Linux (network e mount) namespaces

boyska

4 settembre 2021

namespace de che?

In genere siamo abituati che tutti i programmi sul computer condividano lo stesso “stato”: se metto un file in una directory, tutti vedranno quel file (al netto dell’avere i permessi).

I namespace cambiano questa cosa.

privileged?

ci sono pure gli unprivileged namespace, ma non ne so nulla.

sì, il nome è una merda, non è colpa mia.

Che ci faccio

Molta gente ci fa i container, ci si può fare anche altro.

No ai nomi securitari (`firejail`), `apt install bubblewrap`

Mount namespaces

Ogni processo ha una certa visione del filesystem.

```
# mount|grep ' / '
/dev/mapper/tegaminovg-root on / type ext4
(rw,relatime,errors=remount-ro)
# bwrap --ro-bind / / mount|grep ' / '
/dev/mapper/tegaminovg-root on / type ext4
(ro,nosuid,nodev,relatime,errors=remount-ro)
```

Occhio: da **rw** siamo passati a **ro**

```
# bwrap --ro-bind / / touch /root/foo
touch: cannot touch '/root/foo': Read-only fi
```

```
% sudo bwrap --bind / / --tmpfs /etc/ touch /e
% ls /etc/foo
ls: cannot access '/etc/foo': No such file or
```

Con trucchi di questo genere si può:

- “nascondere” alcuni file ad un processo
- far vedere file ad un procsso che altrimenti non vedrebbe

Esempio

Vogliamo fare uno script che un utente normale può lanciare, e modifica un file in una directory "privilegiata". Come facciamo?

Potremmo fare uno script che gira come root e fa tutto. Ma **NON funziona su wayland!** (e non è buona pratica in generale)

```
#!/bin/sh
[ $(id -u) -eq 0 ] || exec sudo -n -- "$@"
touch /segreti/antani || zenity --error
```

```
#!/bin/sh
# wrapper
unshare --bind / /
 --bind /dir-inaccessibile/segreti /segreti/
 mioscript.sh

#!/bin/sh
# mioscript.sh
touch /segreti/antani || zenity --error
```

Network

Ogni processo ha accesso allo stack di rete che il sistema operativo ha.

Se in un namespace, ha accesso ad un *altro* stack di rete.

- no interfacce di rete (tranne loopback)
- e il loopback non è lo stesso dell'host!
- ha `iptables`, ma ha un elenco di regole separato rispetto all'host
- detta così, sembra che semplicemente non abbia alcun accesso alla rete. ma...

```
# unshare --net ip a
1: lo: <LOOPBACK> mtu 65536 qdisc noop state D
    link/loopback 00:00:00:00:00:00 brd 00:00
```

puoi aggiungere un'interfaccia di rete virtuale ad un namespace:

```
ip link add miaeth type veth \
    peer name mionamespace
```

quell'interfaccia ha un lato sul namespace e un lato sull'host.

dai gli indirizzi come vuoi, e ovviamente puoi farci regole iptables come vuoi.

puoi rendere un servizio accessibile *solo* da uno specifico netns

```
iptables -A INPUT -p tcp -i miaeth \
    --dport 1234 -j ACCEPT
iptables -A INPUT -p tcp --dport 1234 \
    -j DROP
```

puoi forzare tutto il traffico che viene da un certo netns a passare attraverso un proxy: `orjail` fa questo per forzare l'uso di Tor.

Puoi loggare tutto il traffico di rete che viene da uno specifico netns.