Helec

R202 - TD capabilities

2 Lecture des capabilities :

1. Quelles sont les pouvoirs des "capabilities" suivantes: CAP_NET_ADMIN,CAP_NET_RAW, CAP_NET_BIND_S CAP_SYS_ADMIN.

`CAP_NET_ADMIN` : Permet de modifier les paramètres réseaux, de créer ou supprimer des interfaces réseaux, de modifier les tables de routage, de modifier les règles de pare-feu, de capturer ou injecter des paquets, etc.

`CAP_NET_RAW` : Permet de créer des sockets de type RAW, de modifier les adresses MAC, de modifier les adresses IP, de modifier les ports, etc.

`CAP_NET_BIND_S` : Permet de créer des sockets de type STREAM, de modifier les adresses MAC, de modifier les adresses IP, de modifier les ports, etc.

`CAP_SYS_ADMIN` : Permet de modifier les paramètres du système, de monter ou démonter des systèmes de fichiers, de modifier les quotas, de modifier les priorités, de modif

2. Retrouvez les capabilities de votre kernel à l'aide la commande `firejail --debug-caps`.

`firejail --debug-caps` : En laçant cette commande j'ai obtenu :

```
0 -Chown
1 -Dac_override
2 -Dac_read_search
3 -Fowner
4 -Fsetid
5 -Kill
6 -Setgid
7 -Setuid
8 -Setpcap
9 -Linux_immutable
10 -Net_bind_service
11 -Net_broadcast
12 -Net_admin
13 -Net_raw
14 -IPC_lock
15 -IPC_owner
16 -Sys_module
17 -Sys_rawio
18 -Sys_chroot
19 -Sys_ptrace
20 -Sys_pacct
21 -Sys_admin
22 -Sys_boot
```

```
23 -Sys_nice
24 -Sys_resource
25 -Sys_time
26 -Sys_tty_config
27 -Mknod
28 -Lease
29 -Audit_write
30 -Audit_control
31 -Setfcap
32 -Mac_override
33 -Mac_admin
34 -Syslog
35 -Wake_alarm
36 -Block_suspend
37 -Audit_read
```

3. Listez tous les programmes de votre machine avec "capabilities" avec la commande suivante:

`getcap -r / 2>/dev/null`

4. A l'aide de la commande getcaps retrouvez les capabilities attachées au programme ping.

`getcap /bin/ping` : En lançant cette commande j'ai obtenu :

```
/bin/ping = cap_net_raw+ep
```

5. Sous l'utilisateur "test" visualisez les capabilities de votre processus bash dans /proc à l'aide de la commande suivante:

`cat /proc/$$/status | egrep "^Cap"` :

```
```
CapInh: 0000000000000000
CapPrm: 0000000000000000
CapEff: 0000000000000000
CapBnd: 0000001fffffffff
CapAmb: 0000000000000000
```
```

6. A quoi correspondent CapInh,CapPrm,CapEff ?

`CapInh` : Capabilities héritées du parent.

`CapPrm` : Capabilities permises.

`CapEff` : Capabilities effectives.

7. Que donne la commande précédente avec un processus bash sous root ?

`sudo cat /proc/$$/status | egrep "^Cap"` :

```
```
CapInh: 0000000000000000
CapPrm: 0000001fffffffff
CapEff: 0000001fffffffff
CapBnd: 0000001fffffffff
CapAmb: 0000000000000000
```
```

8. Utilisez capsh -decode=valeur pour décoder les "effective capabilities" ?

`capsh --decode=0000001fffffffff` : En lançant cette commande j'ai obtenu :

```
0x0000001fffffffff=cap_chown,cap_dac_override,cap_dac_read_search,cap_fowne
r,cap_fsetid,cap_kill,cap_setgid,cap_setuid,cap_setpcap,cap_linux_immutable
,cap_net_bind_service,cap_net_broadcast,cap_net_admin,cap_net_raw,cap_ipc_l
ock,cap_ipc_owner,cap_sys_module,cap_sys_rawio,cap_sys_chroot,cap_sys_ptrac
e,cap_sys_pacct,cap_sys_admin,cap_sys_boot,cap_sys_nice,cap_sys_resource,ca
p_sys_time,cap_sys_tty_config,cap_mknod,cap_lease,cap_audit_write,cap_audit
_control,cap_setfcap,cap_mac_override,cap_mac_admin,cap_syslog,cap_wake_ala
rm,cap_block_suspend,cap_audit_read+i
```

9. Sous un utilisateur non privilégié refaites les opérations précédentes. Rassuré ?

`capsh --decode=0000000000000000` : En lançant cette commande j'ai obtenu :

```
0x0000000000000000=
```

3 Modifications des capabilities :

1. Lancez la commande `python3 -m http.server port` avec comme port 9000 puis 80. Que se passe-t-il avec le port 80 ?

`python3 -m http.server 9000` : En lançant cette commande j'ai obtenu :

```
Serving HTTP on
```

`python3 -m http.server 80` : En lançant cette commande j'ai obtenu :

```
Serving HTTP on
error SSL
```

2. Donnez la "capability" permettant de se binder sur le port 80 avec setcap.

`setcap cap_net_bind_service=+ep /usr/bin/python3.8` : En lançant cette commande j'ai obtenu :

```
/usr/bin/python3.8 = cap_net_bind_service+ep
```

3. Vérifiez avec getcap que la capability a bien été acquise.

`getcap /usr/bin/python3.8` : En lançant cette commande j'ai obtenu :

```
/usr/bin/python3.8 = cap_net_bind_service+ep
```

4. sous root enlevez toutes les capabilities à votre processus bash et verifiez que vous ne pouvez plus rien faire (ping, tcpdump...). capsh --drop=all --secbits=1 --

`capsh --drop=all --secbits=1 --`: En lançant cette commande puis `ping 0.0.0.0` j'ai obtenu :

```
ping: socket: Operation not permitted
```