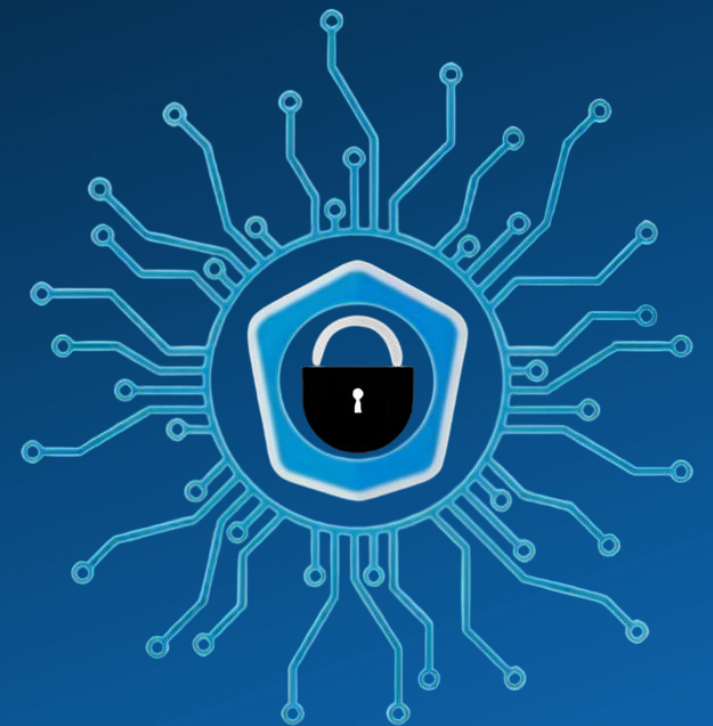


# **ANONYMOUS FILE CHALLENGE WALKTHROUGH**

**WRITTEN BY:  
TALEEN SKAFI**



**BZU - CSC**

Welcome to **AnonymousFile**, a classic challenge where an FTP server is open to the public. Connect using anonymous login, explore the server, and find the file named flag.txt to get your reward.



First, we need to run an **nmap** scan on the server to see which ports are open. We use the following command:

```
# nmap -sC -sV -p- <Server-IP>
```

From the results, we can clearly see that port 21 (FTP) is open with anonymous login allowed.

This means anyone can access the FTP server by logging in with the username **anonymous** (usually without a password), without needing valid credentials.

```
(root@kali:~/var/www/html)
# nmap -sC -sV -p- localhost
Starting Nmap 7.95 ( https://nmap.org ) 25-08-14 16:35 EDT
Nmap scan report for localhost (127.0.0.1)
Host is up (0.000037s latency).
Other addresses for localhost (not scanned): ::1
Not shown: 65529 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
21/tcp    open  ftp    vsftpd 3.0.5
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
|_ -rw-r--r-- 1 ftp ftp 27 Aug 13 20:48 flag.txt
| ftp-syst:
| STAT:
| FTP server status:
|   Connected to 127.0.0.1
|   Logged in as ftp
|   TYPE: ASCII
|   No session bandwidth limit
|   Session timeout in seconds is 300
|   Control connection is plain text
|   Data connections will be plain text
|   At session startup, client count was 1
| vsFTPD 3.0.5 - secure, fast, stable
```



We can also notice that the FTP server contains a file named **flag.txt**.

Let's connect to the server using the anonymous credentials and download the file:

```
# ftp <Server-IP>
```

When prompted for a username, enter anonymous, and press Enter (you can leave the password blank).

```
# ftp localhost
Trying [::1]:21 ...
ftp: Can't connect to `::1:21': Connection refused
Trying 127.0.0.1:21 ...
Connected to localhost.
220 (vsFTPd 3.0.5)
Name (localhost:kali): anonymous
230 Login successful.
```

Once connected, use the following commands:

```
└─# ls
```

```
└─# get flag.txt
```

Using binary mode to transfer files.

```
ftp> ls
```

229 Entering Extended Passive Mode (|||42454|)

150 Here comes the directory listing.

```
-rwxr-xr-x 1 ftp  ftp    27 Aug 13 20:48 flag.txt
```

226 Directory send OK.

```
ftp> get flag.txt
```

local: flag.txt remote: flag.txt

229 Entering Extended Passive Mode (|||45138|)

150 Opening BINARY mode data connection for flag.txt (27 bytes).

100% |\*\*\*\*\*| 27 16.62 KiB/s 00:00 ETA

226 Transfer complete.

27 bytes received in 00:00 (8.21 KiB/s)

```
ftp> |
```



Now exit the FTP server using **exit** command, and simply read the **flag.txt** file using:

```
└─# cat flag.txt
```

And that's it , you've got your flag 😊

```
└─(root@kali)~# cat flag.txt  
CSC-BZU{th1s_1$_Th3_FTP_Fl@g}
```