MonitorsTwo			
Organization: Hack The Box		Type: online CTF	
Categories:	☐ Network Security ☐ Cryptography ☐ Mobile Applications	☐ Reverse Engineering ☑ Web Applications ☐ Forensics	Difficulty: Easy
Name: Kasper Verhulst		Release date: 29 Apr 2023 Completing date:16 May 2023	

Scanning & Reconaissance

First, let us start scanning the machine to see which services are running. As usual, let's start by running an nmap command.

```
sudo nmap -A -sS -p1-1024 $BOX_IP -oN nmap.out
```

We find the following services running on the machine:

Port	Service	Version
22/tcp open	SSH	OpenSSH 8.2p1
80/tcp open	HTTP	nginx 1.18.0

We don't find any critical vulnerabilities present in these services.

Let us try to visit the website http://$BOX_IP. We see the box is running an instance of Cacti v1.2.22 and a quick Google search immediately warns us for potential RCE vulnerabilities for this specific version of Cacti.

Gaining Access

Let 's download a PoC that exploits the vulnerability. Open a reverse shell on our attacker's machine:

```
nc -nlvp 9393
```

And configure the PoC to connect to our IP address and the port opened. Once we run the exploit script, we have a shell on our Cacti web server. As soon as we have access to the web server, we can start with some manual exploration. Weirdly enough, the $sudo - l$ command is not available. The Cacti server is running as the user www-data. There are no interesting environment variables set and we don't find anything in the user's home directory. I also inspected the application files but didn't immediately find anything that could help me. Next, I checked which users are existing on the box by opening the `/etc/passwd` file. Surprisingly, the root user is the only user that has a shell, so I didn't immediately know which user would hold the user flag. Eventually in the `/` directory I found a `entrypoint.sh` and `dockerenv` which indicates we are not yet on the server level but inside a docker container. This also explains why there was no user account available.

In the Docker entrypoint script, the cacti server is connected with another Docker container that is running a MySQL database. The Docker entrypoint script also reveals the credentials to connect with this database. Let us see which tables exist on the database:

```
mysql --host=db --user=root --password=root cacti -e "show tables"
```

The the *user_auth* table might be interesting because it probably contains credentials:

```
mysql --host=db --user=root --password=root cacti -e "SELECT * FROM user_auth"
```

which reveals us the hashed password for users *marcus* and *admin*. A hash analyser shows the hashes are in the bcrypt format. Let us try to crack these hashes using John the Ripper:

```
john --format=bcrypt --wordlist /usr/share/wordlists/rockyou.txt marcus_hash.txt
john --format=bcrypt --wordlist=/usr/share/wordlists/rockyou.txt admin_hash.txt
```

We don't manage to find the password for the admin user, but marcus's password can be easily cracked. The credentials from marcus are accepted when accessing the cactus portal, but he doesn't have any rights inside cacti. However, marcus is reusing his password for his linux account and we can access the VM:

```
ssh marcus@$BOX_IP
```

in marcus's home directory, we can find the user flag.

Privilege Escalation

Once we have access to the machine as user marcus, we start to investigate how we can furhter elevate our privileges. The `/etc/passwd` file shows there is no other user except marcus and root that have a shell attached. We find no blatanly obvious ways to elevate our privileges so time to run some scripts that help us with this. First, let us transfer the scripts to the box:

```
scp linpeas.sh marcus@$BOX_IP$:/tmp/
scp LinEnum.sh marcus@$BOX_IP$:/tmp/
```

The enumeration scripts don't immediately reveal any blaring vulnerabilities, but they warn us there is a mail for marcus stored however:

> From: administrator@monitorstwo.htb
> To: all@monitorstwo.htb
> Subject: Security Bulletin - Three Vulnerabilities to be Aware Of
>
> Dear all,
>
> We would like to bring to your attention three vulnerabilities that have been recently discovered and should be addressed as soon as possible.
>
> CVE-2021-33033: This vulnerability affects the Linux kernel before 5.11.14 and is related to the CIPSO and CALIPSO refcounting for the DOI definitions. Attackers can exploit this use-after-free issue to write arbitrary values. Please update your kernel to version 5.11.14 or later to address this vulnerability.
>
> CVE-2021-41091: This vulnerability affects Moby, an open-source project created by Docker for software containerization. Attackers could exploit this vulnerability by traversing directory contents and executing programs on the data directory with insufficiently restricted permissions. The bug has been fixed in Moby (Docker Engine) version 20.10.9, and users should update to this version as soon as possible. Please note that running containers should be stopped and restarted for the permissions to be fixed.
>
> We encourage you to take the necessary steps to address these vulnerabilities promptly to avoid any potential security breaches. If you have any questions or concerns, please do not hesitate to contact our IT department.
>
> Best regards,

Administrator
CISO
Monitor Two
Security Team

The vulnerability regarding Moby/Docker seems promising since it allows for elevated privileges. Let us quickly check the version of the Docker engine running on the host:

```
docker --version
```

the Docker version 20.10.5 is indeed vulnerable to CVE-2021-41091. I found an exploit, but it requires the Docker container to run with elevated privileges.

Initially, the Docker container was running as *www-data*. Let us check if we can try if we can elevate our privileges inside the docker container, again by running the enumeration scripts. The scripts return two potential attack vectors:

- /sbin/capsh SUID bit is set
- cap_chown capabilty available to www-data user

If we check on GTFOBins, we can easily exploit the capsh SUID bit set for capsh to achieve a root shell (in the docker container) by running:

```
/sbin/capsh --gid=0 --uid=0 --
```

Now, we can run the exploit for the CVE-2021-41091 on the box itself and we have a root shell after we run $/bin/bash -p$ in the overlay volume as explained by the exploit code.