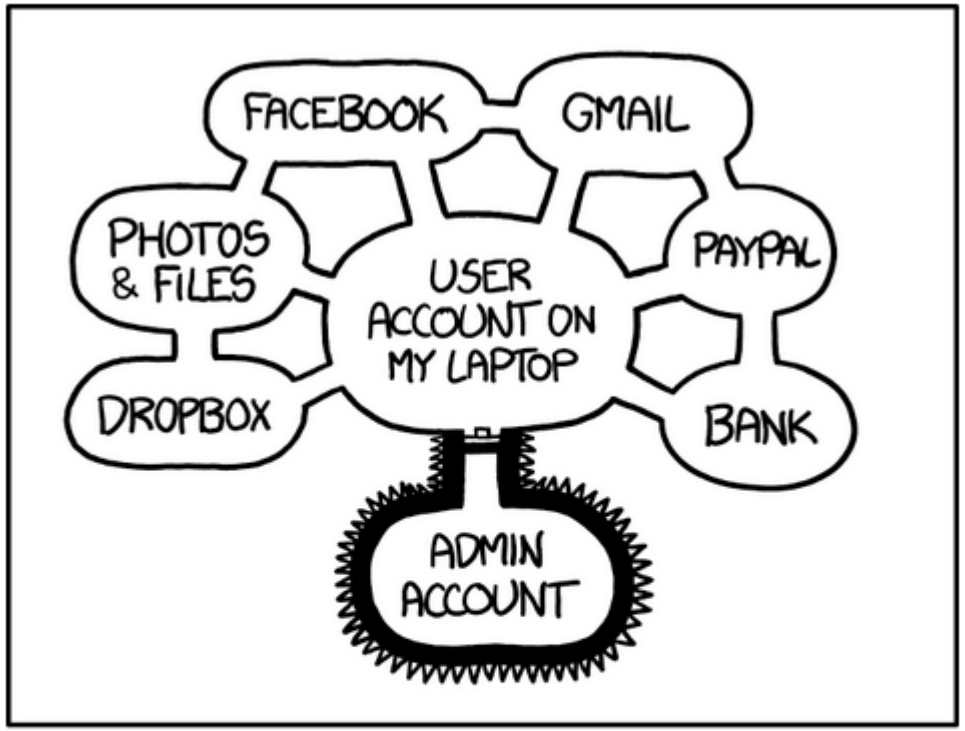

# Jails

## Lightweight, Operating-System-level virtualization

Oct 2018

Fehmi Noyan ISI

# So, what was the motivation




IF SOMEONE STEALS MY LAPTOP WHILE I'M LOGGED IN, THEY CAN READ MY EMAIL, TAKE MY MONEY, AND IMPERSONATE ME TO MY FRIENDS,

BUT AT LEAST THEY CAN'T INSTALL DRIVERS WITHOUT MY PERMISSION.

https://xkcd.com/1200/

- UNIX security model and the root user – sharp, efficient and an extremely dangerous tool :/
- Real isolation?
  - File system access limitations
  - Process isolation
  - Network stack isolation

# chroot(8)

[CHROOT]

> Dr. Marshall Kirk Mckusick, private communication: "According to the SCCS logs, the chroot call was added by Bill Joy on March 18, 1982 approximately 1.5 years before 4.2BSD was released. That was well before we had ftp servers of any sort (ftp did not show up in the source tree until January 1983). My best guess as to its purpose was to allow Bill to chroot into the /4.2BSD build directory and build a system using only the files, include files, etc contained in that tree. That was the only use of chroot that I remember from the early days."

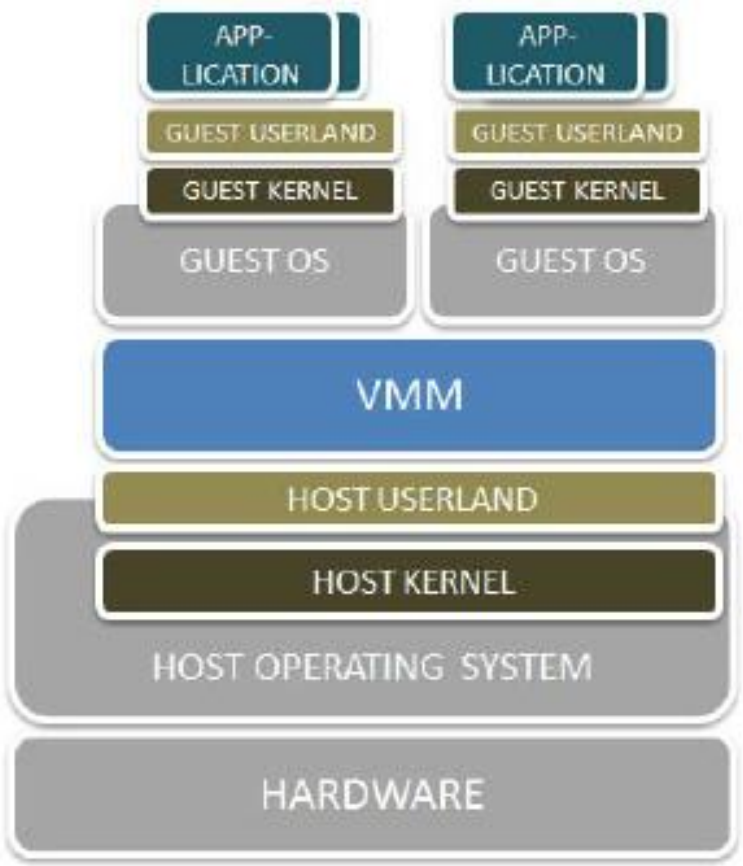
[3]

# Solution?

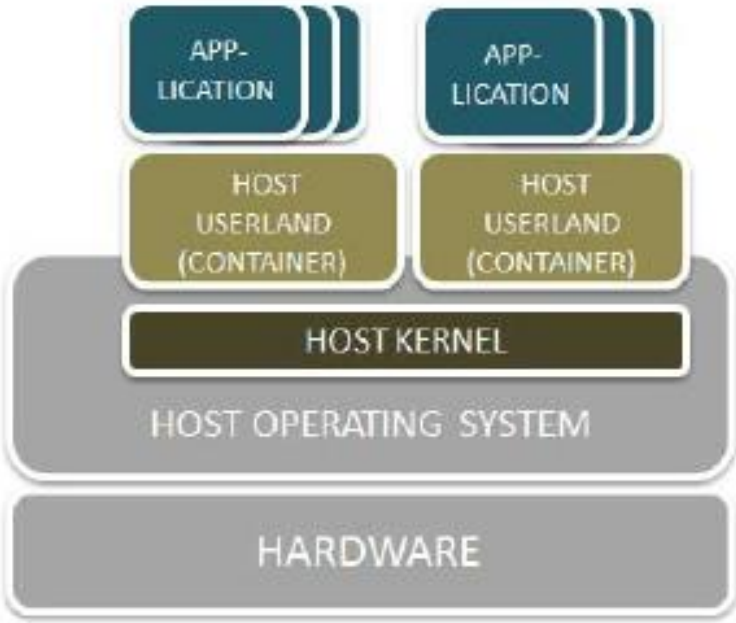
- File system access limitations
- Process isolation
- Network stack isolation

# Operating System level virtualization

- Jails
- Docker
- Zones



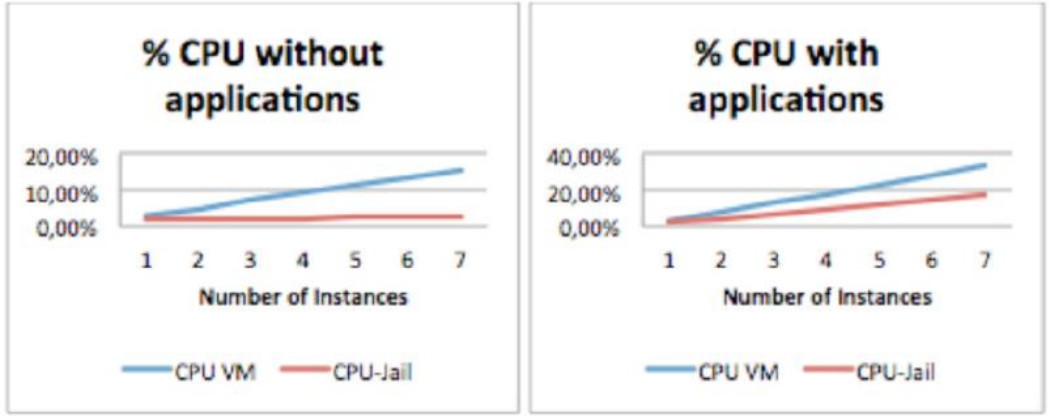

Virtualization via hypervisor (type 2)




Virtualization via isolated OS containers

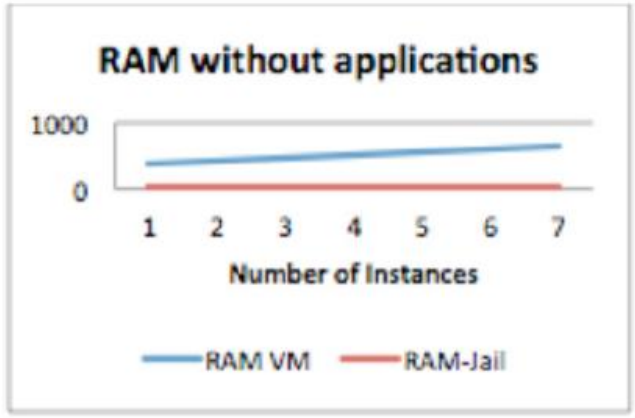
[1]

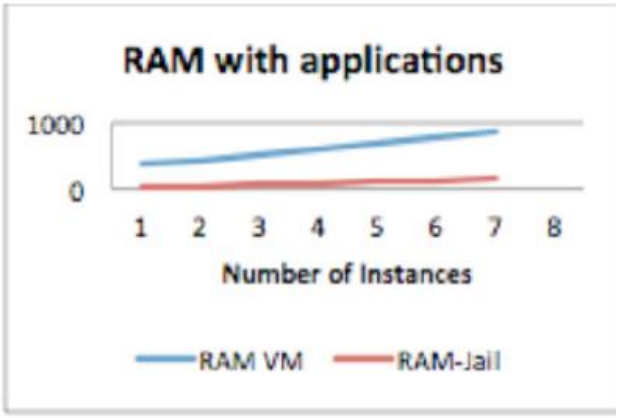
# Operating System level virtualization



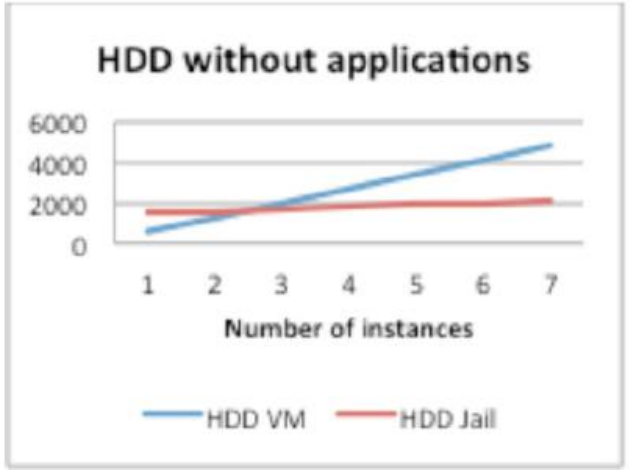


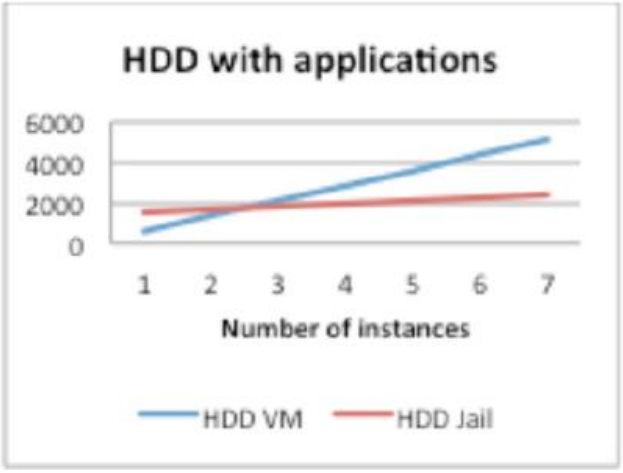




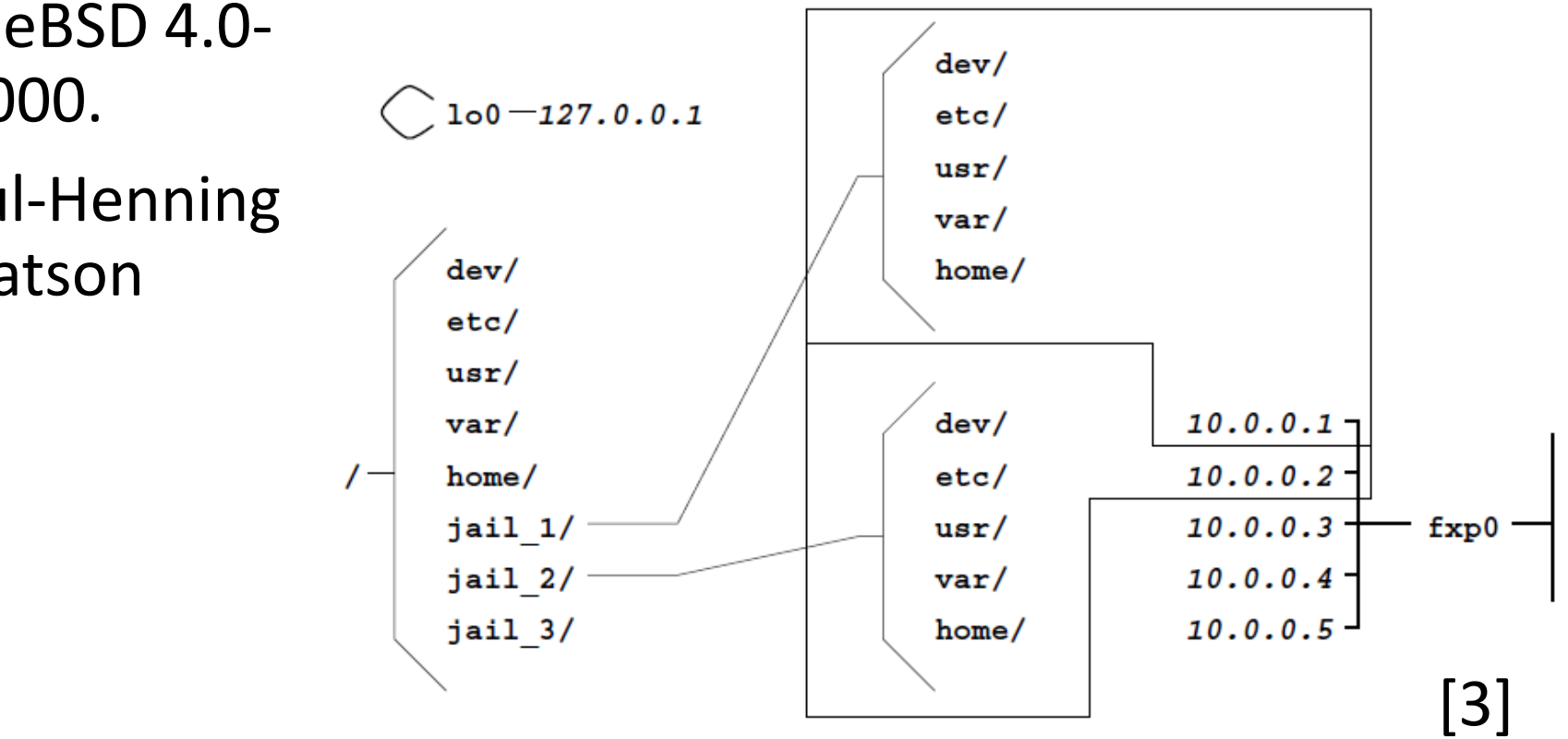

With increasing number of virtual environments with the usage of virtual machines it triggers the resource consumption. While with the use of ambient jail these resources are mainly channelled to the services provided, and the number of instances has a negligible impact on the computer system [6]



# jail(8)

- First appeared in FreeBSD 4.0-RELEASE in March 2000.
- Implemented by Poul-Henning Kamp and Robert Watson
- Provides
  - Virtualization
  - Security
  - Delegation




[3]

# Limitations in a jail

- File system name-space is restricted in the style of chroot(2)
- The ability to bind network resources is limited to jail IP addresses only
- The ability to manipulate system resources is limited
- IPC is limited to communication with the processes in the same jail

# Implementation of jail(8)

- New system calls (jail(2), jail_attach(2) etc.) and data structures
- Fortification of chroot(2)
- Process visibility restrictions (prison_check(2))
- TCP/IP network stack isolation
- Adding jail awareness to some device drivers
- Restriction of super-user powers for jailed root

# Implementation of jail(8)

1 ) jail creation, jail(8) and jail(2)

2 ) Attaching, jexec(8) and jail_attach(2)

```
/usr/src/usr.sbin/jexec/jexec.c
jid = jail_getid(argv[0]);
jail_attach(jid);
chdir("/");
execvp(argv[1], argv + 1);
```

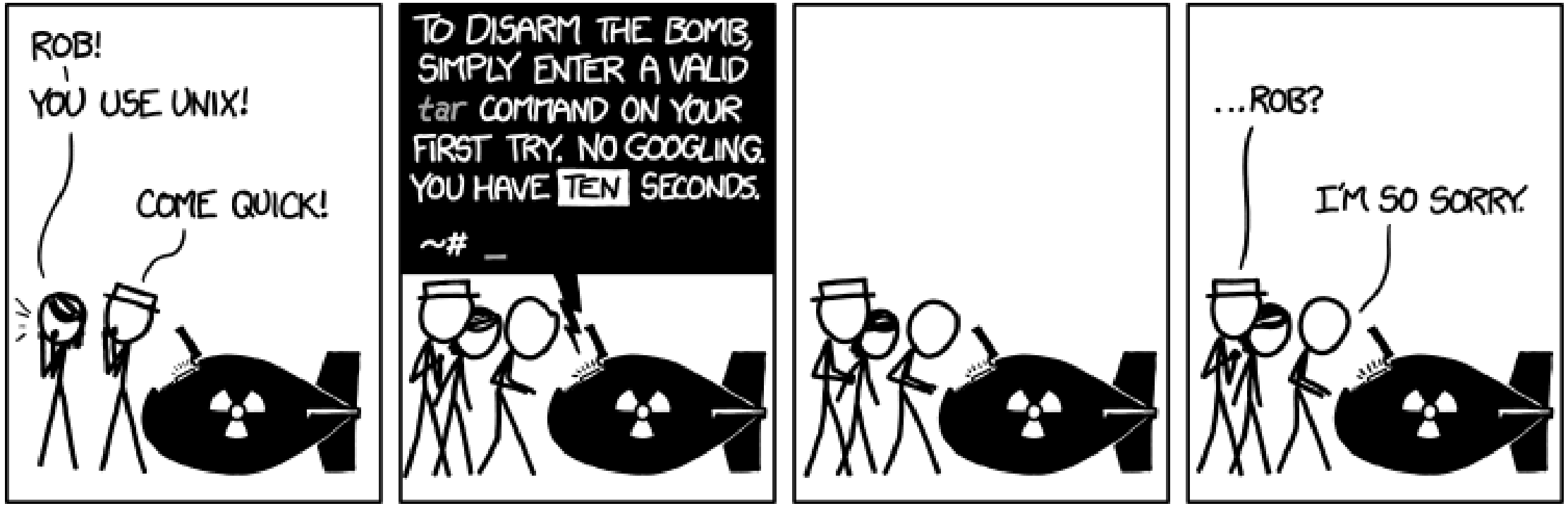
# Fine tuning

1 ) rctl(8)

2 ) sysctl(8)

- `security.jail.set_hostname_allowed: 1`
- `security.jail.socket_unixiproute_only: 1`
- `security.jail.sysvipc_allowed: 0`
- `security.jail.enforce_statfs: 2`
- `security.jail.allow_raw_sockets: 0`
- `security.jail.chflags_allowed: 0`
- `security.jail.jailed: 0`

# Live demo




https://www.explainxkcd.com/wiki/index.php/1168:_tar

# References

[1] Srivastava P, Pande S, A novel architecture for identity management system using virtual appliance technology

[2] Cantrill B, Jails and Solaris Zones

[3] Kamp P, Watson R, Jails: Confining the omnipotent root

[4] The FreeBSD Documentation Project, FreeBSD Architecture Handbook

[5] The FreeBSD Documentation Project, FreeBSD Handbook

[6] Antunes C, Vardasca R, Performance of Jails versus Virtualization for Cloud Computing Solutions