Linception: Containers under Linux

What you need to know to build container stack from scratch

Jay Coles
code@pocketnix.org
http://doger.io

Who am I?

- Sysadmin for Anchor
- Programmer on the side
- Crazy person

Jay Coles
code@pocketnix.org
http://doger.io

http://www.anchor.com.au

Jay Coles
code@pocketnix.org
http://doger.io

Background

- Asked to play with OpenVZ in 2010
- Wrote proof of concept “isolation”
- Wrote a more complete implementation “Asylum” in 2011
- Waited for more complete containers implementation
- Currently working on next gen code “Hammerhead”

So what is this containers thing anyway?

- Linux Namespace
- Check Point / Restore (CRIU)
- Standard Linux Networking
- Standard Linux Resource Management
- Standard Linux Security Mechanisms

Jay Coles
code@pocketnix.org
http://doger.io

anchor

Containers: A new Hope

- Chroot (poor mans container)
- FreeBSD Jails
- Solaris Zones
- OpenVZ
- Linux Vserver
- UML

Jay Coles
code@pocketnix.org
http://doger.io

anchor

Namespaces: The Tux strikes back

- Presents a subset of host resources to a set of processes
- Highly granular
- Allows picking and choosing
  - In practice, only a few combinations useful
- Align with Logically distinct subsystems
- Not everything is namespace aware

Jay Coles
code@pocketnix.org
http://doger.io

anchor

Return of the namespaces

- Chroot
- Mount
- UTS/Hostname
- IPC
- Network
- PID
- UID

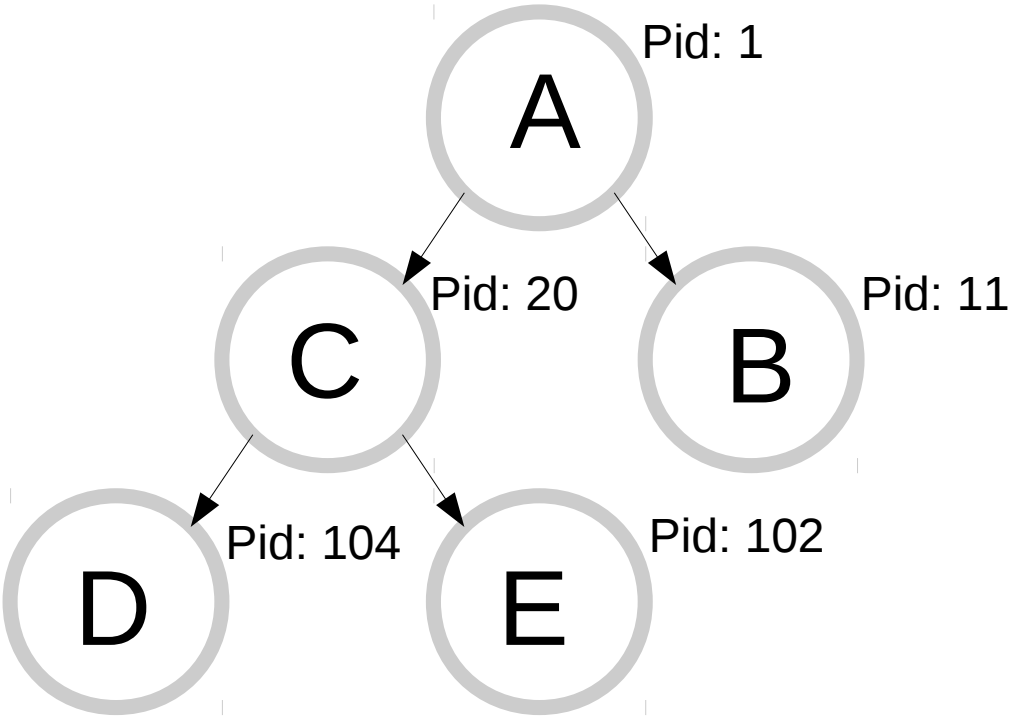
Return of the namespaces

- Chroot (Hierarchy)
- Mount (Isolated/Cloned)
- UTS/Hostname (Isolated)
- IPC (Isolated)
- Network (Isolated)
- PID (Hierarchy)
- UID (Hierarchy)

Jay Coles
code@pocketnix.org
http://doger.io

anchor

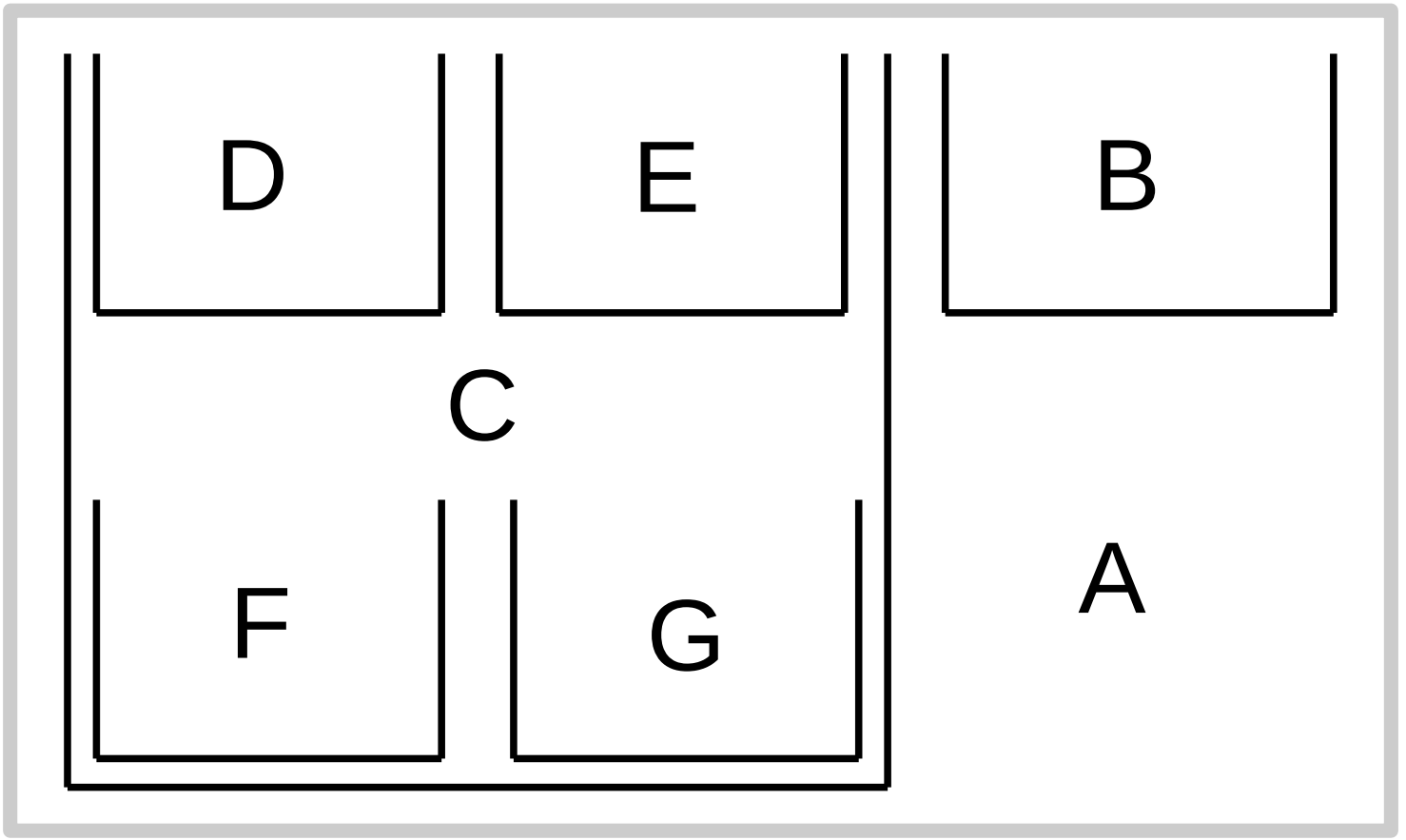
Return of the namespaces



Jay Coles
code@pocketnix.org
http://doger.io

Return of the namespaces

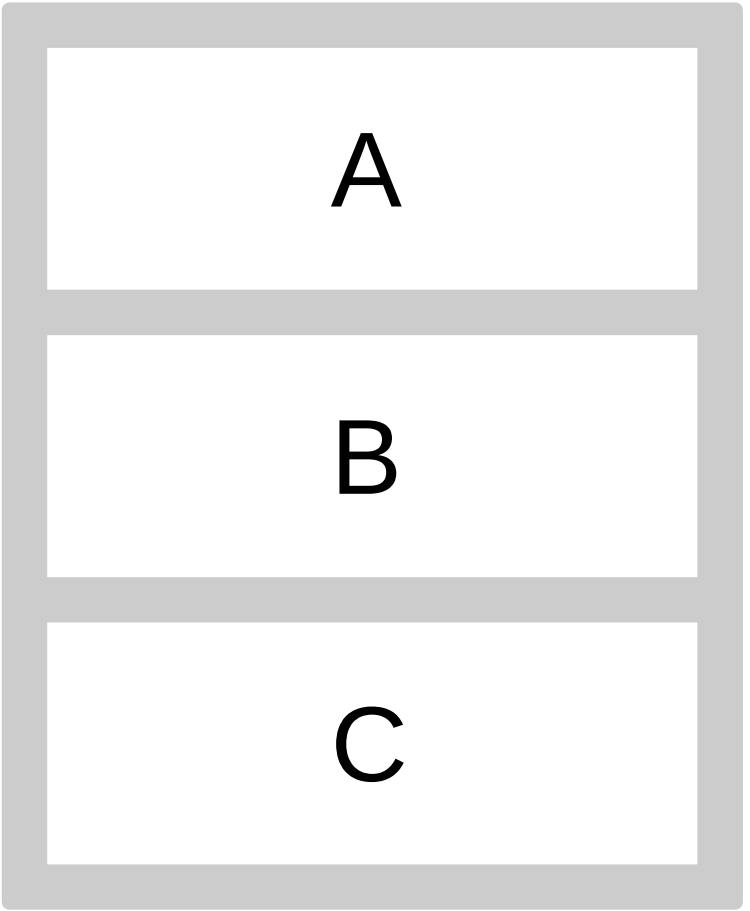
Parent can see child resources



Jay Coles
code@pocketnix.org
http://doger.io

Return of the namespaces

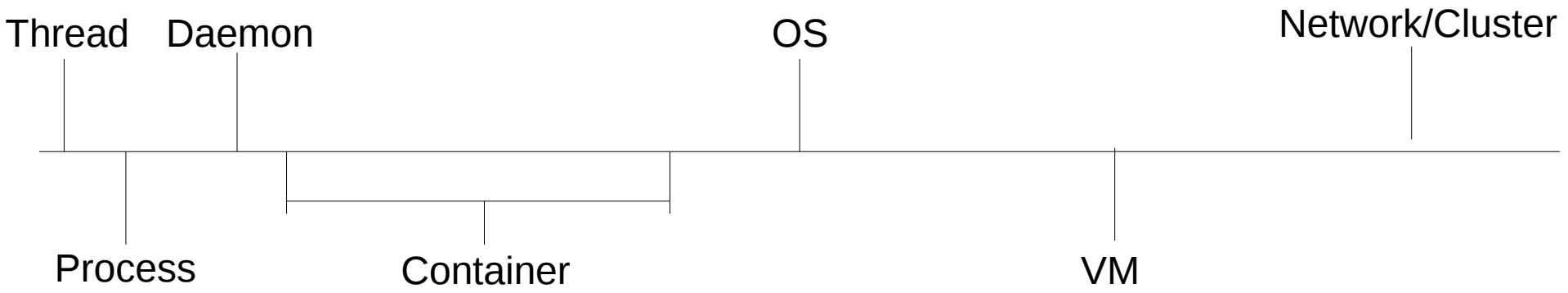
Completly isolated



Jay Coles
code@pocketnix.org
http://doger.io

anchor

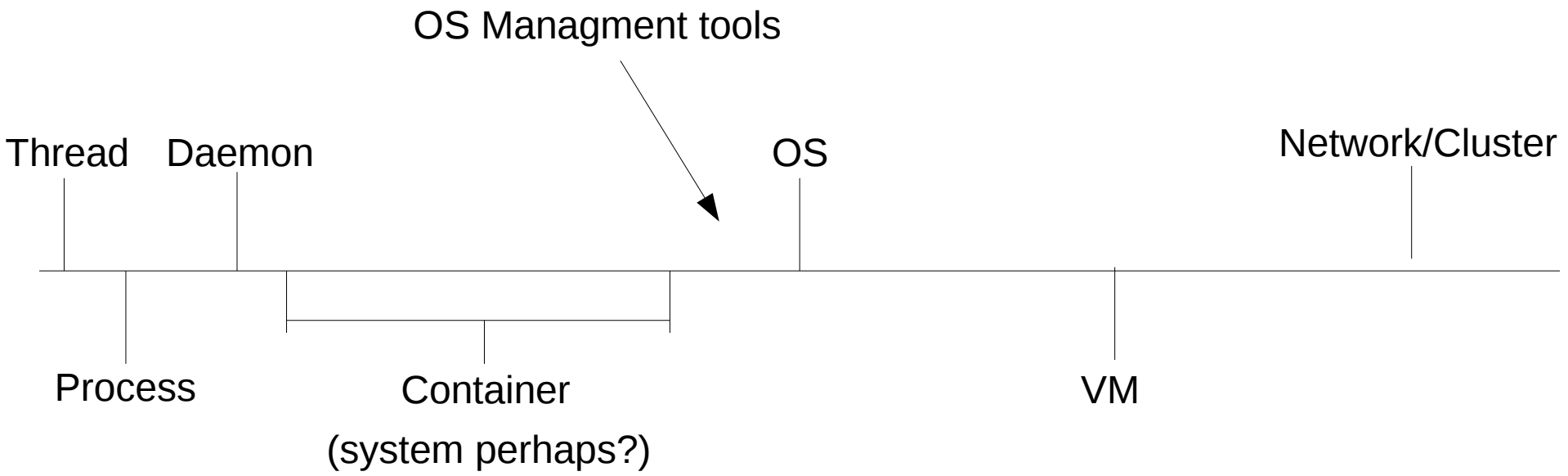
Container Sweet spot



Jay Coles
code@pocketnix.org
http://doger.io

anchor

Container Sweet spot



Jay Coles
code@pocketnix.org
http://doger.io

anchor

Sweet spot expanded

- Move machine/OS management tools outside container (makes / smaller)
- Move remote access outside container (ie ssh) and setns to container on login
- Move Network management upstream
- Basically strip out administrative stuff and make containers only about the apps/userland you want to execute

Attack of the clone()

- clone()
- unshare()
- setns()
- sethostname()
- getpid()
- prctl()

Jay Coles
code@pocketnix.org
http://doger.io

anchor

Attack of the clone(): clone/unshare

- Pass in flags, get New namespace
- Clone() is used to implement fork() with glibc
- Glibc gets in your way

Jay Coles
code@pocketnix.org
http://doger.io

Attack of the clone(): clone/unshare

Pass in flags, get New namespace

- Clone() is used to implement fork() with glibc
- Glibc gets in your way
  - Forces you to specify a base address for the stack. Linux allows a null pointer for COW semantics

Attack of the clone(): clone/unshare

- Pass in flags, get New namespace
- Clone() is used to implement fork() with glibc
- Glibc gets in your way
  - Solution: Just call sys_clone() directly

Attack of the clone(): clone/unshare

- Pass in flags, get New namespace
- Clone() is used to implement fork() with glibc
- Glibc gets in your way
  - Solution: Just call sys_clone() directly
- Unshare sets up a namespace, but only child processes are inside it

Attack of the clone(): clone/unshare

- Fork model allows your management code to be outside container so init is pid 1
- Clone() sounds good but recommend using unshare()
  - Easier to spawn multiple separate processes in namespace
- Bind mount namespaces to keep alive after pid 1 has exited (allows setns'ing into, no need to set up network again)

Attack of the clone(): setns

- Allows you to enter a namespace
- Like unshare, child processes are in the namespace
- Give it an FD to a namespace (/proc/self/ns) to enter it
- Consider this as an alternative to ssh'ing into a guest (why do containers need ssh anyway?)

Attack of the clone()

- Sethostname

  - dont call /bin/hostname, easier to just syscall it (cant trust container binaries during setup/unsecured phase)

Attack of the clone()

- Sethostname
- Getpid
  - Glibc caches output
  - Calling sys_clone() with some flags will not invalidate the cache

Attack of the clone()

- Sethostname
- Getpid
  - Solution: Bypass glibc

Attack of the clone()

- Sethostname
- Getpid
- Prtcl
  - Lots of useful things in there
  - Seccomp NO_NEW_PRIVS and capability dropping are the most useful things here

Revenge of the namespaces

- /proc/self/ns
- /proc/self/uid_map
- /proc/self/gid_map
- /proc/self/net
- /proc/self/mounts

Jay Coles
code@pocketnix.org
http://doger.io

The Network

- Veth
  - Virtual ethernet pipe
  - Useful to join containers together or connect to the host
  - Use with bridges
- MacVLAN
  - Allows direct sharing of real ethernet device
  - Low overhead (for a few devices)
  - May switch to software processing for > 10 interfaces

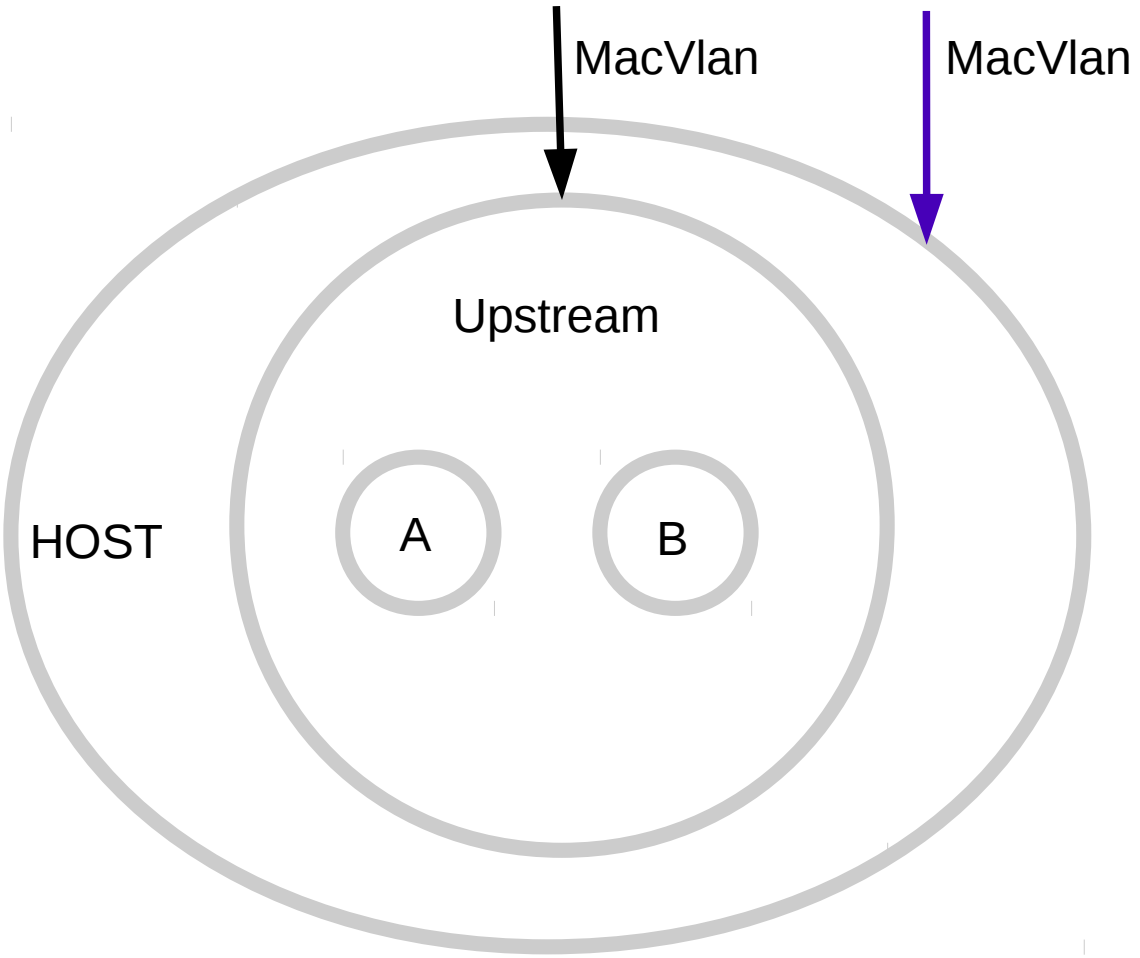
The Network Reloaded

- Why does firewalling have to be in the container
- Extend to routing, do routing external to container
- Why does the Host have to route packets?
- VPN Daemons dont need to live in containers they serve

Jay Coles
code@pocketnix.org
http://doger.io

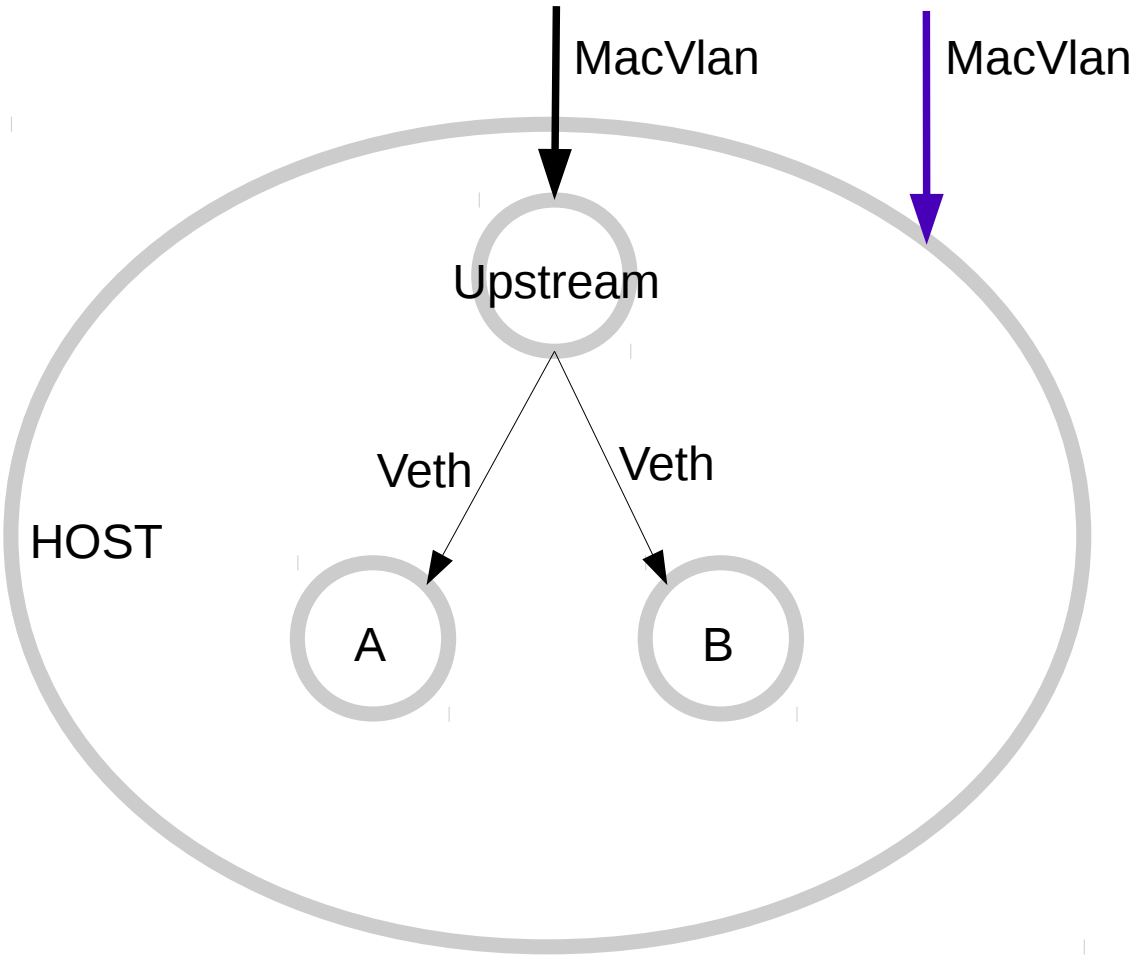
anchor

The Network Revolutions



Jay Coles
code@pocketnix.org
http://doger.io

The Network Revolutions



Jay Coles
code@pocketnix.org
http://doger.io

Security

- Multilayer defense
- No one security mechanism does 100% of what you need/want
- You are going to have to make decisions/trade offs

Security

- Multilayer defense
- No one security mechanism does 100% of what you need/want
- You are going to have to make decisions/trade offs
  - I Highly recommend thinking about throwing out some of the traditional ways to manage machines in order to manage containers

Security

- CGroups
- Capabilities
- Seccomp v2
- SELINUX

Jay Coles
code@pocketnix.org
http://doger.io

CGroups Begins

- Resource accounting
- Resource Limiting
- Basic Security features (Device cgroup)
- “Not the resource management system we want, but the management resource we need”

The Dark Capabilities

- CAP_SYS_ADMIN is overpowered
  - Not actually needed for a container while running
  - Consider setns()ing into a container for admin tasks without dropping capabilities
- Many are not needed
- Couple may be useful (CAP_SYS_BOOT in kernels > 3.8, CAP_SYS_CHROOT)

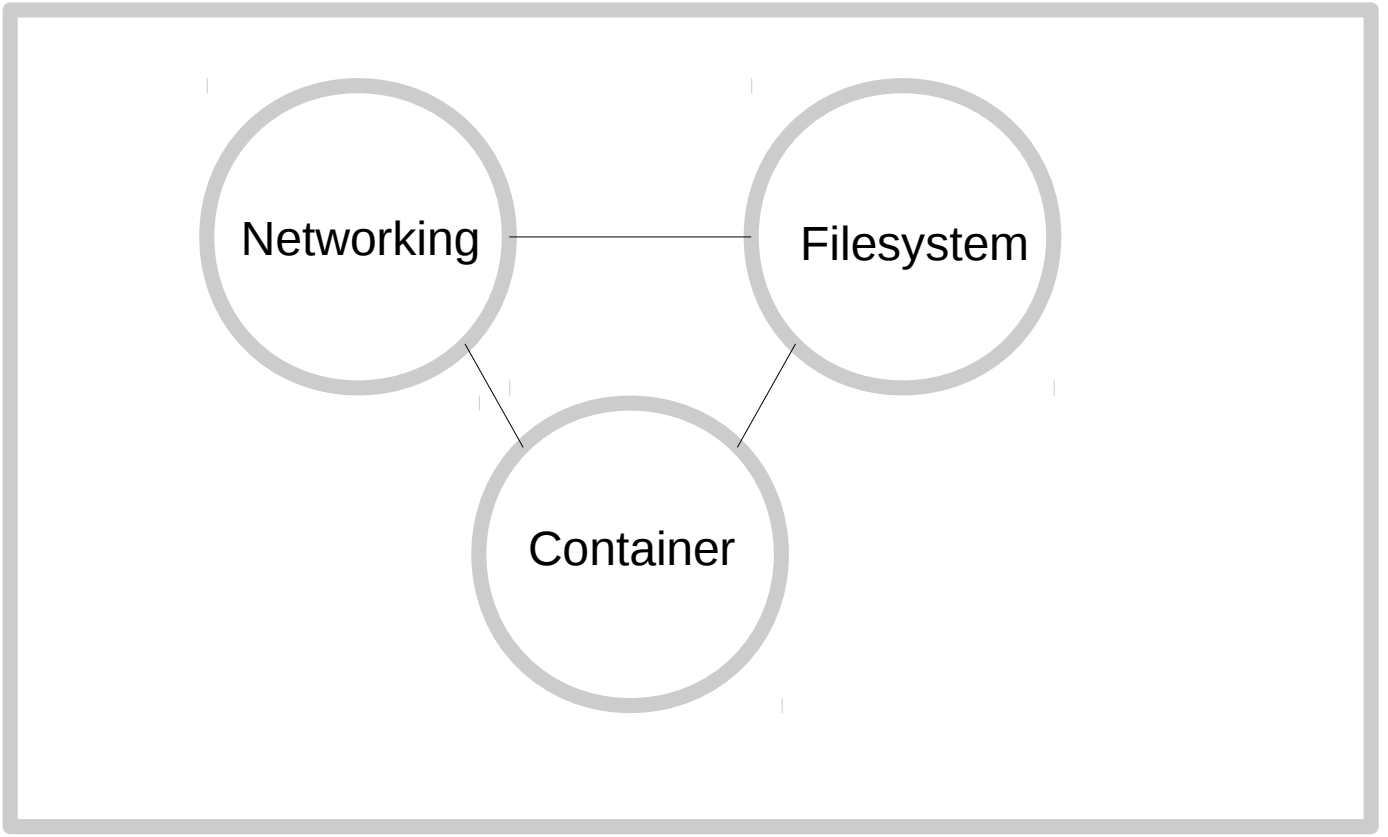
The Seccomp v2 Rises

- Originally proposed for secure computation
- v2 Uses BPF to filter syscalls
- More granular than allow/deny on per syscall basis (can perform actions based on arguments)
- Reduce attack surface (eg limit ioctls allowed without disabling ioctl completely)

SELINUX

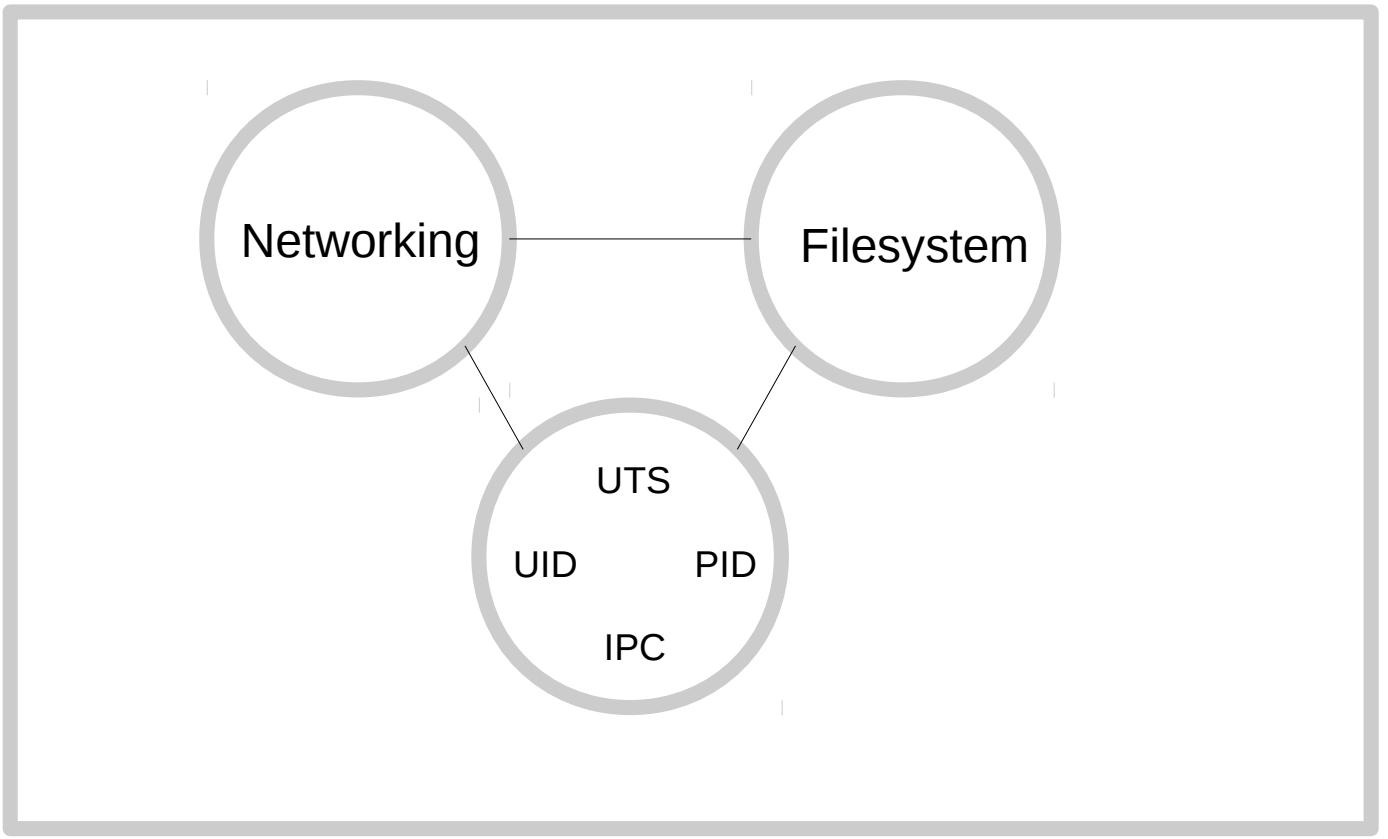
- MCS - Used to isolate containers from each other
- Useful for handing out things like bind mounted CGroups (layer of defense in case of bad programming)
- Just keep it simple and use it for isolation not for locking a container down

Logical Subsystems



Jay Coles
code@pocketnix.org
http://doger.io

Logical Subsystems



Jay Coles
code@pocketnix.org
http://doger.io

Can I have a Pony?

- CLONE_STOPPED
  - This flag was deprecated from Linux 2.6.25 onward, and was removed altogether in Linux 2.6.38
- Easy way to get the PID's of the children of a process

More Information

- http://doger.io
- http://www.pocketnix.org
- Containers mailing list
- LWN
- Man pages
- Talk to me at lunch
- Email me