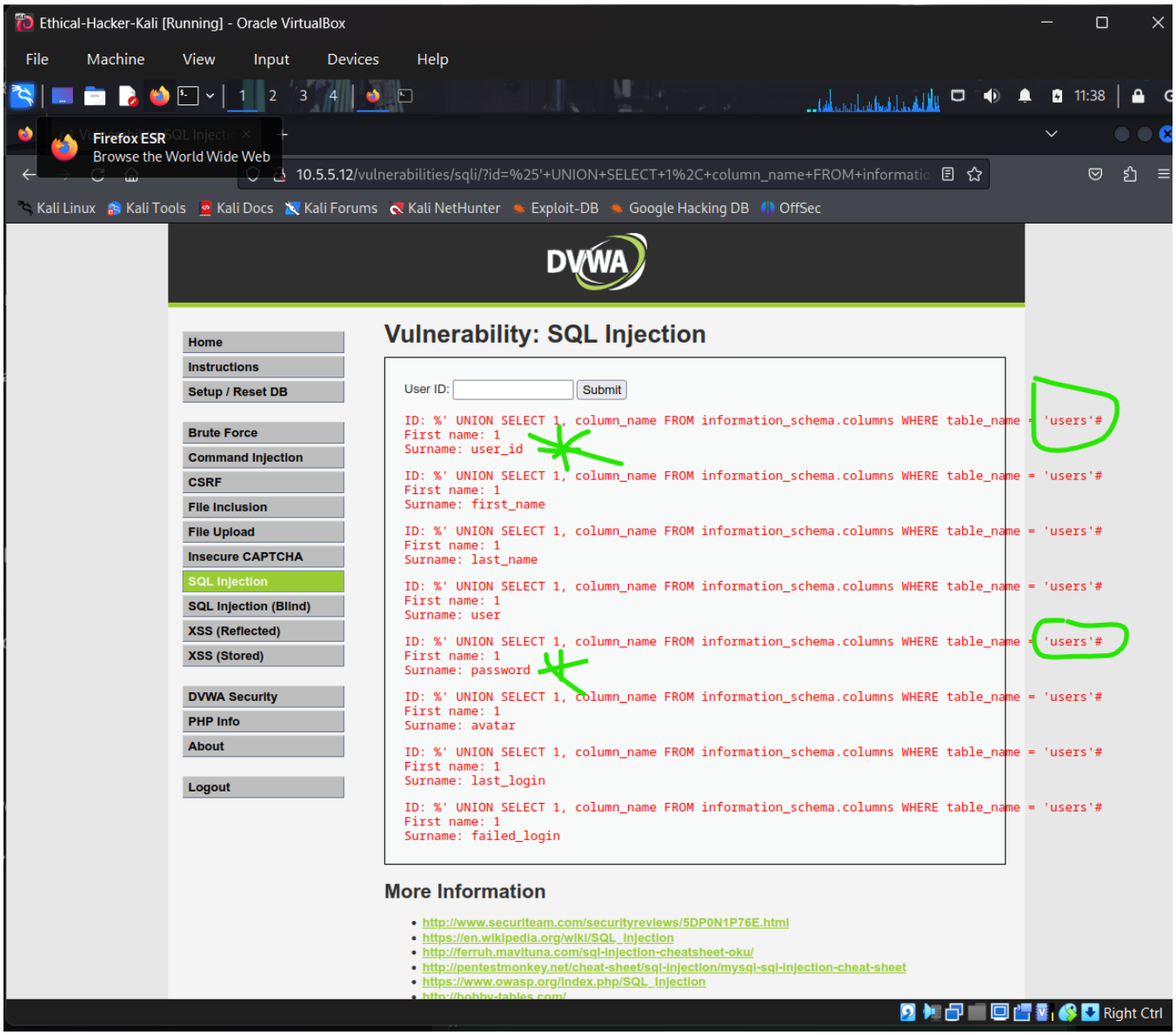
a. Identify the table that contains usernames and passwords.



Name of the database is ‘dvwa’.

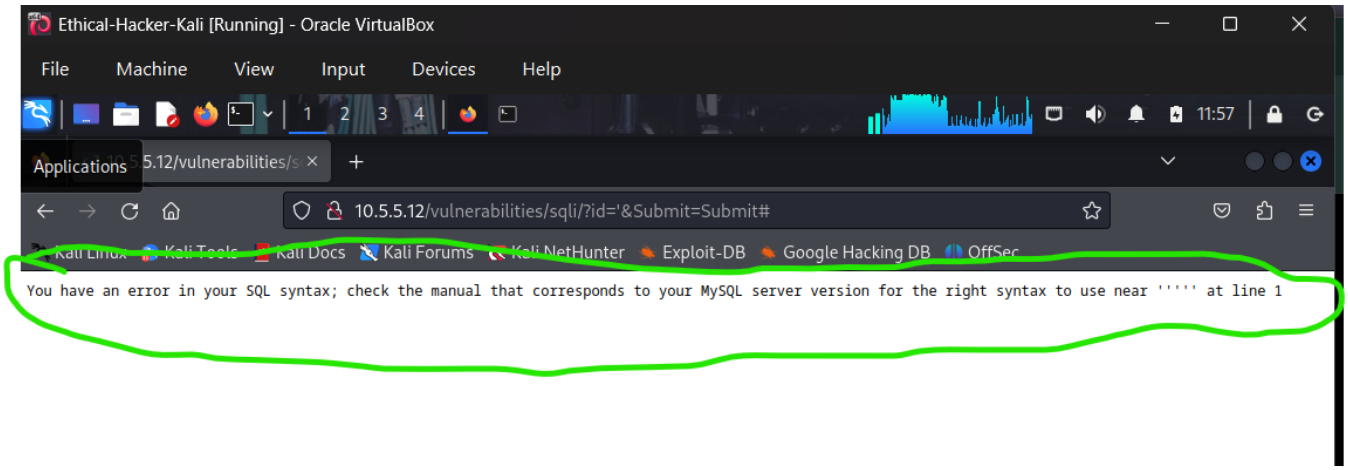


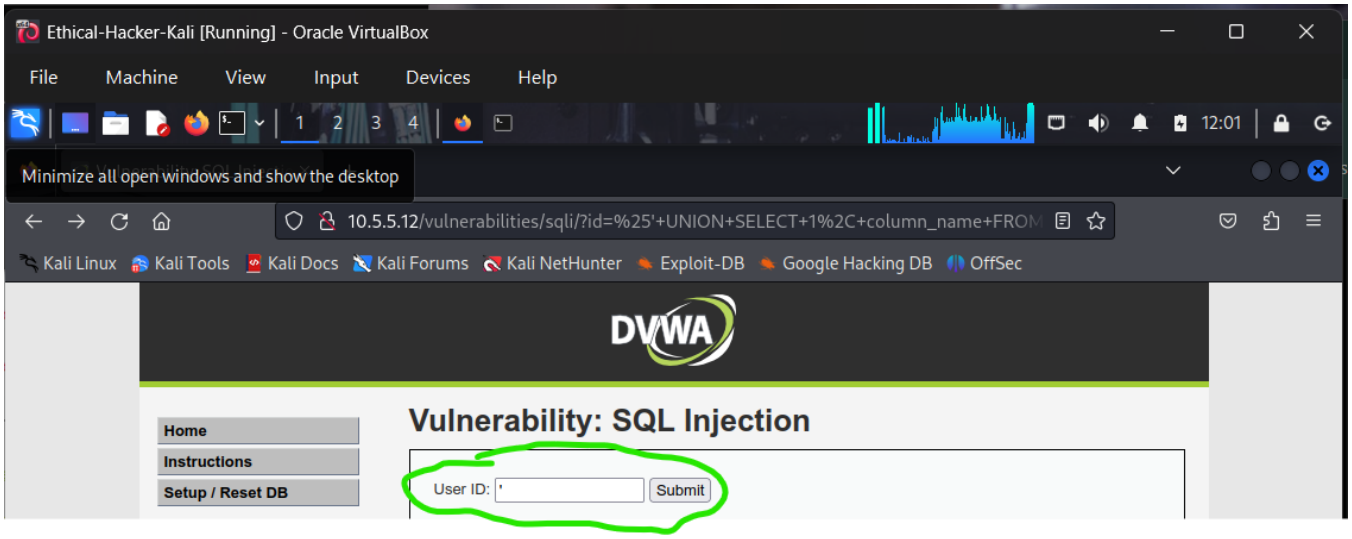
There are two tables in the dvwa database; ‘guestbook’ and ‘users’.



The ‘users’ table contains usernames and passwords.

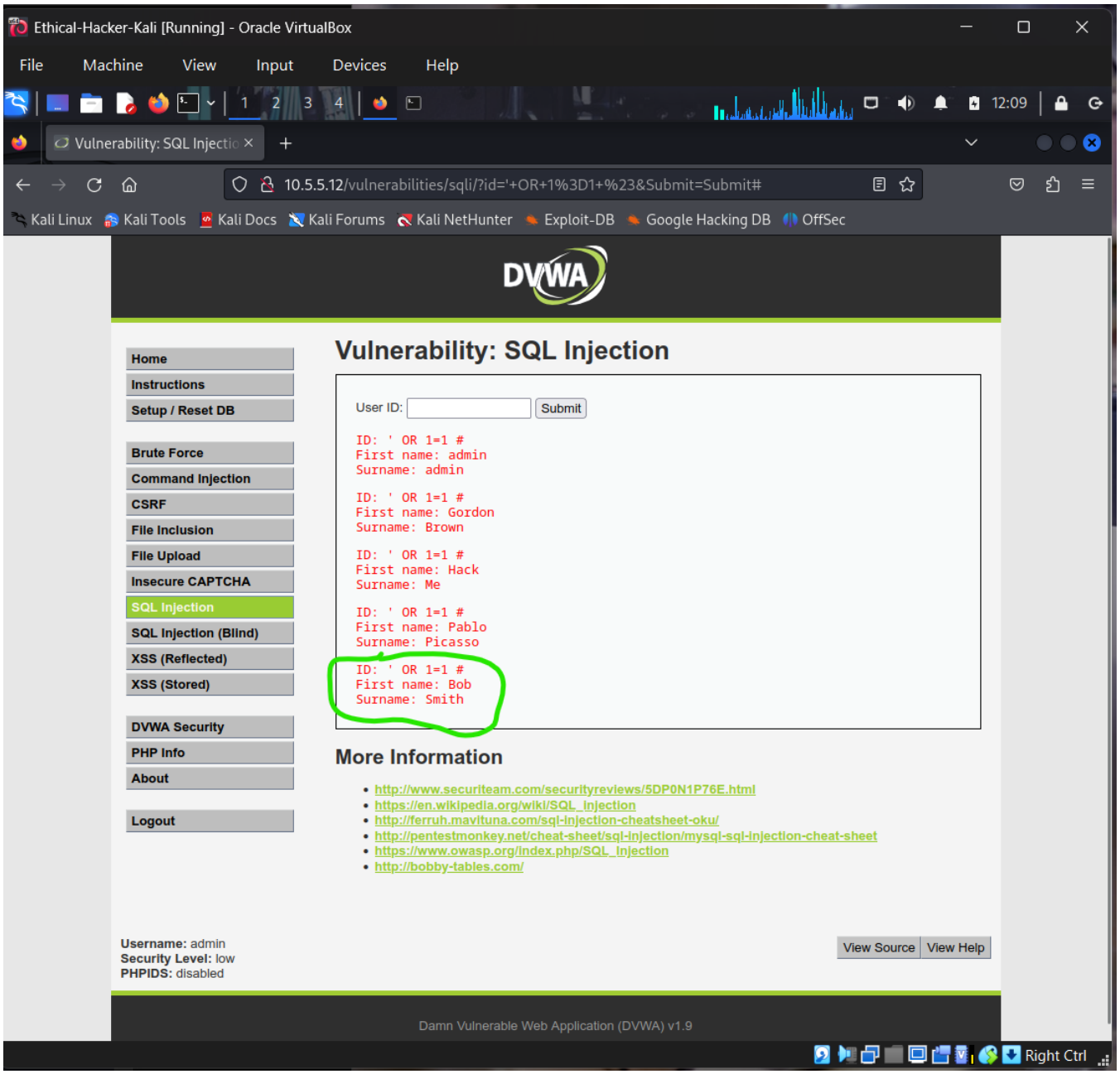
b. Locate a vulnerable input form that will allow you to inject SQL commands.



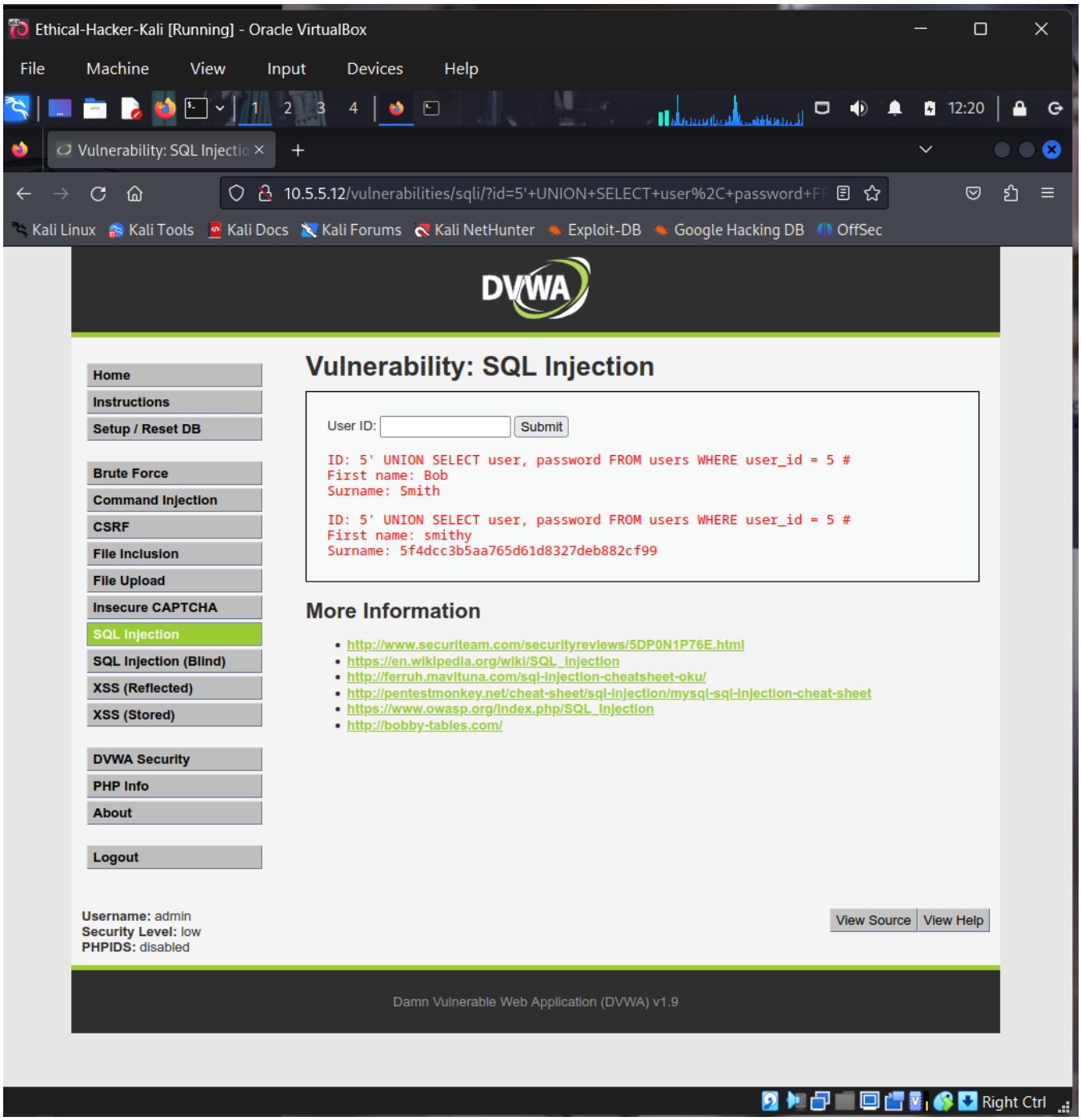


This specifies ‘User Id’ form is vulnerable.

c. Retrieve the username and the password hash for **Bob Smith's** account.



A confirmation of Bob Smith's entry in the database using the basic 'always true' payload " ' OR 1=1 #".



Username: smithy

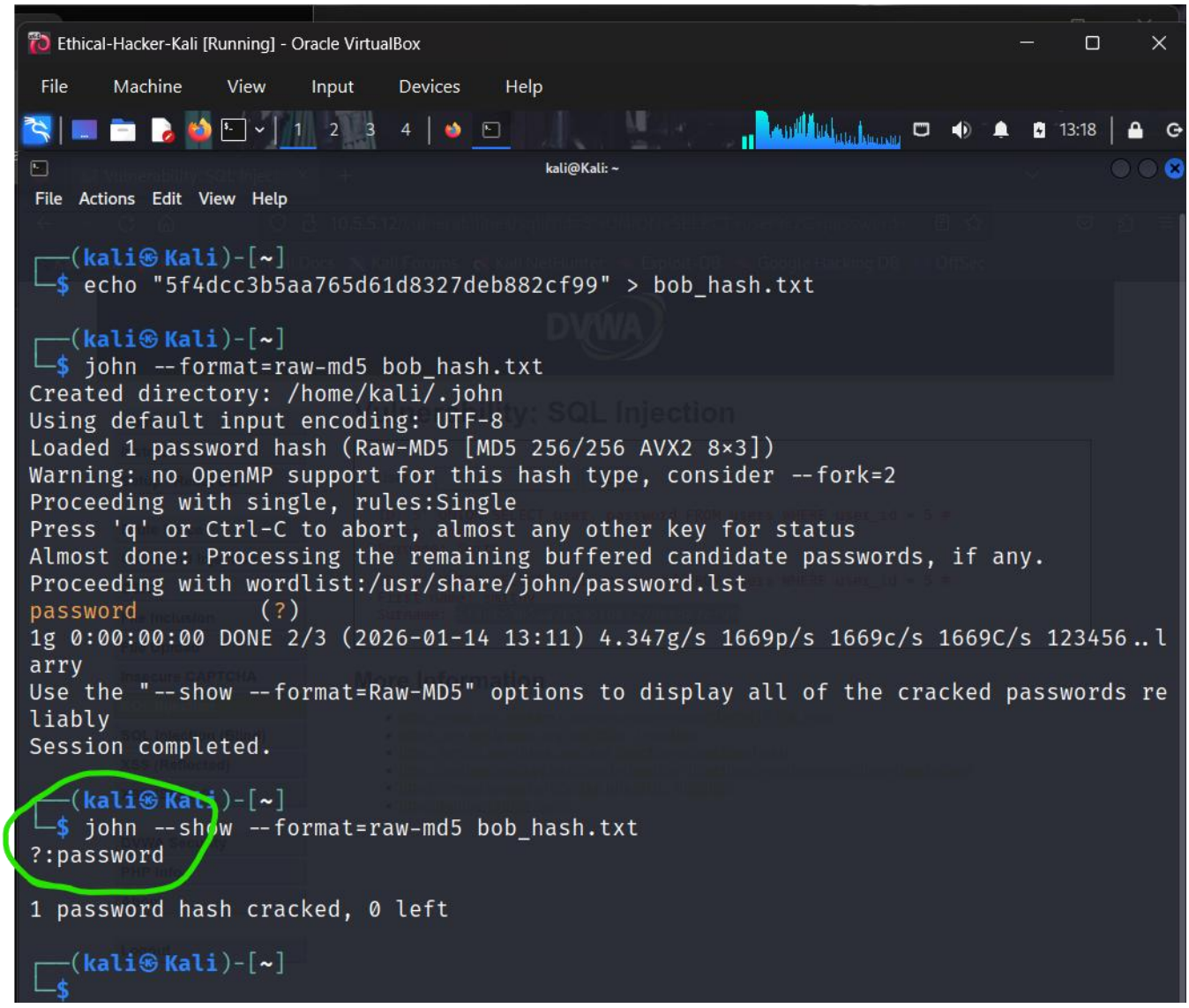
Password hash: 5f4dcc3b5aa765d61d8327deb882cf99

Step 3: Crack Bob Smith's account password.

Use any password hash cracking tool desired to crack **Bob Smith**'s password.

What is the password of **Bob Smith's** account?

Answer: password



I created a text file with the hash in it and used John the Ripper to perform the crack which gave me the plaintext 'password' for the hash.

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

a. Log into **192.168.0.10** as **Bob Smith**.

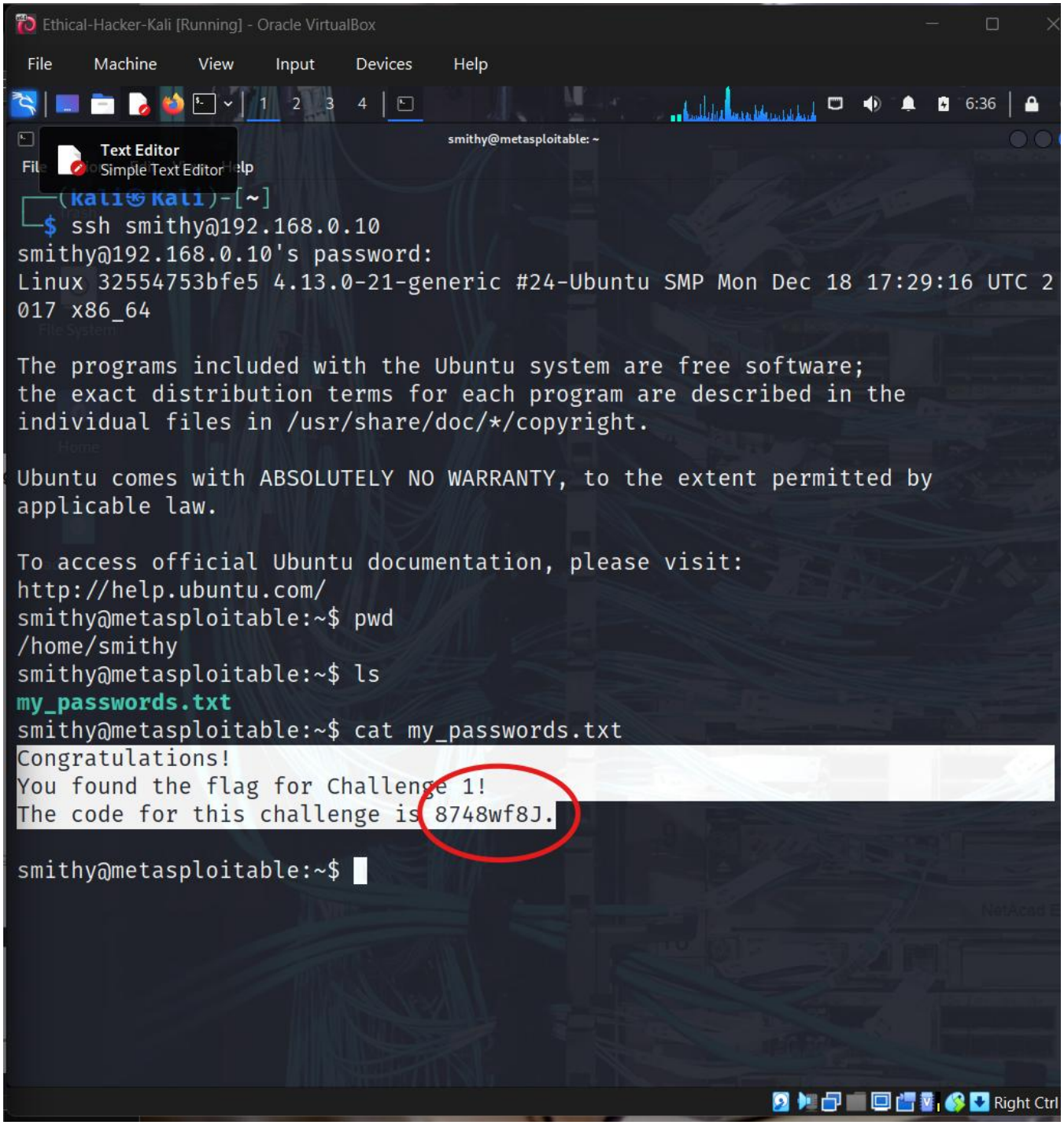
b. Locate and open the flag file in the user's home directory.

What is the name of the file with the code?

Answer: my_passwords.txt

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

Answer: Congratulations! You found the flag for Challenge 1! The code for this challenge is 8748wf8J.



Procedure:

- I ssh into smithy@192.xxx.x.10, using the cracked password.
- Used the command ‘pwd’ to confirm the directory.

- Used the command ‘ls’ to list all files available in the directory and I found just one; “my_passwords.txt”.
- I now used the cat command to open the file.

Step 5: Research and propose SQL attack remediation

What are five remediation methods for preventing SQL injection exploits?

Answer: *To prevent SQLi, it is recommended to use multi-layered defense strategies such as the following.*

- ***Parameterized Queries:*** *It involves pre-compiling the SQL query structure and using placeholders for user input. This ensures the database treats input strictly as data, not executable code, preventing even malicious payloads like ' OR '1'='1 from altering the query's intent.*
- ***Input Validation and Sanitization:*** *Treat all user-supplied data as untrusted. Use* ***allow-listing*** *to verify that input matches expected formats, types (e.g., numeric only for IDs), and lengths. Sanitization involves stripping or escaping dangerous characters (such as ', ;, or --) that could be used to manipulate queries.*
- ***Principle of Least Privilege:*** *Minimize the database permissions assigned to the web application’s account. If an application only needs to read data, grant it only SELECT privileges; do not use administrative accounts like root for daily operations.*
- ***Stored Procedures:*** *These encapsulate SQL logic within the database itself, limiting the exposure of raw SQL code to user inputs. However, these must be implemented safely (avoiding dynamic SQL generation within the procedure) to be effective.*
- ***Web Application Firewalls (WAF):*** *A WAF acts as a real-time shield by monitoring incoming HTTP traffic for known SQLi signatures and anomalous patterns. It can block automated attacks and provide "virtual patching" for vulnerabilities that haven't been fixed in the code yet.*

Challenge 2: Web Server Vulnerabilities

Total points: 25

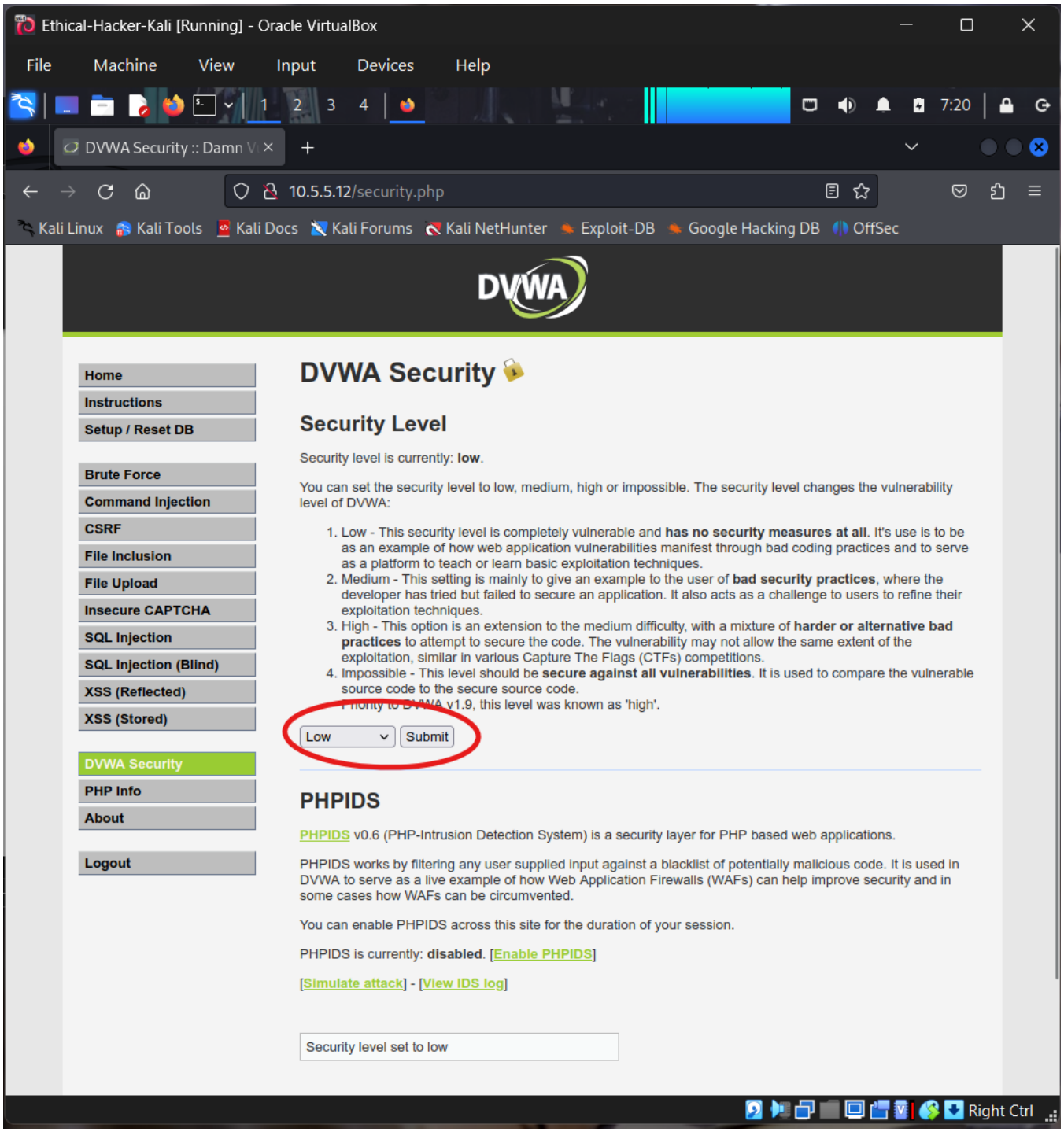
In this part, you must find vulnerabilities on an HTTP server. Misconfiguration of a web server can allow for the listing of files contained in directories on the server. You can use any of the tools you learned in earlier labs to perform reconnaissance to find the vulnerable directories.

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.



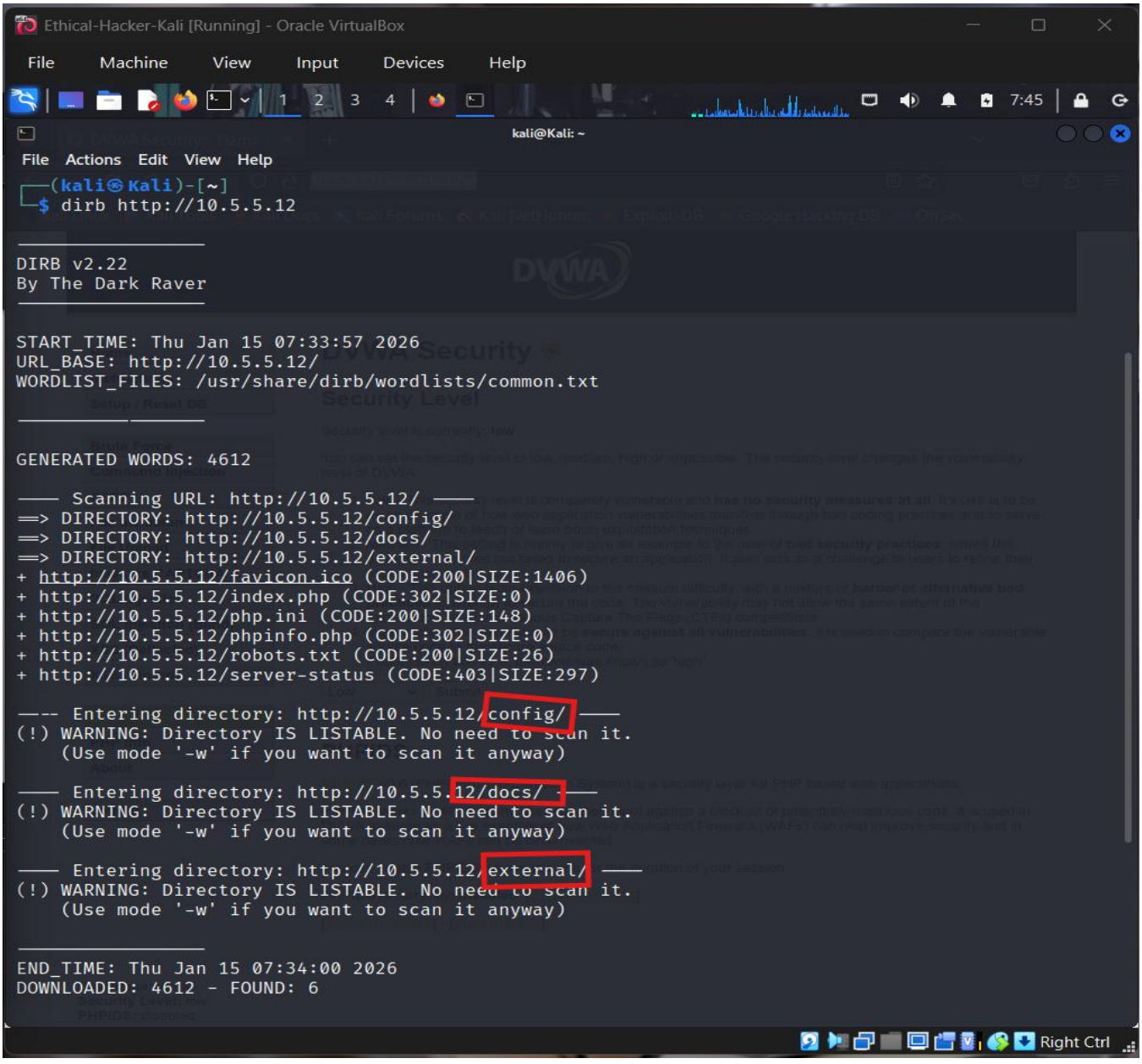
Step 2: From the results of your reconnaissance, determine which directories are viewable using a web browser and URL manipulation.

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

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

The following directories can be accessed through a web browser to list the files and subdirectories that they contain.

- **/config/**
- **/docs/**
- **/external/**

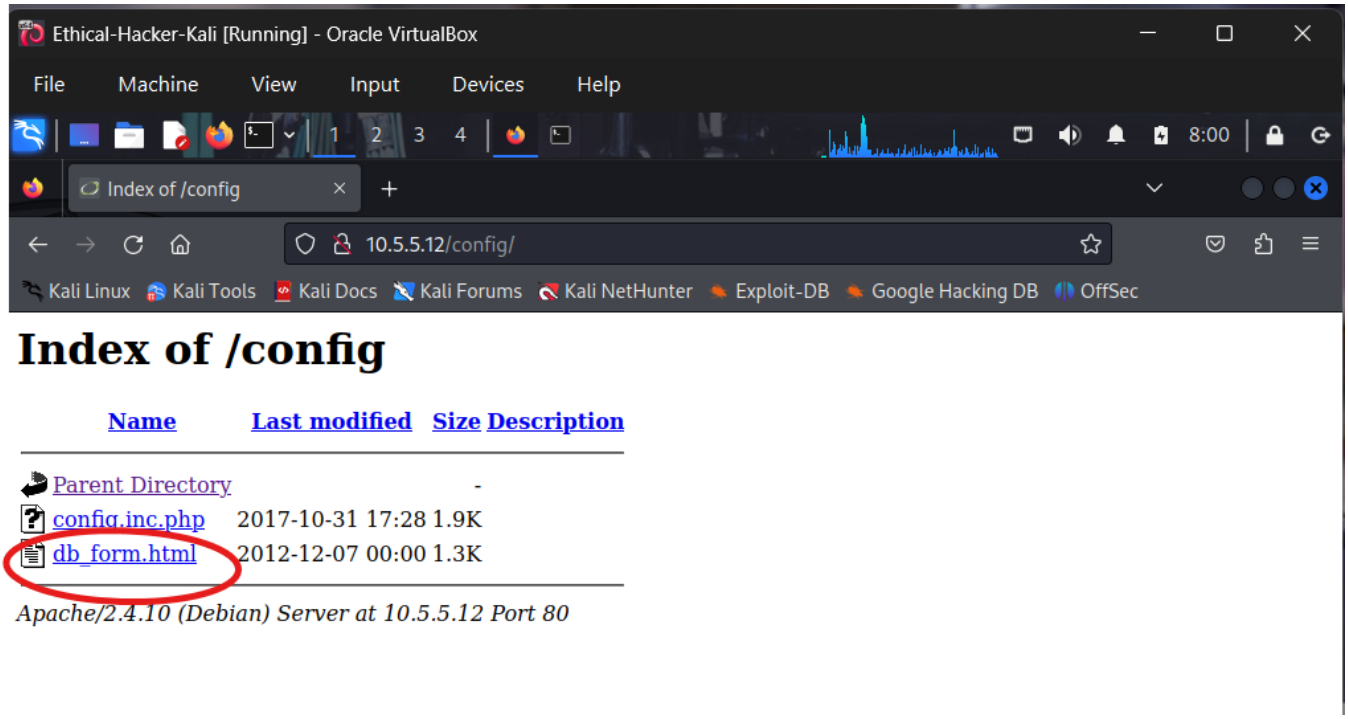


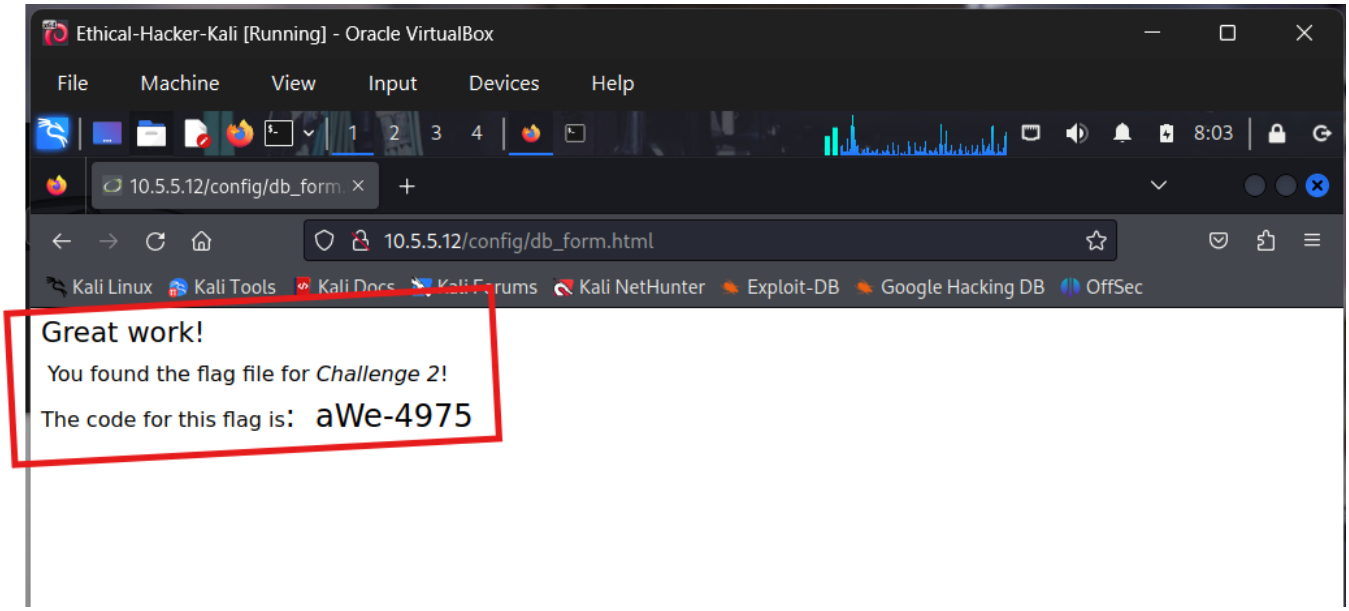
Step 3: View the files contained in each directory to find the db_form.html file.

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.

In which two subdirectories can you look for the file?

Answer: 10.5.5.12/config/





What is the filename with the Challenge 2 code?

Answer: db_form.html

Which subdirectory held the file?

Answer: /config/

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

Answer: 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?

Answer: To prevent directory listing exploits, ensure that the web server does not automatically display the contents of a folder when a default file (like index.php or index.html) is missing.

- Disable Directory Indexing in Web Server Configuration: The most effective method is to modify the web server's configuration file to explicitly forbid directory browsing.
- Use Placeholder Index Files: If you cannot modify the server configuration, you can "mask" the directory by placing an empty or generic index file in every subdirectory.

Challenge 3: Exploit open SMB Server Shares

Total points: 25

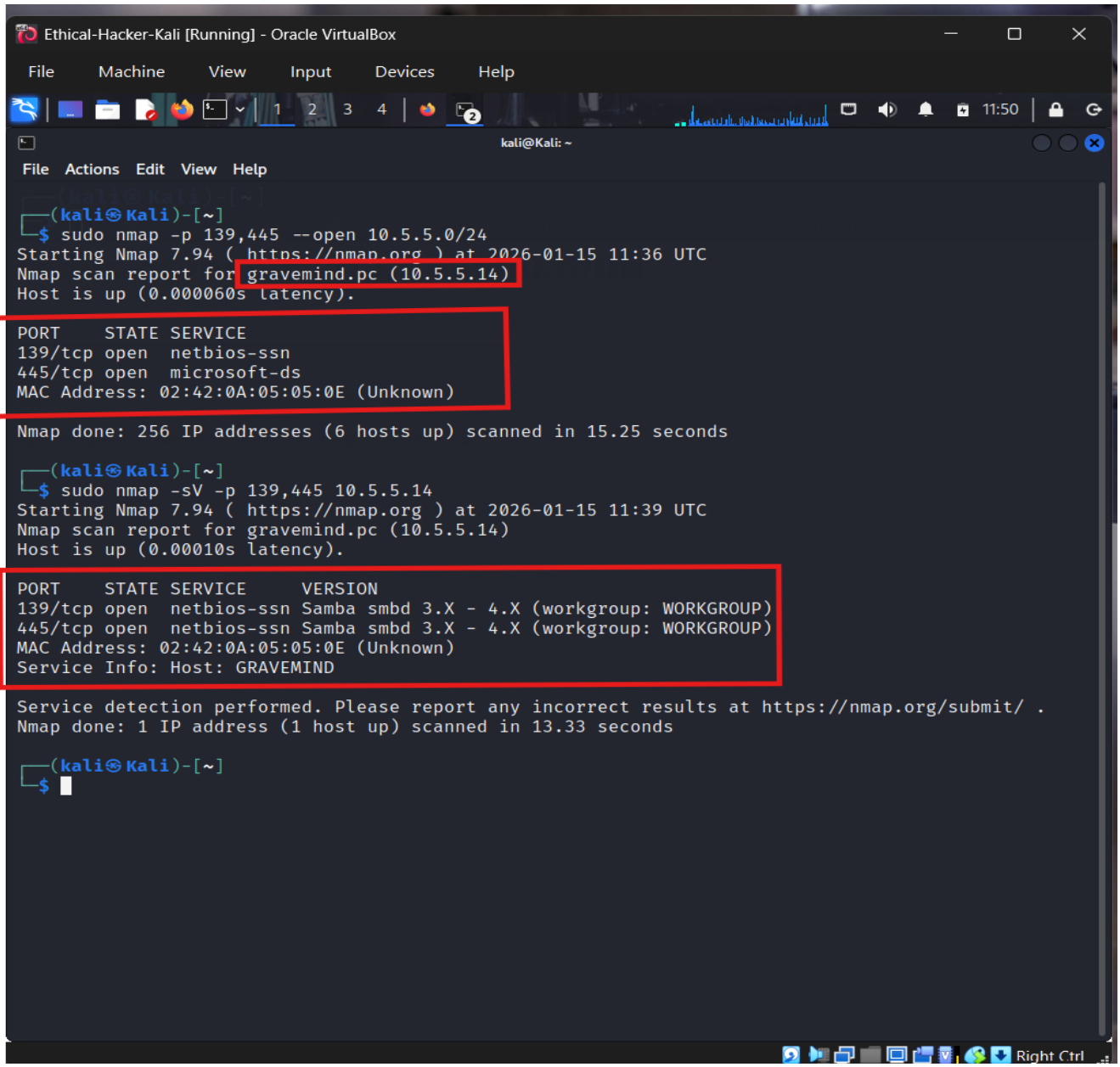
In this part, you want to discover if there are any unsecured shared directories located on an SMB server in the 10.5.5.0/24 network. You can use any of the tools you learned in earlier labs to find the drive shares available on the servers.

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.

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

Answer: gravemind.pc (10.5.5.14)



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.

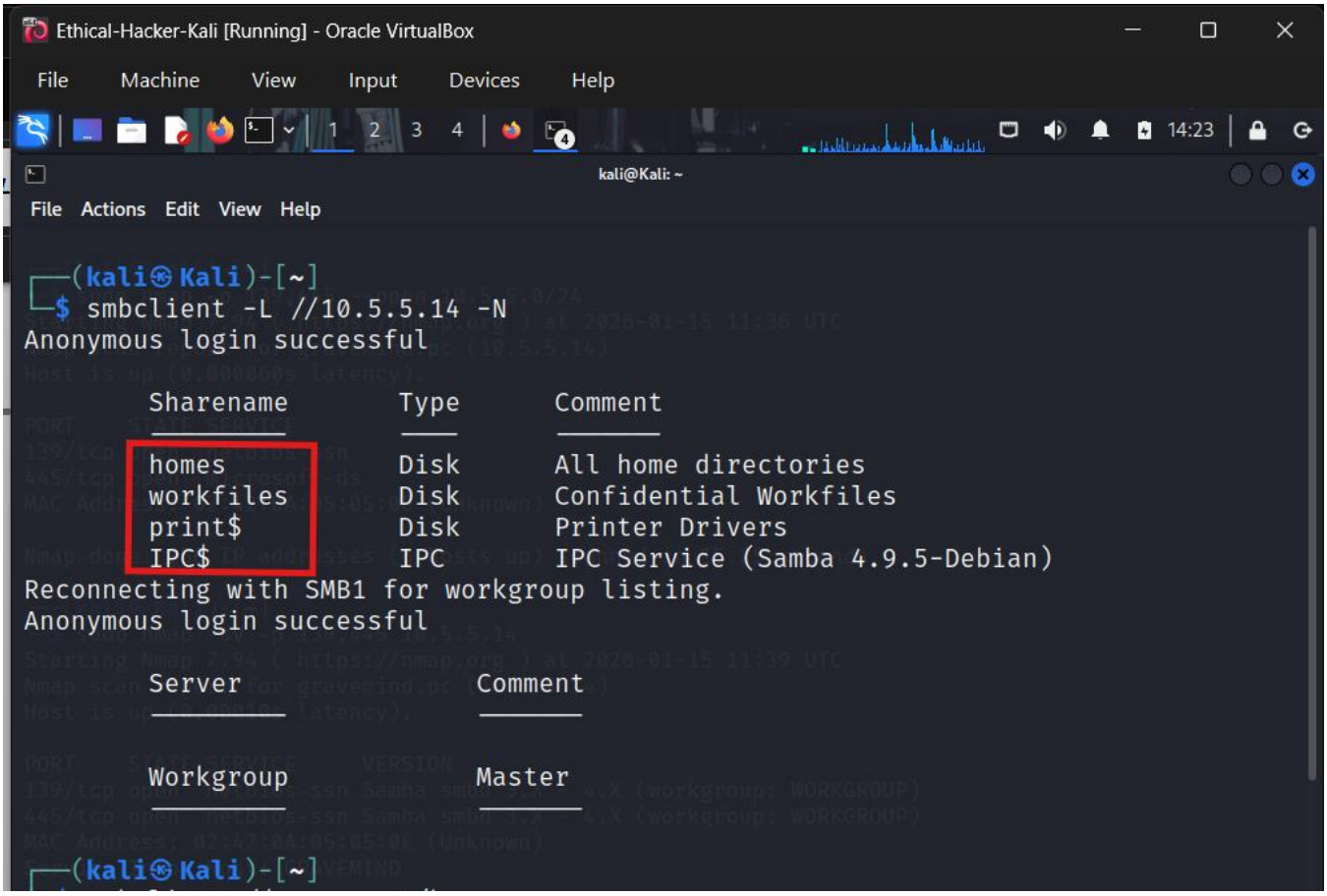
What shares are listed on the SMB server? Which ones are accessible without a valid user login?

Answer:

Listed shares:

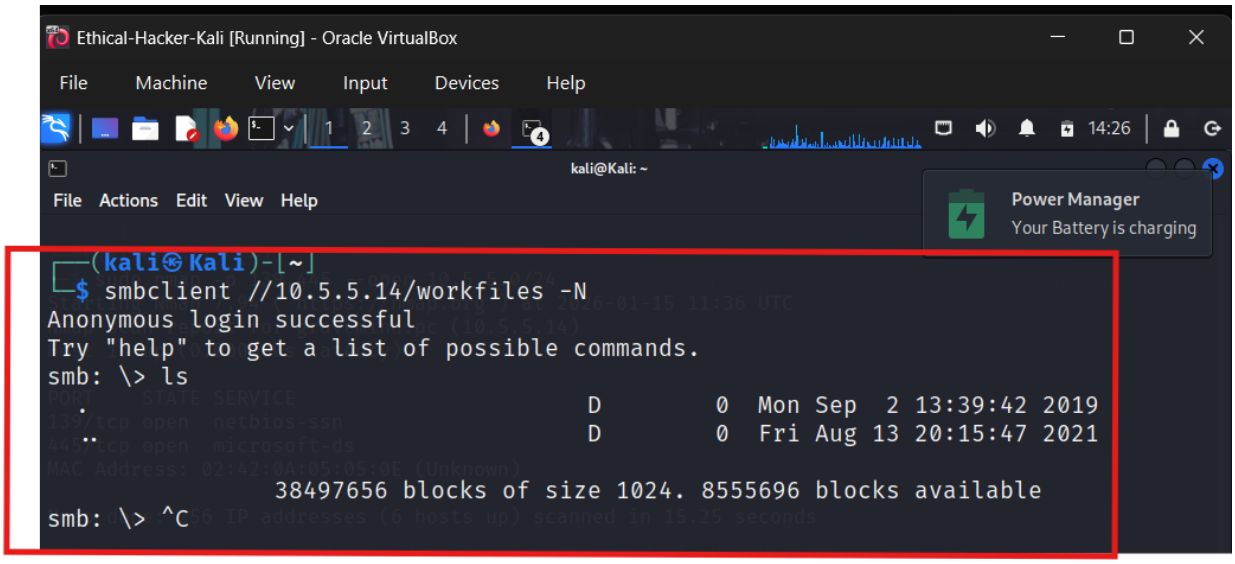
- ***homes***
- ***workfiles***

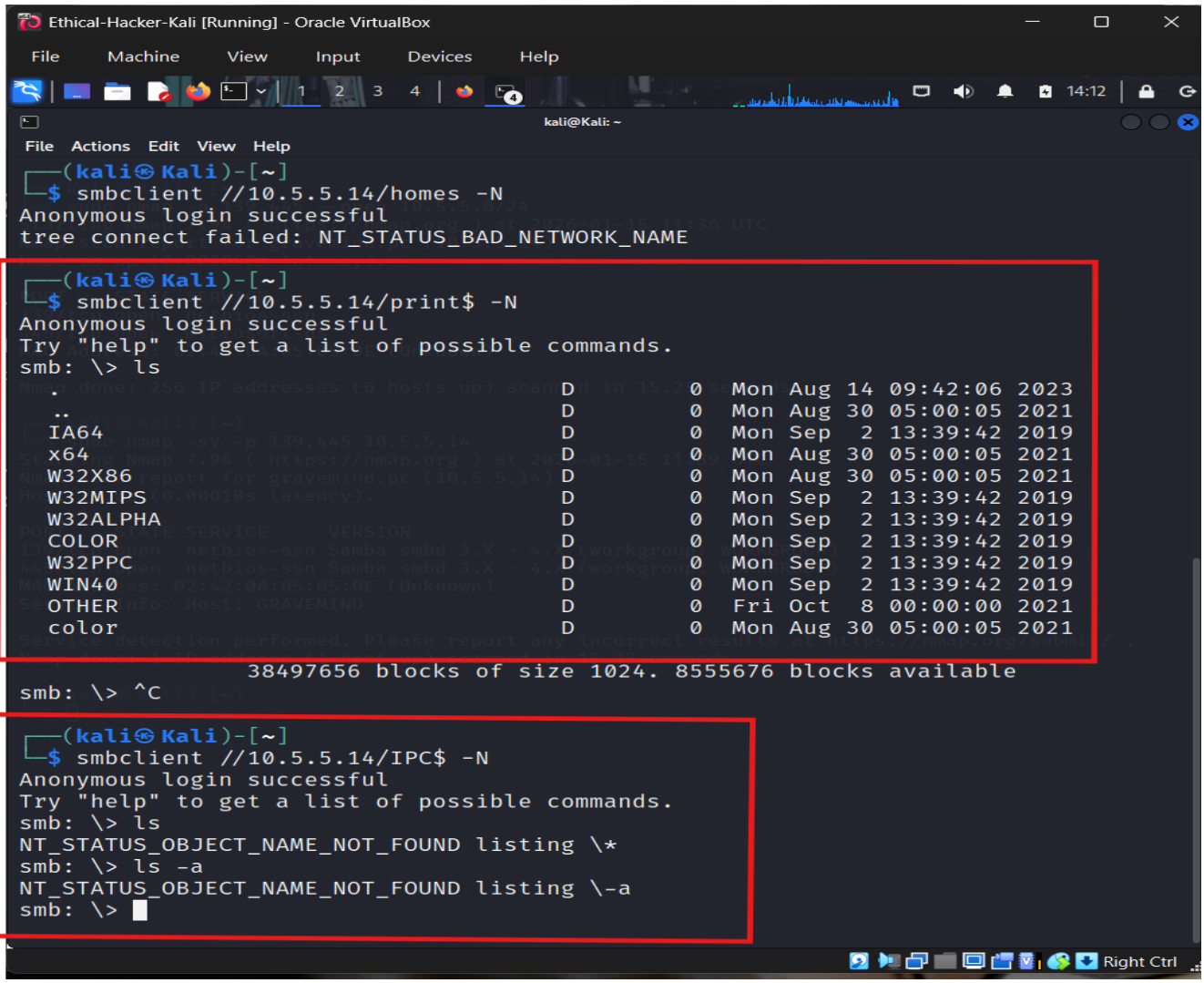
- ***print$***
- ***IPC$***



Shares accessible without a valid user login:

- ***workfiles***
- ***print$***
- ***IPC$***





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.

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

In which share is the file found?

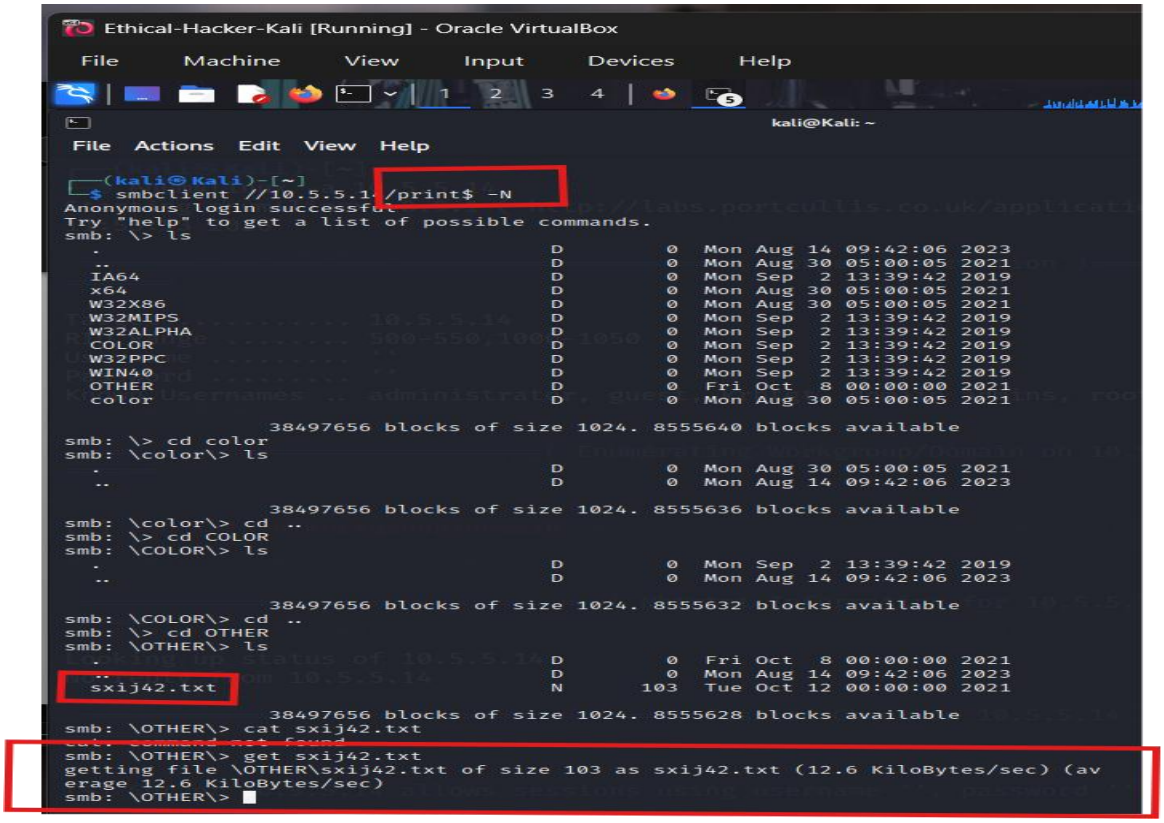
Answer: print$

What is the name of the file with Challenge 3 code?

Answer: sxij42.txt

Enter the code for Challenge 3 below.

Answer: NWs39691



Step 4: Research and propose SMB attack remediation.

What are two remediation methods for preventing SMB servers from being accessed?

ANSWER: *To prevent unauthorized access to SMB servers, such as anonymous enumeration, the following methods of remediation are recommended:*

1. Disable Null Sessions and Anonymous Access: The most effective way to prevent this reconnaissance is to configure the server to require authentication for all share and IPC access.

- *Implementation (Windows): Modify the Windows Registry or Group Policy to set RestrictAnonymous to 1 or 2. This prevents anonymous users from listing share names, account names, and SID information.*
- *Implementation (Linux/Samba): In the smb.conf file, ensure the map to guest parameter is set to Never and that guest ok = no is defined for all specific shares.*

2. Enforce SMB Signing and Encryption: Enforcing security signatures prevents "Man-in-the-Middle" (MITM) attacks where an attacker intercepts or modifies SMB traffic between the client and server.

- *SMB Signing: Ensures that every packet contains a signature to verify its origin and integrity. If the signature doesn't match, the packet is discarded.*
- *SMB Encryption: Starting with SMB 3.0, you can enforce end-to-end encryption. This prevents attackers from using packet sniffers (like Wireshark) to view the contents of files being transferred across the network.*

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

Total Points: **25**

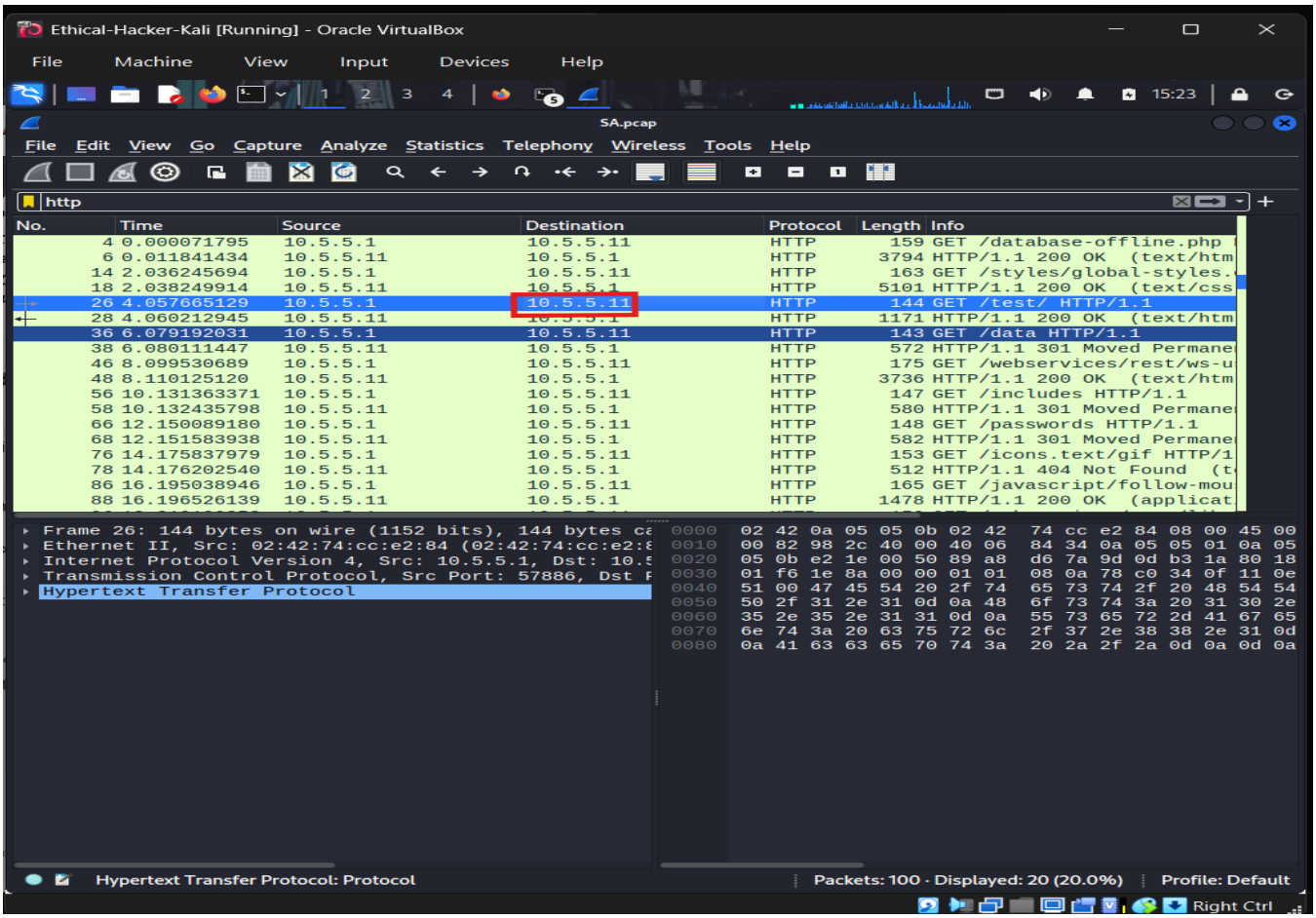
As part of your reconnaissance effort, your team captured traffic using Wireshark. The capture file, **SA.pcap**, is located in the **Downloads** subdirectory within the **kali** user home directory.

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?

Answer: 10.5.5.11



What directories on the target are revealed in the PCAP?

Answer: The PCAP snippet provided reveals several directories on the target server 10.5.5.11 that were accessed or attempted to be accessed by the client 10.5.5.1.

The following directories were accessed via HTTP GET requests:

- /styles/: Accessed via a GET request for /styles/global-styles.css.
- /test/: Accessed via a GET request to /test/.
- /data: Accessed via a GET request to /data, which resulted in a 301 Moved Permanently response.
- /webservices/rest/: Accessed via a GET request for /webservices/rest/ws-user-account.php.
- /includes/: Accessed via a GET request to /includes/, which resulted in a 301 Moved Permanently response.
- /passwords: Accessed via a GET request to /passwords, which resulted in a 301 Moved Permanently response.
- /icons.text/: Accessed via a GET request for /icons.text/gif, which resulted in a 404 Not Found response.
- /javascript/: Accessed via a GET request for /javascript/follow-mouse.js.
- /webservices/soap/lib: Accessed via a GET request to /webservices/soap/lib, which resulted in a 301 Moved Permanently response.

data

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

Use a web browser to investigate the URLs listed in the Wireshark output. Find the file with the code for Challenge 4.

What is the URL of the file?

Answer: http://10.5.5.11/data/user_accounts.xml

What is the content of the file?

Answer:

<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>jeremy</UserName>

<Password>password</Password>

<Signature>d1373 1337 speak</Signature>

<Type>Admin</Type>
</Employee>
<Employee ID="5">
<UserName>bryce</UserName>
<Password>password</Password>
<Signature>I Love SANS</Signature>
<Type>Admin</Type>
</Employee>
<Employee ID="6">
<UserName>samurai</UserName>
<Password>samurai</Password>
<Signature>Carving fools</Signature>
<Type>Admin</Type>
</Employee>
<Employee ID="7">
<UserName>jim</UserName>
<Password>password</Password>
<Signature>Rome is burning</Signature>
<Type>Admin</Type>
</Employee>
<Employee ID="8">
<UserName>bobby</UserName>
<Password>password</Password>
<Signature>Hank is my dad</Signature>
<Type>Admin</Type>
</Employee>
<Employee ID="9">
<UserName>simba</UserName>
<Password>password</Password>
<Signature>I am a super-cat</Signature>
<Type>Admin</Type>
</Employee>
<Employee ID="10">
<UserName>dreveil</UserName>
<Password>password</Password>
<Signature>Preparation H</Signature>
<Type>Admin</Type>

```
</Employee>
<Employee ID="11">
<UserName>scotty</UserName>
<Password>password</Password>
<Signature>Scotty do</Signature>
<Type>Admin</Type>
</Employee>
<Employee ID="12">
<UserName>cal</UserName>
<Password>password</Password>
<Signature>C-A-T-S Cats Cats Cats</Signature>
<Type>Admin</Type>
</Employee>
<Employee ID="13">
<UserName>john</UserName>
<Password>password</Password>
<Signature>Do the Duggie!</Signature>
<Type>Admin</Type>
</Employee>
<Employee ID="14">
<UserName>kevin</UserName>
<Password>42</Password>
<Signature>Doug Adams rocks</Signature>
<Type>Admin</Type>
</Employee>
<Employee ID="15">
<UserName>dave</UserName>
<Password>set</Password>
<Signature>Bet on S.E.T. FTW</Signature>
<Type>Admin</Type>
</Employee>
<Employee ID="16">
<UserName>patches</UserName>
<Password>tortoise</Password>
<Signature>meow</Signature>
<Type>Admin</Type>
</Employee>
```

```
<Employee ID="17">
<UserName>rocky</UserName>
<Password>stripes</Password>
<Signature>treats?</Signature>
<Type>Admin</Type>
</Employee>
<Employee ID="18">
<UserName>tim</UserName>
<Password>lanmaster53</Password>
<Signature>Because reconnaissance is hard to spell</Signature>
<Type>Admin</Type>
</Employee>
<Employee ID="19">
<UserName>ABaker</UserName>
<Password>SoSecret</Password>
<Signature>Muffin tops only</Signature>
<Type>Admin</Type>
</Employee>
<Employee ID="20">
<UserName>PPan</UserName>
<Password>NotTelling</Password>
<Signature>Where is Tinker?</Signature>
<Type>Admin</Type>
</Employee>
<Employee ID="21">
<UserName>CHook</UserName>
<Password>JollyRoger</Password>
<Signature>Gator-hater</Signature>
<Type>Admin</Type>
</Employee>
<Employee ID="22">
<UserName>james</UserName>
<Password>i<3devs</Password>
<Signature>Occupation: Researcher</Signature>
<Type>Admin</Type>
</Employee>
<Employee ID="23">
```

<UserName>ed</UserName>

<Password>pentest</Password>

<Signature>Commandline KungFu anyone?</Signature>

<Type>Admin</Type>

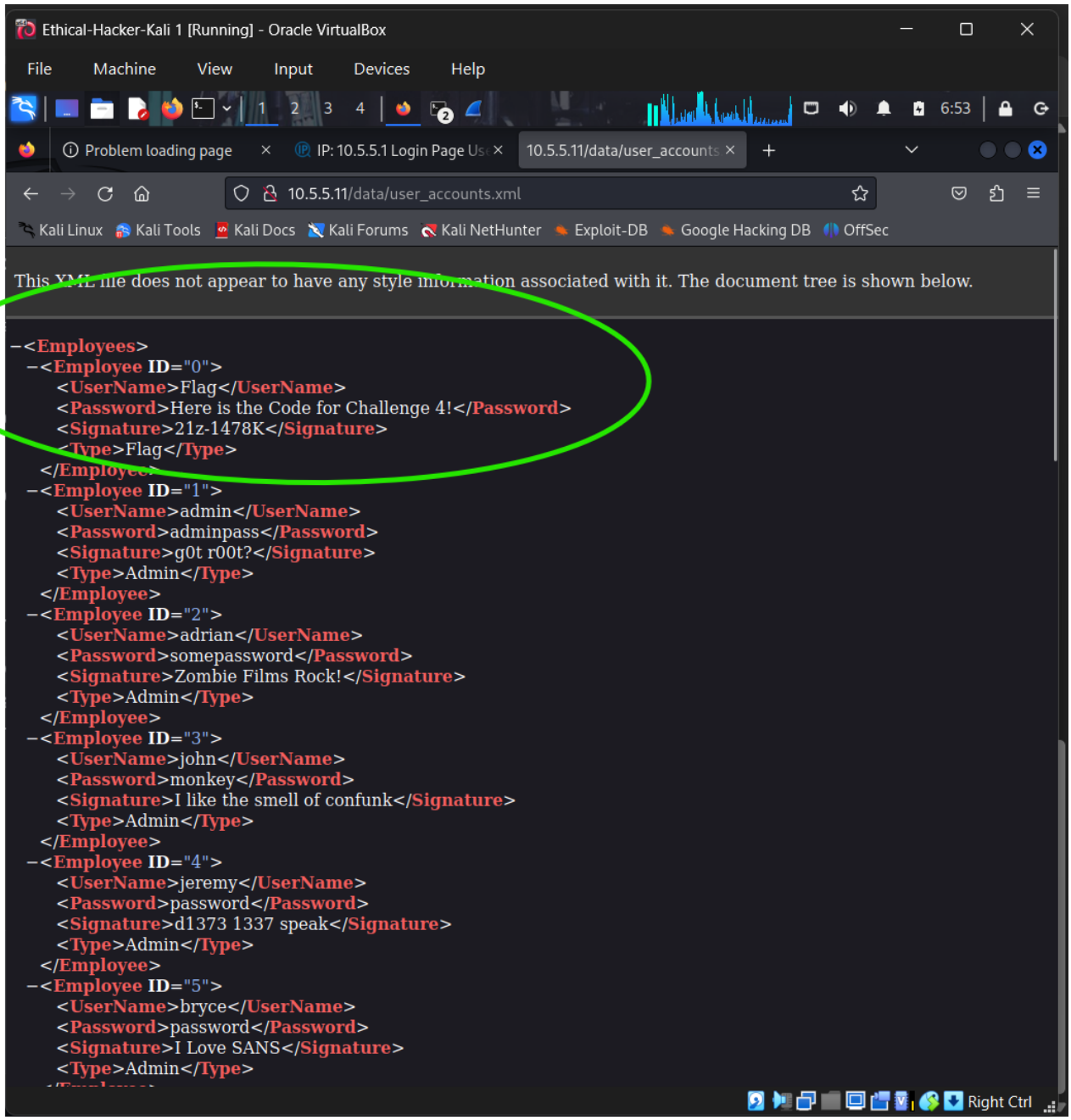
</Employee>

</Employees>

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

Answer: Here is the Code for Challenge 4!

Code: 21z-1478K



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

Further examine the capture file. The contents of the files are transmitted in clear text and can be viewed in Wireshark.

What are two remediation methods that can prevent unauthorized persons from viewing the content of the files?

Answer: The two primary remediation methods that can prevent sensitive file content from being transmitted in clear text and viewed by unauthorized persons, using a tool like Wireshark are:

1. Implement Transport Layer Security (TLS/SSL)

The most effective method is to enforce **encryption** for all network traffic. This is achieved using protocols like Transport Layer Security (TLS), the modern successor to SSL.

2. Use Secure File Transfer Protocols

For file transfers, the insecure File Transfer Protocol (FTP) should be disabled and replaced with a secure alternative.

Congratulations! You have completed the skills assessment.