Useful Uses of chroot/proot

An introduction to chroot, proot, and the internals of containerization

Simon Elmir
Artificial Intelligence
Production Engineering

Meta
About me

Production Engineering
Reliability, scalability, efficiency

Artificial Intelligence
AI powers many products at Meta

(Not Containers)
Production Engineers tend to have a breadth of knowledge across systems
This talk

What chroot does
chroot - CHange ROOT

Rescue a Linux install
Change a password, install a package

Debian on Android
Use Linux apps on your (rooted) phone

Faking things with PRoot
Without needing to be root

Faking a CPU architecture
Running ARM programs on an x86_64 CPU
Isolation

• Untrusted applications/users
• Control resource usage
• Foreign systems

Containers

• Lighter than virtual machines
• Docker, LXC, etc.
• “It works on my machine”
Using chroot
Isolate the directory a program has access to

Change the root directory

```bash
mkdir -p tmp/bin
cp busybox tmp/bin
sudo chroot tmp /bin/busybox sh
```
Chroot: Rescue a System
Rescue a System

**Boot from a liveUSB (or liveCD)**
Or attach the target’s disk to a working system

**Mount the target’s filesystem(s)**
sudo mkdir /mnt/target
sudo mount /dev/sda2 /mnt/target
# similar for /mnt/target/boot,/mnt/target/boot/efi

**Supplemental filesystems**
sudo mount --bind /dev /mnt/target/dev
# similar for /proc,/sys

**Chroot**
sudo chroot /mnt/target /bin/bash

**Additional configs**
# caution: resolv.conf may be a symlink.
# mv /etc/resolv.conf /etc/resolv.conf.bak
echo nameserver 8.8.8.8 > /etc/resolv.conf
Chroot on (rooted) Android
Chroot on
(rooted)
Android

Rootfs

Extract
su
cd /data/media
mkdir linux
chattr -F linux
tar -xvzf 0/Download/ubuntu-base-22.04-base-arm64.tar.gz -C linux

Supplemental filesystems
mount --bind /dev /data/media/linux/dev
# similar for /proc, /sys, /dev/pts

Chroot
chroot /data/media/linux /bin/login -f root

Additional configs
echo nameserver 8.8.8.8 > /etc/resolv.conf
groupadd -g 3003 android_inet
groupadd -g 3004 android_inet_raw
usermod _apt -g android_inet
PRoot
**Userspace (non-root)**

```bash
croot, mount --bind, binfmt_misc
```

**Substitute a file (or directory)**

```bash
proot -b redhat-release:/etc/redhat-release lsb_release -a
```

**Chroot**

```bash
proot -R /bin/target
```

**ARM vs x86: Raspberry Pi**

```bash
# binfmt_misc
proot -R rootfs/ -q /usr/bin/qemu-aarch64-static
#proot -R rootfs/ -q /usr/bin/qemu-arm-static
```

**As Root**

```bash
sudo proot -S rootfs/ -q /usr/bin/qemu-aarch64-static login -f root
```
Shared Resources
Namespaces
Namespaces

What processes can see
mount, network, pid, user, ...

unshare
sudo unshare --mount.proc --pid --fork -R ubuntu-base-22.04
mount -t devtmpfs devtmpfs /dev
mount -t sysfs sysfs /sys
Limits on shared resources
CPU, memory, IO, PID, ...

/sys/fs/cgroup
mount -t cgroup2 cgroup2 /cgroup2
mkdir, write to files, chown, ...

Demo: limit CPU
stress-ng -c0 &
cd /sys/fs/cgroup
sudo mkdir demo
cd demo
sudo tee -a cgroup.procs <<< "PID"
sudo tee -a cpu.max <<< "600000 100000"
Questions?

THANK YOU FOR YOUR TIME
Further Resources
This slide deck:

https://www.socallinuxexpo.org/scale/19x/presentations/useful-uses-chrootproot

chroot

man 1 chroot (CLI), man 2 chroot (syscall)

https://www.gnu.org/software/coreutils/chroot

PRoot

https://proot-me.github.io/

Namespaces

man 1 unshare

man 7 namespaces (cgroup, ipc, mount, network, pid, time, user, uts)

cgroups


https://facebookmicrosites.github.io/cgroup2/
PRoot on Android

Termux
https://wiki.termux.com/wiki/PRoot

Apps
(I haven’t personally tested these)
UserLAnd https://userland.tech
Andronix https://andronix.app
Or just search for: android linux without root