



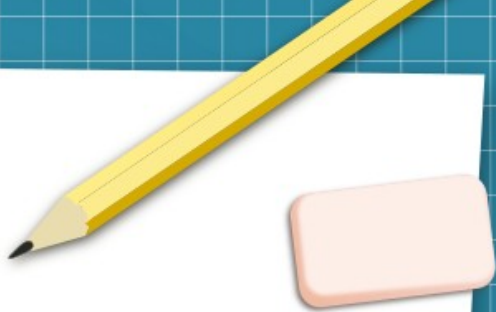
Capabilities, isolation mechanisms

CSE 536 Fall 2026
jedimaestro@asu.edu

Capabilities



- Jerome Saltzer and Michael Schroeder's 1975 seminal paper, "The Protection of Information in Computer Systems," defined the principle of least privilege:
 - Users and programs should operate with the bare minimum privileges required to perform their jobs.
- Before Linux kernel version 2.2 it was root or nothing in Linux
 - A lot of the info that follows about Linux capabilities is from...
 - <https://earthly.dev/blog/intro-to-Linux-capabilities/>



Capability Myths Demolished

Mark S. Miller

Combex, Inc.
markm@caplet.com

Ka-Ping Yee

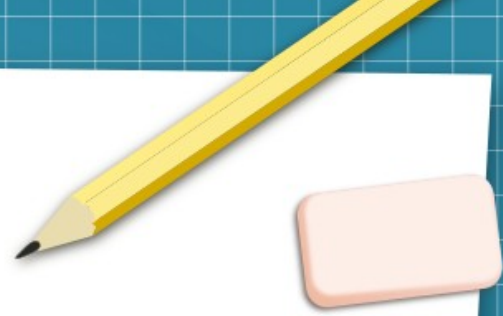
University of California, Berkeley
ping@zesty.ca

Jonathan Shapiro

Johns Hopkins University
shap@cs.jhu.edu

<https://papers.agoric.com/assets/pdf/papers/capability-myths-demolished.pdf>

A little history...



Plessey System 250 (1969)



SYSTEM
250

<https://archive.org/details/plessey-system-250-general-introduction>

<https://retrocomputingforum.com/t/a-capability-based-computer-from-britain-early-1970s/2602/2>

Cambridge CAP (1970 to 1977)

https://en.wikipedia.org/wiki/CAP_computer



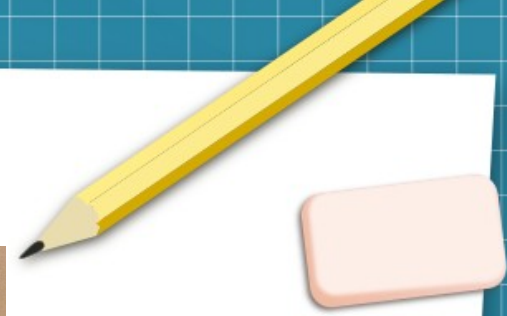
IBM System/38 (1978 to 1988)

https://en.wikipedia.org/wiki/IBM_System/38



IBM AS/400

https://en.wikipedia.org/wiki/IBM_AS/400



rpgle

```
H* Column 6 is the Spec Type (H, F, I, O)
FCUSTMAS IP  E          K          DISK
FQSYSPRT O   F      132          PRINTER
I* Input Specs define fields if not externally described
ICUSTREC
I          CUSTID          1      5
I          NAME            6     30
I          CITY            31     50
C* The Logic Cycle handles the 'READ' and 'LOOP' automatically
C* We only code specific calculations here if needed.
C* 'Detail' output happens for every record read.
O* Output Specs define the report layout
OQSYSPRT H  101    1P
O* 1P is 'First Page' indicator for headings
O          10 'CUSTOMER LIST'
O          D  1
O* D is 'Detail' - prints once for every record in the cycle
O          CUSTID          5
O          NAME            35
O          CITY            60
```

IBM Power System i / iSeries



<https://store.flagshiptech.com/ibm-power-system-i-series/?srsltid=AfmBOopJuDQbwA1x9vSHW9bXdiy9aGMGJcE8Op2vksJB8xVakQZZVJUJ>

Intel iAPX 432

https://en.wikipedia.org/wiki/Intel_iAPX_432





```
procedure Change_Calculator is
  type Money is delta 0.01 digits 10;
  Input_Amount : Money := 0.0;
  package Money_IO is new Text_IO.Decimal_IO(Money);

  type Currency_Denomination is record
    Name : String(1..10);
    Value : Money;
  end record;

  Currency_Names : array (1..10) of Currency_Denomination;
  Currency_Counts : array (1..10) of Integer := (others=>0);
  Currency_Index : Integer range 1..10;

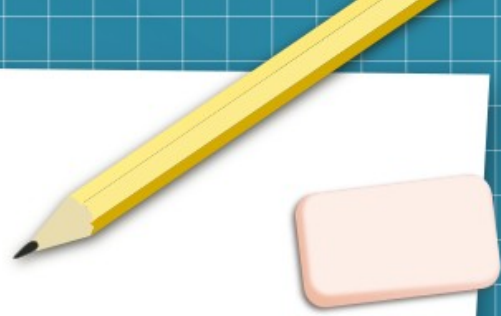
  procedure Get_Money_Prompt(Amount: out Money) is
    Response : String(1..20);
    Last : Natural;
```

Elbrus E2K, etc.

<https://www.tomshardware.com/news/russian-company-tapes-out-16-core-elbrus-cpu-20-ghz-16-tb-of-ram-in-4-way-system>

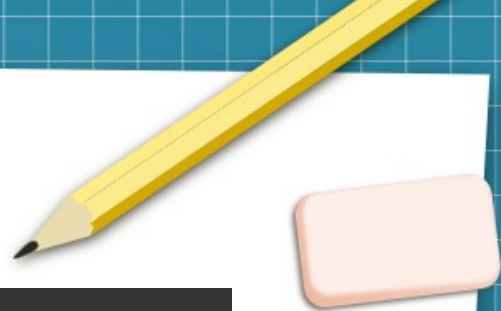


OSes...



- KeyKOS / EROS
- SeL4
 - In lots of cars as SkyOS-M
- CHERI with FreeBSD
- Capsicum
 - Ships with FreeBSD to sandbox, *e.g.*, tcpdump
- <https://homes.cs.washington.edu/~levy/capabook/>

Linux capabilities



```
root@tortuga:~# getcap /usr/bin/*  
/usr/bin/dumpcap cap_net_admin,cap_net_raw=eip  
/usr/bin/mtr-packet cap_net_raw=ep  
/usr/bin/ping cap_net_raw=ep  
root@tortuga:~#
```

```
root@tortuga:~# getcap -r / 2>/dev/null  
/usr/bin/dumpcap cap_net_admin,cap_net_raw=eip  
/usr/bin/mtr-packet cap_net_raw=ep  
/usr/bin/ping cap_net_raw=ep  
/usr/lib/x86_64-linux-gnu/libexec/kf5/start_kdeinit cap_sys_resource=ep  
/usr/lib/x86_64-linux-gnu/gstreamer1.0/gstreamer-1.0/gst-ptp-helper cap_net_bind  
_service,cap_net_admin=ep
```

```
jedi@tortuga:~$ ls -l /usr/bin/ping
-rwxr-xr-x 1 root root 76680 Jul 24 2025 /usr/bin/ping
jedi@tortuga:~$ ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=117 time=34.3 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=117 time=14.1 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=117 time=18.3 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=117 time=37.3 ms
64 bytes from 8.8.8.8: icmp_seq=5 ttl=117 time=13.8 ms
64 bytes from 8.8.8.8: icmp_seq=6 ttl=117 time=27.6 ms
64 bytes from 8.8.8.8: icmp_seq=7 ttl=117 time=31.2 ms
64 bytes from 8.8.8.8: icmp_seq=8 ttl=117 time=20.1 ms
^Z
[1]+  Stopped                  ping 8.8.8.8
jedi@tortuga:~$ pidof ping
8571
jedi@tortuga:~$
```



jedi@tortuga: ~



```
jedi@tortuga:~$ getpcaps -p 8571
```

```
-p: =
```

```
8571: cap_net_raw=p
```

```
jedi@tortuga:~$ less /proc/8571/status
```

```
Name:      ping
Umask:     0002
State:     T (stopped)
Tgid:      8571
Ngid:       0
Pid:       8571
PPid:      8470
TracerPid: 0
Uid:       1000    1000    1000    1000
Gid:       1000    1000    1000    1000
FDSize:    256
Groups:    4 27 123 1000
NStgid:    8571
NSpid:     8571
NSpgid:    8571
NSsid:     8470
Kthread:   0
VmPeak:    20132 kB
VmSize:    20132 kB
VmLck:     0 kB
VmPin:     0 kB
VmHWM:     2832 kB
VmRSS:     2832 kB
/proc/8571/status
```



jedi@tortuga: ~



```
VmData:      236 kB
VmStk:       132 kB
VmExe:        44 kB
VmLib:      2044 kB
VmPTE:        64 kB
VmSwap:        0 kB
HugetlbPages:      0 kB
CoreDumping:      0
THP_enabled:      1
untag_mask: 0xffffffffffffffff
Threads:          1
SigQ:    0/124495
SigPnd: 0000000000000000
ShdPnd: 0000000000000000
SigBlk: 0000000000000000
SigIgn: 0000000000000000
SigCgt: 0000000000002006
CapInh: 0000000000000000
CapPrm: 0000000000002000
CapEff: 0000000000000000
CapBnd: 000001fffffffffff
CapAmb: 0000000000000000
NoNewPrivs:      0
:█
```

```
jedi@merced: /usr/bin — passwd
/usr/bin

jedi@merced:/usr/bin$ ls -la | grep "^...s"
-rwsr-sr-x  1 daemon daemon    55696 Feb  3  2026 at
-rwsr-xr-x  1 root  root      72840 Feb  2  2026 chfn
-rwsr-xr-x  1 root  root      48936 Feb  2  2026 chsh
-rwsr-xr-x  1 root  root      39376 Mar 21  2026 fusermount3
-rwsr-xr-x  1 root  root      76328 Feb  2  2026 gpasswd
-rwsr-xr-x  1 root  root      51664 Aug 19 07:23 mount
-rwsr-xr-x  1 root  root     167000 Jul 11 08:55 ntfs-3g
-rwsr-xr-x  1 root  root      93640 Feb  2  2026 passwd
-rwsr-xr-x  1 root  root      31024 Sep 11 10:30 pkexec
-rwsr-xr-x  1 root  root      55760 Aug 19 07:23 su
-rwsr-xr-x  1 root  root     282080 Oct  6 06:13 sudo.ws
-rwsr-xr-x  1 root  root      39376 Aug 19 07:23 umount
jedi@merced:/usr/bin$ passwd
Changing password for jedi.
Current password: 
```



jedi@merced: ~



```
jedi@merced:~$ pidof passwd
```

```
618648
```

```
jedi@merced:~$ getpcaps -p 618648
```

```
Cannot parse pid -p: (Invalid argument)
```

```
618648: =ep cap_wake_alarm+i
```

```
jedi@merced:~$ getpcaps 618648
```

```
618648: =ep cap_wake_alarm+i
```

```
jedi@merced:~$
```

Because the output says `=ep` (meaning *all* capabilities are Effective and Permitted), minus the `cap_wake_alarm` exception, the `passwd` process **currently possesses 39 out of the 40 standard Linux capabilities** in its active sets.  Medium

This includes powerful administrative capabilities such as:

- `CAP_DAC_OVERRIDE` : Bypasses all file read, write, and execute permission checks (allowing it to modify `/etc/shadow`).
- `CAP_SYS_ADMIN` : The "catch-all" root capability for system configuration.
- `CAP_SETUID` / `CAP_SETGID` : Allows the process to forge user and group IDs.
- *...and nearly every other capability required to function with full root authority.*  man7.org +1

```
Name:      passwd
Umask:     0077
State:     S (sleeping)
Tgid:      8777
Ngid:      0
Pid:       8777
PPid:      8478
TracerPid: 0
Uid:       1000      0      0      0
Gid:       1000      1000    1000    1000
FDSize:    256
Groups:    4 27 123 1000
NStgid:    8777
NSpid:     8777
NSpgid:    8777
NSsid:     8478
Kthread:   0
VmPeak:    24016 kB
VmSize:    24016 kB
VmLck:     0 kB
VmPin:     0 kB
VmHWM:     4836 kB
VmRSS:     4836 kB
/proc/8777/status
```



jedi@tortuga: ~



```
VmData:      600 kB
VmStk:       132 kB
VmExe:        28 kB
VmLib:      4980 kB
VmPTE:        72 kB
VmSwap:        0 kB
HugetlbPages:      0 kB
CoreDumping:      0
THP_enabled:      1
untag_mask: 0xffffffffffffffff
Threads:          1
SigQ:    0/124495
SigPnd: 0000000000000000
ShdPnd: 0000000000000000
SigBlk: 0000000000008000
SigIgn: 0000000000287007
SigCgt: 0000000000000000
CapInh: 0000000000000000
CapPrm: 000001ffffffffffff
CapEff: 000001ffffffffffff
CapBnd: 000001ffffffffffff
CapAmb: 0000000000000000
NoNewPrivs:      0
:█
```



root@tortuga: ~



```
root@tortuga:~# for i in /proc/[0-9]*; do p=`basename $i`; echo $p; getpcaps $p;  
done | grep "cap" | less
```

```
1080: cap_audit_write=eip
1081: cap_dac_override,cap_kill,cap_setgid,cap_setuid,cap_net_bind_service,cap_net_admin,cap_net_raw,cap_sys_module,cap_sys_chroot,cap_audit_write=ep
1084: =ep cap_sys_module,cap_sys_rawio,cap_mknod=ep
1090: cap_net_bind_service,cap_net_admin=ep
1127: =ep cap_sys_module=ep
1139: cap_chown,cap_dac_override,cap_dac_read_search,cap_fowner,cap_linux_immutable,cap_sys_admin,cap_sys_tty_config,cap_audit_control,cap_mac_admin=ep
1213: cap_net_admin,cap_sys_admin=ep
1223: cap_chown,cap_dac_override,cap_dac_read_search,cap_fowner,cap_fsetid,cap_kill,cap_setgid,cap_setuid,cap_net_bind_service,cap_audit_write,cap_wake_alarm=ep
1270: cap_net_bind_service,cap_sys_time=ep
1274: cap_net_bind_service,cap_sys_time=ep
1679: cap_dac_read_search,cap_sys_nice=ep
2211: cap_net_bind_service=ep
2922: =ep cap_sys_module=ep
3709: cap_sys_admin=ep
616: cap_chown,cap_dac_override,cap_dac_read_search,cap_fowner,cap_setgid,cap_setuid,cap_sys_ptrace,cap_sys_admin,cap_audit_control,cap_mac_override,cap_syslog,cap_audit_read=ep
652: =ep cap_sys_time,cap_wake_alarm=ep
990: cap_net_raw=eip
```

~

(END)

Info about a few of those processes...



- 1139 is systemd-logind running as root
- 1080 is dbus-broker running as user “messagebus”
- 1084 is accounts-daemon running as root
- 1223 is cupsd (printer) running as root
- 1270 is chronyd (NTP) running as user “_chrony”

More uses of Linux capabilities



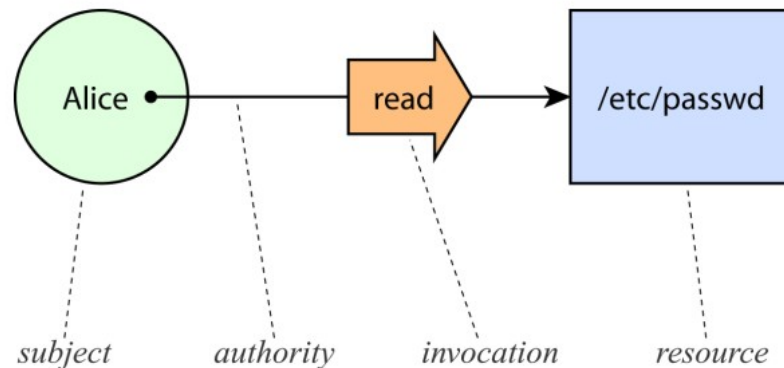
- Apache web server can bind to ports 80 and 443 without root privileges
- Docker/Kubernetes

Linux capabilities are not “real” capabilities, something more like Capsicum in FreeBSD is.



Ambient authority

In a capability system, each capability points from a subject to a resource. Consequently every capability can serve both to designate which resource to access, and to provide the authority to perform that access. This change provides us with the option to avoid introducing a shared namespace into the foundations of the model, and thereby avoid the complex issues involved in managing a shared namespace – issues rarely acknowledged as a cost of non-capability models.



FreeBSD: tcpdump inside Capsicum

Example: `tcpdump -n -i em0`

1. Prepare access

Open capture device (BPF)

Keep terminal output

Limit BPF descriptor rights

```
cap_rights_limit(...)
```

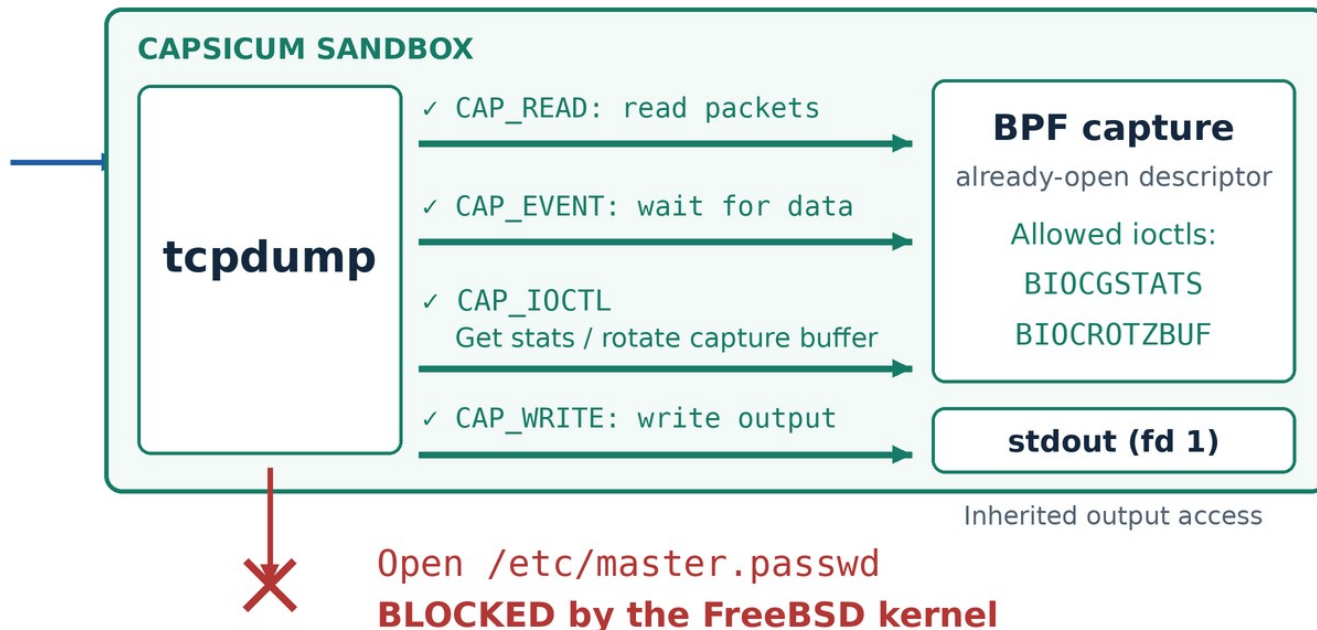
```
cap_ioctls_limit(...)
```

Capability = resource + rights

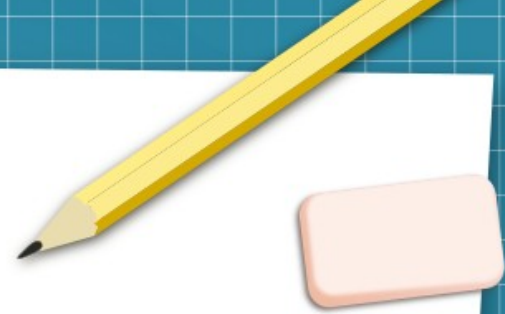
Green shows permitted access.

2. Enter capability mode

`cap_enter()`



Containers (e.g., Docker)



- **Namespaces control what a process can see.**
 - Network namespace: interfaces, IP addresses, routes, firewall rules, sockets, port-number space
 - PID namespace: process IDs
 - Mount namespace: filesystem mounts
 - IPC, UTS/hostname, user, time, and cgroup namespaces
- **cgroups control how much a process can use.**
 - CPU, memory, I/O, process counts, and accounting
- **Containers combine those mechanisms.**
 - Docker or Podman creates namespaces for isolation
 - Places processes into cgroups for resource control
 - Adds filesystems, capabilities, seccomp, and other security policy

Mechanism	What it controls
Namespaces	What processes can see
cgroups	How much CPU, memory, etc. they can consume
Linux capabilities	Which privileged operations they can perform
seccomp	Which system calls they can invoke
SELinux/AppArmor and other LSM policies	Which resource accesses are permitted

<https://en.wikipedia.org/wiki/Cgroups>

cgroups are responsible for resource management. It makes sense to grant some daemon exclusive access to this functionality to avoid lots of problems.

If your sound card can do hardware mixing, and your Linux device driver supports this feature, then multiple programs can access your sound card at the same time and you hear them all simultaneously!

PulseAudio daemon does software mixing. Without hardware or software mixing, only one program can access the sound card; as a result, you cannot have Audacious AND VLC put out sound at the same time!

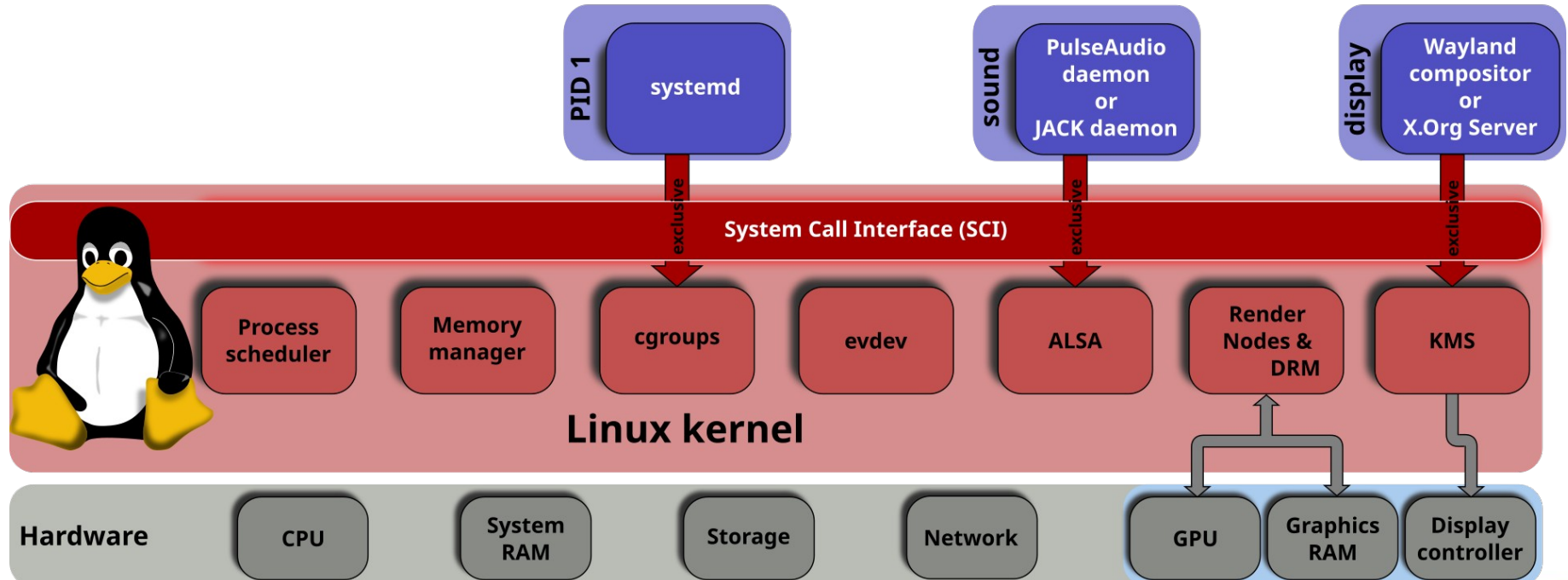
JACK daemon does the same but targets professional audio editors.

DRM manages the GPU.

KMS manages the display controller (CRTC). The display controller usually sits on the die of the GPU and communicates with the monitor, e.g. changes the resolution or the refresh rate.

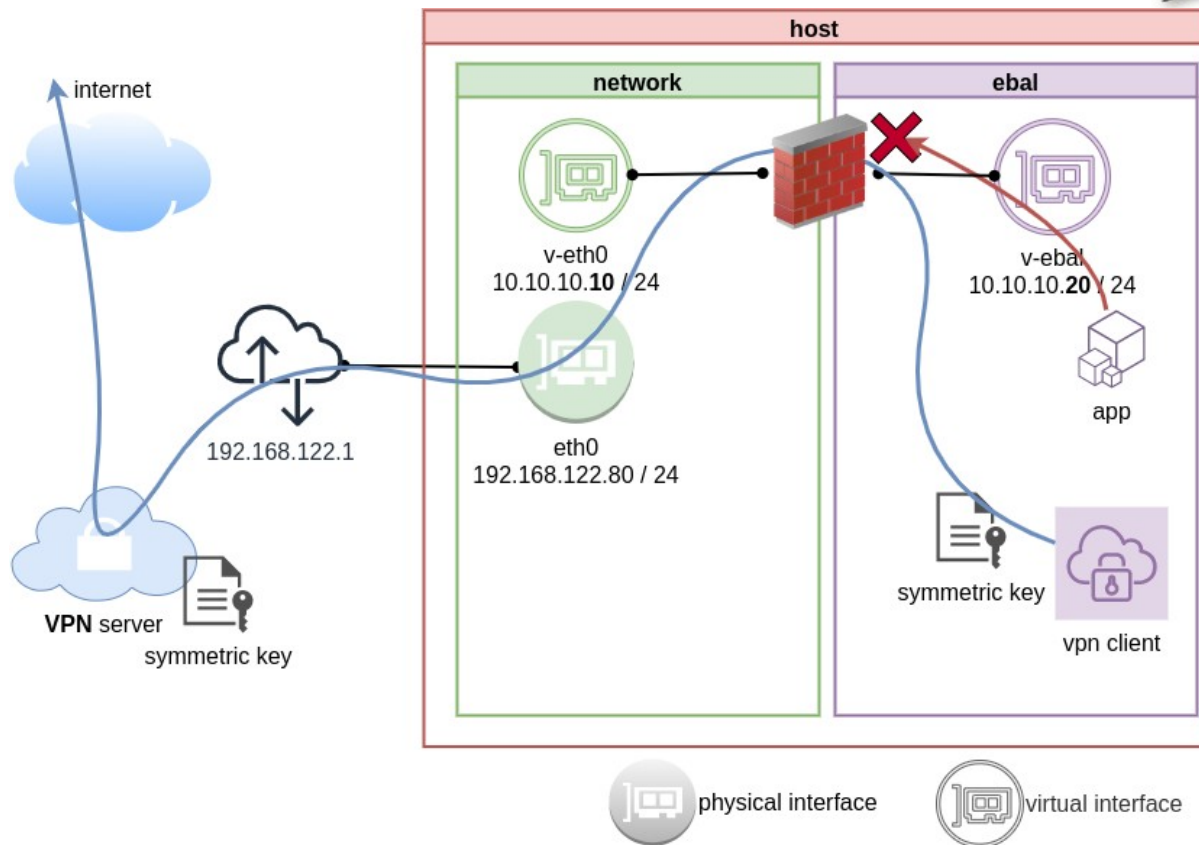
David Herrmann split DRM and KMS, then added "render nodes" to the DRM.

X.Org doesn't need to be root any longer, but it's still wise (technically necessary?) to grant it exclusive access to the KMS.



Namespaces

<https://balaskas.gr/blog/2020/05/16/network-namespaces-part-three/>



https://www.usenix.org/system/files/sec21_slides_tolley.pdf

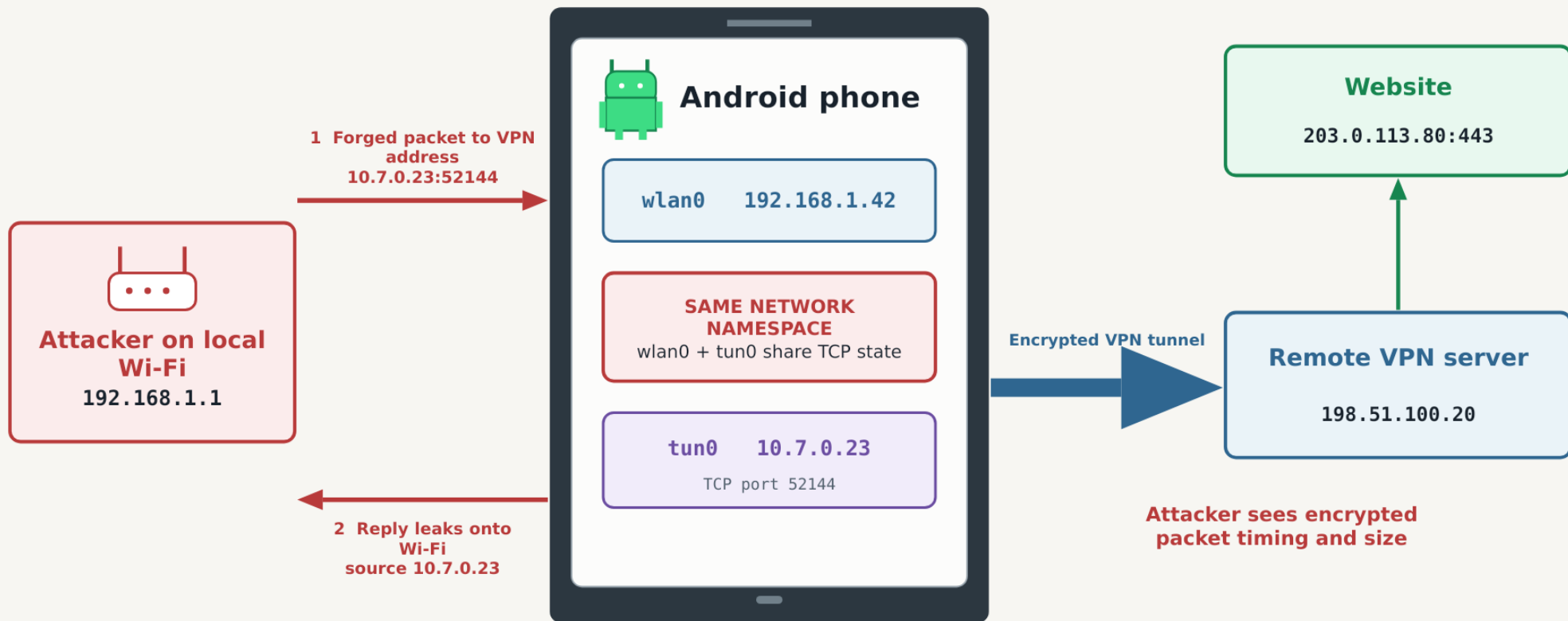
Blind In/On-Path Attacks *and Applications to VPNs*

William J. Tolley^{†‡}, Beau Kujath^{†‡}, Mohammad Taha Khan[§], Narseo
Vallina-Rodriguez^{¶£}, and Jedidiah R. Crandall^{†‡}

Arizona State University[†], Breakpointing Bad[‡],
Washington & Lee University[§],
IMDEA Networks Institute[¶], International Computer Science Institute[£]

Client-side VPN attack: one shared network namespace

Android handles Wi-Fi and VPN traffic in the same network stack.



Result: the attacker identifies an active tunneled connection without decrypting the VPN.

fartbook

Email or Phone

Password

Log In

[Forgot account?](#)



TWiT

@NiTWiTNetwork

Home

About

Photos

Videos

Events

Posts

Community

Email Signup



Like

Share

Suggest Edits

...

Watch Video

Send Message

Photos



Community

[See All](#)

25,350 people like this

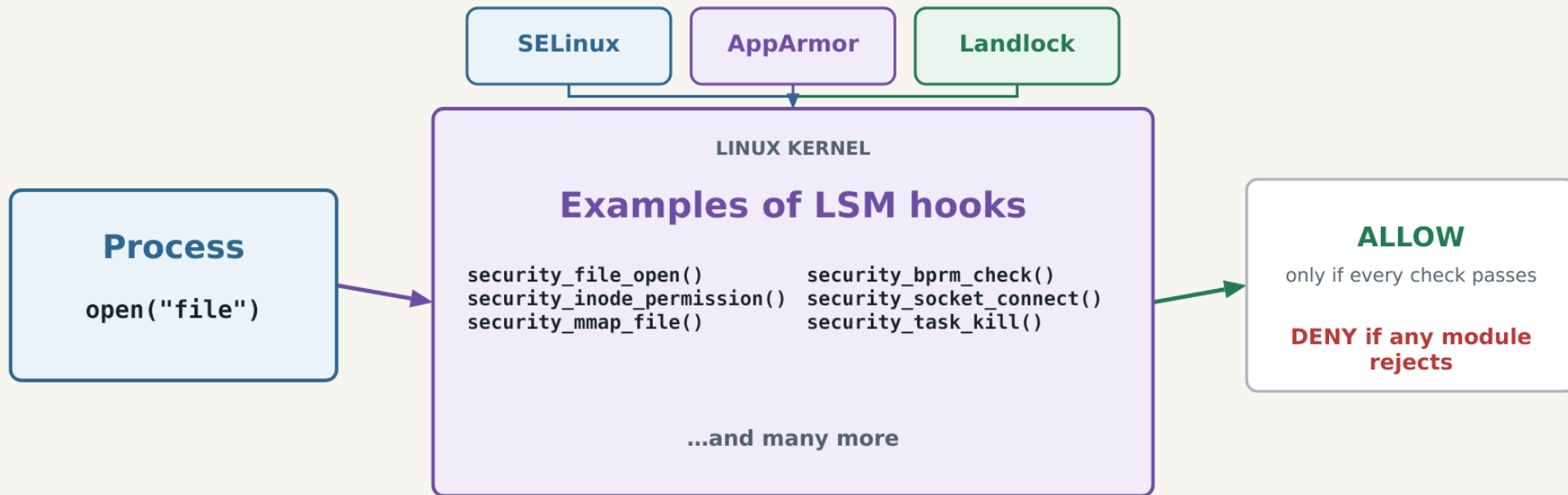
25,804 people follow this

About

[See All](#)

Linux Security Modules (LSM)

Kernel hooks let security modules approve or deny sensitive operations.



LSM is the plug-in framework. SELinux, AppArmor, and Landlock are modules that use it.

Landlock vs. AppArmor vs. SELinux



- Landlock: a process can put restrictions on itself, using the Linux Security Modules (LSM) framework
- AppArmor: An application can use the LSM framework to apply access controls specific to that app
- SELinux: system-wide policies, using the LSM framework

Landlock: Zathura restricts itself

The sandbox applies the policy before opening the PDF.



Can read the PDF



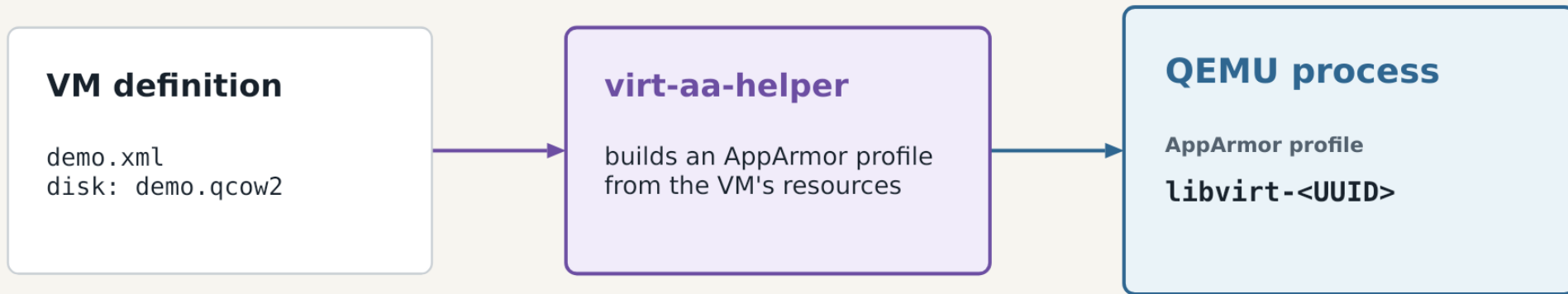
Cannot write, delete, or execute files

Reads of other user-readable files remain possible.

Zathura enables seccomp separately to filter system calls.

AppArmor in Ubuntu 26.04: libvirt and QEMU

Each system VM gets a dynamic profile named from its UUID.



Allowed:

/var/lib/libvirt/images/demo.qcow2



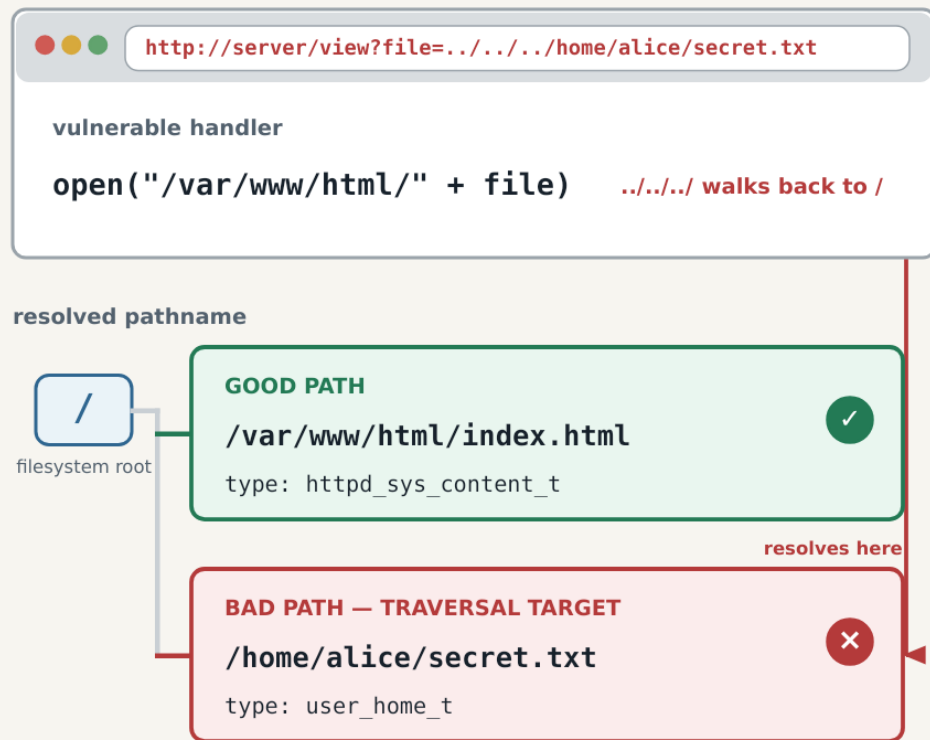
Denied: /home/alice/secret.txt

The profile names host paths and gives this VM only the files it needs.

`/etc/apparmor.d/libvirt/libvirt-<UUID>`

Directory traversal meets SELinux

A vulnerable URL can escape /var/www. SELinux still checks the resolved file's type.



Ubuntu 26.04 SELinux policy

Optional repolicy package

1 Label the web tree

```
apache.fc
/var/www(/.*)?
gen_context(system_u:object_r:httpd_sys_content_t,s0)
```

2 Allow Apache to read web content

```
apache.te
allow httpd_t httpd_ro_content:file
{ map read_file_perms };

httpd_sys_content_t belongs to httpd_ro_content
```

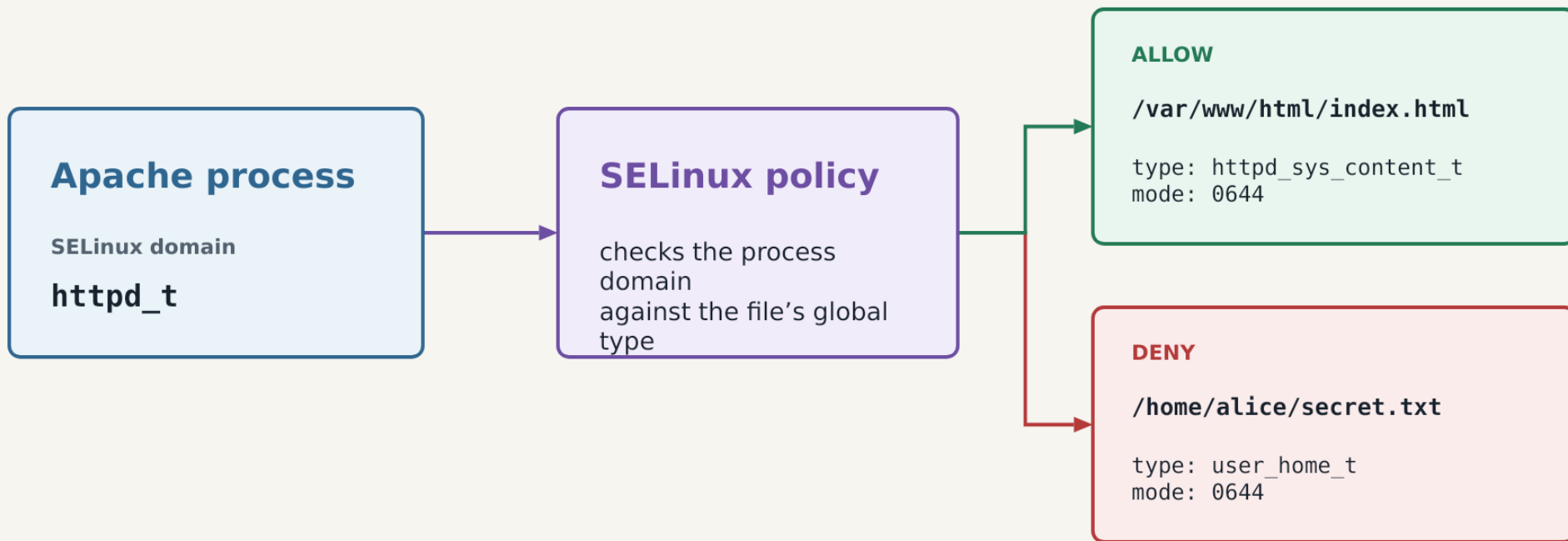
user_home_t does not match that allow rule

No matching permission: DENY

The file type is system-wide. Apache's domain determines what it may read.

SELinux on Ubuntu 26.04: Apache

Ubuntu defaults to AppArmor. Its optional SELinux policy uses system-wide file labels.



The file label is system-wide. Policy decides what each process domain may do with it.

KVM and QEMU

Virtualization gives each guest OS a computer backed by shared hardware.

KVM

CPU and memory

Guest CPUs and memory



KVM in the Linux kernel



Physical CPUs and RAM

Uses CPU virtualization hardware
Controls guest memory mappings

QEMU

Devices and machine model

Guest disk, network, display



QEMU in host user space



Host files, network, window

Provides virtual devices
Can also emulate a CPU in software

Together: QEMU uses KVM to run guest CPUs on the real CPU.



QUBES OS

Separate VMs. One desktop.

[work] xen-notes.txt - Mousepad

File Edit Search View Document Help

Qubes uses Xen to isolate qubes (VMs).
Each qube has its own guest kernel.
Different qubes share one desktop.

Paravirtualization:

The guest knows it is virtualized and
uses hypervisor-aware interfaces.

Qubes normally uses Xen PVH:
hardware-assisted CPU and memory,
with paravirtualized I/O.

[untrusted] Downloads - Mozilla Firefox

about:downloads

Downloads



unknown-attachment.pdf
11.4 MB • Complete



[personal] Documents - Thunar

File Edit View Go Bookmarks Help

/home/user/Documents

Places

- Computer
- user
- Desktop
- Documents**
- Downloads
- Music
- Pictures
- Videos
- Trash



family-notes.txt



budget.ods



FIRECRACKER

Many small cloud workloads

Host editor - firecracker-notes.txt

File Edit Search View Document Help

Firecracker is a small VM monitor.
It replaces QEMU and still uses KVM.
Each microVM boots its own Linux kernel.
KVM runs CPUs. Virtio provides disks and networking.

Built for many small cloud workloads:
fast startup, low overhead, fewer virtual devices.
Control it through a host-side API.

Example: CSE 536 Werewolves.
Run each match in a separate microVM.
Players in one match share its Linux kernel.
The VM boundary helps contain a compromised match.
Start a fresh microVM for each new match.

The jailer sandboxes the Firecracker host process.

Host terminal - match A serial console

Linux guest A

Werewolves match A
Players connected: 14
Day 1: discussion
Voting open

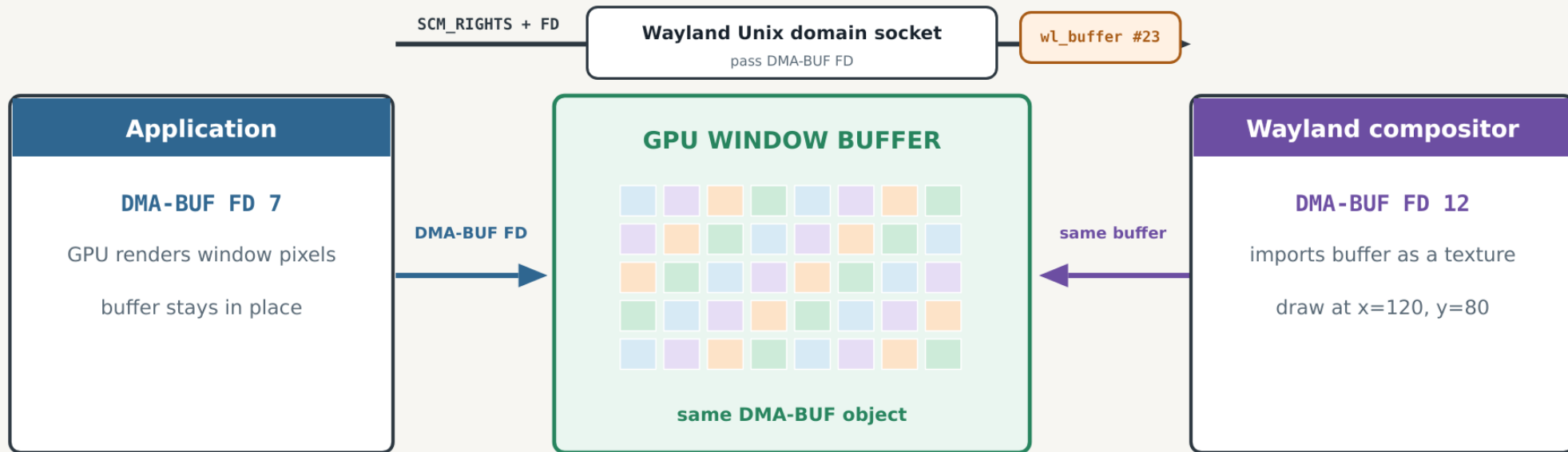
Host terminal - match B serial console

Linux guest B

Werewolves match B
Players connected: 16
Night 2
Waiting for player actions

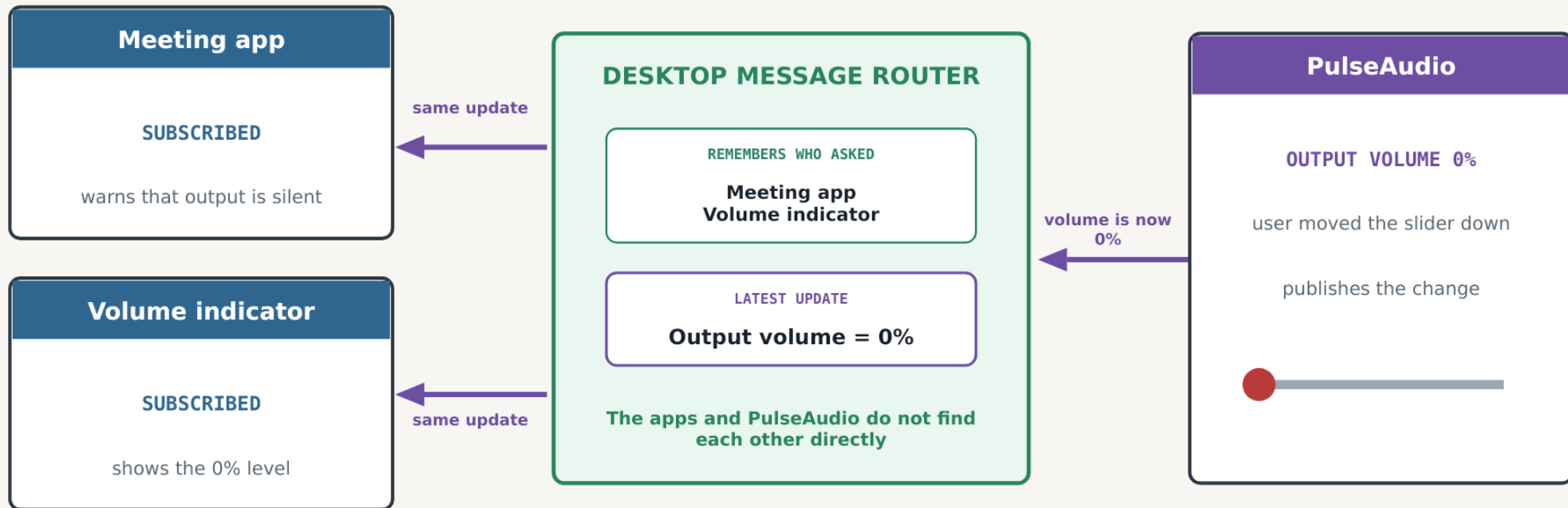
Wayland GPU windows

The compositor moves a window by changing where the GPU draws the same buffer.

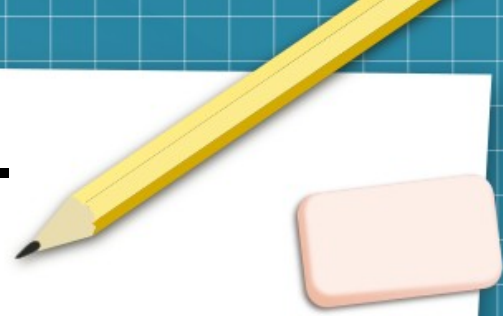


PulseAudio volume notification

Two apps asked for volume changes. PulseAudio reports 0%, and the desktop router updates both.



Other OSes have stuff, too...



- FreeBSD
 - Jails
 - TrustedBSD
 - Capsicum
- Solaris
 - Solaris Zones, Predictive Self-Healing (about robustness, not security), RBAC



This work is licensed under a Creative Commons
Attribution-ShareAlike 3.0 Unported License.
It makes use of the works of Mateus Machado Luna.

