

CHROOT

and the Raiders of the Lost Ark

Daniel 'paradigm' Thau

The Open Source Club
The Ohio State University

Virtualization Meeting

Outline

What is Chroot?

History

Example usage

Setting it up

Security

Schroot

What is Chroot?

Root Directory

- ▶ **Windows:**

- ▶ `C:\Program Files\Mozilla\Firefox`

- ▶ ***nix:**

- ▶ `/usr/lib/iceweasel/iceweasel`

What is Chroot?

Root Directory

- ▶ **Windows:**

- ▶ `C:\Program Files\Mozilla\Firefox`
- ▶ `C:\` is the root directory of the filesystem

- ▶ ***nix:**

- ▶ `/usr/lib/iceweasel/iceweasel`
- ▶ The first `/` is the root directory of the filesystem

What is Chroot?

Changing the root folder

- ▶ What if we changed the defined location of the root directory to another directory?

What is Chroot?

Changing the root folder

- ▶ What if we changed the defined location of the root directory to another directory?
- ▶ `/chroot/paradigm/usr/lib/iceweasel/iceweasel`

What is Chroot?

Changing the root folder

- ▶ What if we changed the defined location of the root directory to another directory?
- ▶ `/chroot/paradigm/usr/lib/iceweasel/iceweasel`
 - ▶ `as root: chroot /chroot/paradigm/ COMMAND`

What is Chroot?

Changing the root folder

- ▶ What if we changed the defined location of the root directory to another directory?
- ▶ `/chroot/paradigm/usr/lib/iceweasel/iceweasel`
 - ▶ **as root:** `chroot /chroot/paradigm/ COMMAND`
 - ▶ `/usr/lib/iceweasel/iceweasel/`

What is Chroot?

Changing the root folder

- ▶ What if we changed the defined location of the root directory to another directory?
- ▶ `/chroot/paradigm/usr/lib/iceweasel/iceweasel`
 - ▶ `as root: chroot /chroot/paradigm/ COMMAND`
 - ▶ `/usr/lib/iceweasel/iceweasel/`
- ▶ That, essentially, is all chroot does.

What is Chroot?

Changing the root folder

- ▶ What if we changed the defined location of the root directory to another directory?
- ▶ `/chroot/paradigm/usr/lib/iceweasel/iceweasel`
 - ▶ as root: `chroot /chroot/paradigm/ COMMAND`
 - ▶ `/usr/lib/iceweasel/iceweasel/`
- ▶ That, essentially, is all chroot does.
- ▶ While conceptually it is simple, it can be quite a powerful tool.

History

- ▶ Bill Joy (the guy who wrote vi) introduced chroot on 18 March 1982.

History

- ▶ Bill Joy (the guy who wrote vi) introduced chroot on 18 March 1982.
 - ▶ I don't know if 'introduce' means 'wrote.'

History

- ▶ Bill Joy (the guy who wrote vi) introduced chroot on 18 March 1982.
 - ▶ I don't know if 'introduce' means 'wrote.'
 - ▶ It was written to help test the installation and build system of 4.2BSD, which came out 17 months later.

History

- ▶ Bill Joy (the guy who wrote vi) introduced chroot on 18 March 1982.
 - ▶ I don't know if 'introduce' means 'wrote.'
 - ▶ It was written to help test the installation and build system of 4.2BSD, which came out 17 months later.
- ▶ And that's all I know. There isn't much available.

Example usage

Examples

- ▶ A system is unbootable, but fixing it is more complicated than editing text files.

Example usage

Examples

- ▶ A system is unbootable, but fixing it is more complicated than editing text files.
- ▶ You have another Linux system available (e.g. dual-boot) but don't want to reboot

Example usage

Examples

- ▶ A system is unbootable, but fixing it is more complicated than editing text files.
- ▶ You have another Linux system available (e.g. dual-boot) but don't want to reboot
- ▶ You want to mix-and-match programs/libraries/etc from different distros without dealing dependency problems

Example usage

Examples

- ▶ A system is unbootable, but fixing it is more complicated than editing text files.
- ▶ You have another Linux system available (e.g. dual-boot) but don't want to reboot
- ▶ You want to mix-and-match programs/libraries/etc from different distros without dealing dependency problems
 - ▶ I'll demo/explain this in the most detail here. The concepts carry over to the other situations.

Example usage

Debian

- ▶ Debian Stable is super awesome stable.

Example usage

Debian

- ▶ Debian Stable is super awesome stable.
- ▶ Cutting-edge packages go into Debian Unstable aka Sid

Example usage

Debian

- ▶ Debian Stable is super awesome stable.
- ▶ Cutting-edge packages go into Debian Unstable aka Sid
 - ▶ Many people use this as their main distro

Example usage

Debian

- ▶ Debian Stable is super awesome stable.
- ▶ Cutting-edge packages go into Debian Unstable aka Sid
 - ▶ Many people use this as their main distro
- ▶ After some time without problems, packages from Sid go to Testing.

Example usage

Debian

- ▶ Debian Stable is super awesome stable.
- ▶ Cutting-edge packages go into Debian Unstable aka Sid
 - ▶ Many people use this as their main distro
- ▶ After some time without problems, packages from Sid go to Testing.
 - ▶ Other people like to use this as their distro. It's comparable to Arch Linux.

Example usage

Debian

- ▶ Debian Stable is super awesome stable.
- ▶ Cutting-edge packages go into Debian Unstable aka Sid
 - ▶ Many people use this as their main distro
- ▶ After some time without problems, packages from Sid go to Testing.
 - ▶ Other people like to use this as their distro. It's comparable to Arch Linux.
- ▶ After even more time without problems, Testing is branched into the new Debian Stable.

Example usage

Debian

- ▶ Debian Stable is super awesome stable.
- ▶ Cutting-edge packages go into Debian Unstable aka Sid
 - ▶ Many people use this as their main distro
- ▶ After some time without problems, packages from Sid go to Testing.
 - ▶ Other people like to use this as their distro. It's comparable to Arch Linux.
- ▶ After even more time without problems, Testing is branched into the new Debian Stable.
- ▶ As a result, Debian Stable \approx Debian Old.

Example usage

Debian Up-to-date and Stable

► Goal:

Example usage

Debian Up-to-date and Stable

- ▶ Goal:
 - ▶ Always-available stable *nix system

Example usage

Debian Up-to-date and Stable

- ▶ Goal:
 - ▶ Always-available stable *nix system
 - ▶ Readily available access to cutting-edge packages

Example usage

Debian Up-to-date and Stable

- ▶ Goal:
 - ▶ Always-available stable *nix system
 - ▶ Readily available access to cutting-edge packages
 - ▶ Minimal overhead (eepc don't have much oomph)

Setting it up

Debootstrap

- ▶ Debootstrap downloads a base install of Debian(-based distro) from a repository into a folder.

Setting it up

Debootstrap

- ▶ Debootstrap downloads a base install of Debian(-based distro) from a repository into a folder.
 - ▶ Debootstrap is, essentially, just a bash script which uses wget to download files.

Setting it up

Debootstrap

- ▶ Debootstrap downloads a base install of Debian(-based distro) from a repository into a folder.
 - ▶ Debootstrap is, essentially, just a bash script which uses wget to download files.
- ▶ as root:
debootstrap distro-release-name directory repository

Setting it up

Debootstrap

- ▶ Debootstrap downloads a base install of Debian(-based distro) from a repository into a folder.
 - ▶ Debootstrap is, essentially, just a bash script which uses wget to download files.
- ▶ as root:
debootstrap distro-release-name directory repository
- ▶ e.g.:
debootstrap sid /chroot/sid/
<http://debian.mirrors.tds.net/debian/>

Setting it up

Debootstrap

- ▶ Debootstrap downloads a base install of Debian(-based distro) from a repository into a folder.
 - ▶ Debootstrap is, essentially, just a bash script which uses wget to download files.
- ▶ as root:
debootstrap distro-release-name directory repository
- ▶ e.g.:
debootstrap sid /chroot/sid/
<http://debian.mirrors.tds.net/debian/>
- ▶ e.g.:
debootstrap karmic /chroot/kamric/
<http://archive.ubuntu.com/ubuntu/>

Setting it up

mounts

- ▶ Linux has several ‘virtual filesystems’ that are needed:

Setting it up

mounts

- ▶ Linux has several ‘virtual filesystems’ that are needed:
 - ▶ proc

Setting it up

mounts

- ▶ Linux has several ‘virtual filesystems’ that are needed:
 - ▶ proc
 - ▶ dev

Setting it up

mounts

- ▶ Linux has several ‘virtual filesystems’ that are needed:
 - ▶ proc
 - ▶ dev
 - ▶ sys

Setting it up

mounts

- ▶ Linux has several ‘virtual filesystems’ that are needed:
 - ▶ proc
 - ▶ dev
 - ▶ sys
- ▶ Can mount these manually or with fstab

Setting it up

mounts

- ▶ Linux has several ‘virtual filesystems’ that are needed:
 - ▶ proc
 - ▶ dev
 - ▶ sys
- ▶ Can mount these manually or with fstab
- ▶ May also want to mount other shared partitions

Setting it up

final configurations

- ▶ By default a debootstrap'd system doesn't have much of anything.

Setting it up

final configurations

- ▶ By default a debootstrap'd system doesn't have much of anything.
 - ▶ adduser

Setting it up

final configurations

- ▶ By default a debootstrap'd system doesn't have much of anything.
 - ▶ adduser
 - ▶ apt-get/taskel

Setting it up

final configurations

- ▶ By default a debootstrap'd system doesn't have much of anything.
 - ▶ adduser
 - ▶ apt-get/taskel
- ▶ inittab

Setting it up

final configurations

- ▶ By default a debootstrap'd system doesn't have much of anything.
 - ▶ adduser
 - ▶ apt-get/taskel
- ▶ inittab
- ▶ X11

Setting it up

final configurations

- ▶ By default a debootstrap'd system doesn't have much of anything.
 - ▶ adduser
 - ▶ apt-get/taskel
- ▶ inittab
- ▶ X11
 - ▶ xhost permissions

Setting it up

final configurations

- ▶ By default a debootstrap'd system doesn't have much of anything.
 - ▶ adduser
 - ▶ apt-get/taskel
- ▶ inittab
- ▶ X11
 - ▶ xhost permissions
 - ▶ DISPLAY variable

Setting it up

final configurations

- ▶ By default a debootstrap'd system doesn't have much of anything.
 - ▶ adduser
 - ▶ apt-get/taskel
- ▶ inittab
- ▶ X11
 - ▶ xhost permissions
 - ▶ DISPLAY variable
 - ▶ multiple X servers/monitors

Security

- ▶ chroot's are **not** fully sandboxed

Security

- ▶ chroot's are **not** fully sandboxed
 - ▶ Root can leave the chroot

Security

- ▶ chroot's are **not** fully sandboxed
 - ▶ Root can leave the chroot
 - ▶ A user in a chroot can still access the same user's out-of-chroot processes

Security

- ▶ chroot's are **not** fully sandboxed
 - ▶ Root can leave the chroot
 - ▶ A user in a chroot can still access the same user's out-of-chroot processes
 - ▶ **Do not use chroot as a security tool**

Security

- ▶ chroot's are **not** fully sandboxed
 - ▶ Root can leave the chroot
 - ▶ A user in a chroot can still access the same user's out-of-chroot processes
 - ▶ **Do not use chroot as a security tool**
- ▶ There's chroot-like options that are fully sandboxed

Security

- ▶ chroot's are **not** fully sandboxed
 - ▶ Root can leave the chroot
 - ▶ A user in a chroot can still access the same user's out-of-chroot processes
 - ▶ **Do not use chroot as a security tool**
- ▶ There's chroot-like options that are fully sandboxed
 - ▶ Linux: OpenVZ

Security

- ▶ chroot's are **not** fully sandboxed
 - ▶ Root can leave the chroot
 - ▶ A user in a chroot can still access the same user's out-of-chroot processes
 - ▶ **Do not use chroot as a security tool**
- ▶ There's chroot-like options that are fully sandboxed
 - ▶ Linux: OpenVZ
 - ▶ FreeBSD: Jails

Security

- ▶ chroot's are **not** fully sandboxed
 - ▶ Root can leave the chroot
 - ▶ A user in a chroot can still access the same user's out-of-chroot processes
 - ▶ **Do not use chroot as a security tool**
- ▶ There's chroot-like options that are fully sandboxed
 - ▶ Linux: OpenVZ
 - ▶ FreeBSD: Jails
 - ▶ Solaris: Containers

Torvalds

This guy made a kernel that can use chroot:



Schroot

Schroot



Schroot

- ▶ Chroot requires root

Schroot

- ▶ Chroot requires root
- ▶ Dchroot does not

Schroot

- ▶ Chroot requires root
- ▶ Dchroot does not
- ▶ Schroot is like dchroot, but further improved.

Schroot

- ▶ Schroot doesn't require root

Schroot

- ▶ Schroot doesn't require root
- ▶ Schroot can change directory when called

Schroot

- ▶ Schroot doesn't require root
- ▶ Schroot can change directory when called
- ▶ Schroot can perserve local variables when called

Schroot

- ▶ Schroot doesn't require root
- ▶ Schroot can change directory when called
- ▶ Schroot can perserve local variables when called
- ▶ Schroot can limit which users/groups have access

Schroot

- ▶ Schroot doesn't require root
- ▶ Schroot can change directory when called
- ▶ Schroot can perserve local variables when called
- ▶ Schroot can limit which users/groups have access
- ▶ Schroot can keep track of sessions

Schroot

- ▶ Schroot doesn't require root
- ▶ Schroot can change directory when called
- ▶ Schroot can perserve local variables when called
- ▶ Schroot can limit which users/groups have access
- ▶ Schroot can keep track of sessions
- ▶ Schroot can run scripts to help set up a chroot

Sources I



Wikipedia

chroot

<http://en.wikipedia.org/wiki/chroot/>



Debian

debootstrap

<http://wiki.debian.org/Debootstrap>



Schroot

schroot(1)

`man schroot`