Subsystem Update: seccomp, Yama, and LoadPin

Linux Security Summit NA
August 21, 2019
San Diego, California

Kees (“Case”) Cook
keescook@chromium.org
@kees_cook

https://outflux.net/slides/2019/lss/seccomp.pdf

Agenda

- Background
  - what's a "small" LSM?
- LoadPin
  - purpose, recent development
- Yama
  - purpose, recent development
- seccomp
  - why is seccomp not an LSM?
  - purpose, matching, filters
  - recent development
  - demo

I. Lietuvių Susivažiavimas Vilniuje.

Lapkričio 21—22 (Gruodžio 4—5) d. 1905.

PROGRAMA:

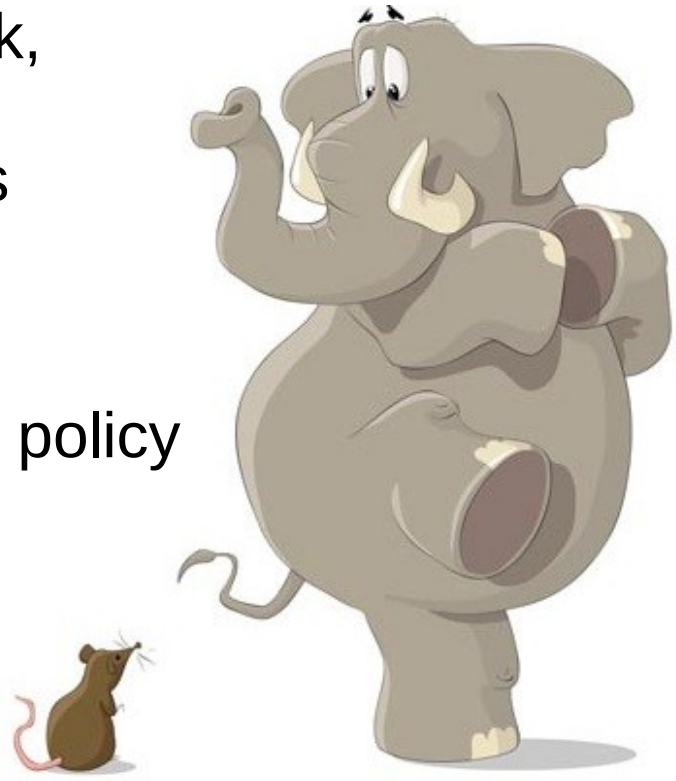
1. *Lietuvystės praeitis ir dabartis.*
2. *Lietuviai, latviai ir įvairios tautos Lietuvoje gyvenančios.*
3. *Caro manifestas nuo Spalinio 17 (30) dienos.*
4. *Lietuvos autonomija ir prikergimas Suvalkų gubernijos prie autonomiškosios Lietuvos.*
5. *Rinkimai į Viešpatystės seimą.*
6. *Lietuvos sodiečiai po rusų valdžia.*
7. *Lietuvos mokyklos.*
8. *Apie įvairius mokesčius.*
9. *Apie žemiečių įstaigas.*
10. *Apie išeivystę (emigraciją).*
11. *Apie kareiviavimą.*

Gali buti įnešami ir kitoki klausimai.

Organizacijos Komitetas.

Background

- Regular LSMs (SELinux, AppArmor, Smack, TOMOYO) have a comprehensive policy language covering a full Mandatory Access Control system
  - also integrity and capabilities
- “small” LSMs have a very narrow (or fixed) policy
  - I maintain LoadPin and Yama
  - there is also SafeSetID and soon lockdown



LoadPin: Overview

- Built with `CONFIG_SECURITY_LOADPIN=y`
- If you think `CONFIG_MODULE_SIG_FORCE=y` is redundant in your environment, LoadPin is for you!
  - Chrome OS uses `dm-verity` to provide a cryptographically verified read-only root filesystem
  - There is no need to sign modules – they just have to come only from the root filesystem
- Also protects other files the kernel reads:
  - kexec images, firmware, security policy, certs, etc

LoadPin: Recent developments

- Thankfully pretty stable (maybe only Chrome OS uses it?)
- v4.20
  - human readable device name in initialization output
  - boot param name changed from “`enabled`” to “`enforce`”

Yama: Overview

NOT
ACTUAL LOGO

- Built with `CONFIG_SECURITY_YAMA=y`
- The first “stacked” LSM (sorry not sorry)
- Narrows scope of ptracing from “same uid” to “ancestor and explicit whitelist”
- Basic goal is to expand the time window needed to steal a user’s credentials after successfully breaking into a machine

Yama: Recent developments

- Also pretty stable (and many distros use it!)
- v5.0
  - fixed a task death RCU race found by syzkaller

seccomp: Overview

NOT ACTUAL LOGO

- Not an LSM. More low-level: it filters system calls
- `no_new_privs` saves the day against setuid issues
- As seen in Chrome, Chrome OS, Android, Docker, systemd

  and Firefox, QEMU, OpenSSH, vsftpd, LXD, Tor, the list goes on…
- Easy to add seccomp to your code!
  - To quickly wrap a program, use `minijail -S policy.txt`
  - To use normal filtering, you want `libseccomp`
  - To do really special things, you'll need to learn BPF
    - actually a subset of classic BPF (not eBPF)

seccomp: Filter matching

- filter can match anything in the seccomp BPF “packet data”:

```
struct seccomp_data {
  int   nr;                    /* System call number           */
  __u32 arch;                  /* AUDIT_ARCH_* <linux/audit.h> */
  __u64 instruction_pointer;   /* CPU instruction pointer      */
  __u64 args[6];               /* Up to 6 system call args     */
};
```

- Can not (yet) read process memory (e.g. filename arg contents) – would be racey

seccomp: Filter results

- `SECCOMP_RET_ALLOW`: continue with syscall
- `SECCOMP_RET_LOG`: emit audit record and continue
- `SECCOMP_RET_TRACE`: generate `PTRACE_EVENT_SECCOMP`
- `SECCOMP_RET_USER_NOTIF`: skip syscall, deliver notification over fd
- `SECCOMP_RET_ERRNO`: skip syscall, return specified `errno`
- `SECCOMP_RET_TRAP`: skip syscall, deliver `SIGSYS` signal
- `SECCOMP_RET_KILL_THREAD`: kill the thread
- `SECCOMP_RET_KILL_PROCESS`: kill the process (thread group)

seccomp: Recent developments

- v5.0: Tycho Andersen added `SECCOMP_RET_USER_NOTIF`
  - Basically `SECCOMP_RET_TRACE` using a file descriptor instead of ptrace
- Behavioral questions
  - should this gain a “okay, continue with syscall” response?
  - should this gain notification toggling?

SECCOMP_RET_USER_NOTIF: Demo

samples/seccomp/user-trap.c

Thoughts?

Kees (“Case”) Cook

keescook@chromium.org
keescook@google.com
kees@outflux.net

@kees_cook

https://outflux.net/slides/2019/lss/seccomp.pdf