Linux Kernel Security Update

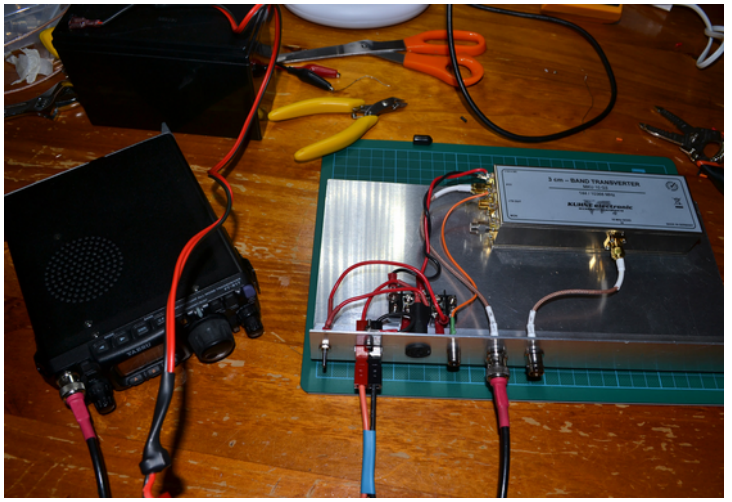
LinuxCon Europe
Berlin, 2016

James Morris
james.l.morris@oracle.com

Introduction

Who am I?

- Kernel security subsystem maintainer
  - Started kernel development w/ FreeS/WAN in 1999
  - which led to Netfilter, SELinux, LSM, Crypto…
  - @xjamesmorris
- Linux since 1993
  - APANA public networking
  - BBS's prior to that
  - Amateur radio (vk2txp)



- Mainline Linux kernel development @ Oracle

Outline

- Overview of Linux kernel security
- Developments in 4.x kernel
- Current and future challenges

Linux Kernel Security Overview

Linux kernel core security model is Discretionary Access Control (DAC)

DAC was inherited from Unix, designed in late **1960s**

“The first fact to face is that UNIX was not developed with security, in any realistic sense, in mind; this fact alone ***guarantees a vast number of holes****.”*

Dennis Ritchie, “On the Security of UNIX”, ***1979***

DAC is insufficient for modern security threats:

DAC does not protect against flawed or malicious code

DAC does not cover all security critical functions

DAC notion of superuser violates user security policy

“It must be recognized that the mere notion of a super-user is a theoretical, and usually practical, blemish on any protection scheme.”

(also from Ritchie 1979)

Linux Kernel Security Extensions

Posix ACLs

Capabilities (privileges)

Audit

seccomp

Namespaces

Netfilter

- IPTables

Cryptography API

- Disk encryption
- IPSec
- Key Management (“keys”)

Linux Security Modules (LSM)

- SELinux
- Smack
- AppArmor

SELinux, Smack, AppArmor provide Mandatory Access Control (MAC)

Platform Security

- TPM, NX, SMEP, SGX, TrustZone etc.

Integrity Management

- Integrity Measurement Architecture (IMA)
  - Extended Verification Module (EVM)

Kernel Self Protection (KSP):

Harden kernel against attack

Kernel Self Protection Project:

Kill classes of bugs vs. individual bugs

Current focus is upstreaming grsec/pax features

Website:

https://kernsec.org/wiki/index.php/Kernel_Self_Protection_Project

Recent Changes

- Linux v4.0 (April 2015) to v4.8 (current)

Capabilities

- Ambient capabilities (v4.3)
  - Allows inheritance of capabilities from non-privileged parent processes.
    - … instead of assigning fs capabilities to binary, which will always run with them.
    - Do not need to give all capabilities to script interpreters.

LSM API

- Generalized security module stacking (v4.2)
  - Simple manual stacking previously allowed
  - Now: any number of smaller LSMs can be stacked on top of a major (“monolithic”) LSM
  - e.g. SELinux + YAMA + Capabilities, but not SELinux + TOMOYO + AppArmor.
- New **LoadPin** module, ensures kernel modules & fimware are loaded from trusted device (dm-verity) (v4.7)

Networking

- CALIPSO IPv6 Labeling (v4.8)
  - RFC 5570
  - Security labels in IP option
  - IPv6 version of CIPSO
  - Usable by label MAC (SELinux, Smack)
  - Verified interop with Solaris TX

AppArmor

- Kernel work focused on AA 3.0 cycle
- *Upcoming (v4.10-v4.11)*
  - *Policy namespaces*
  - *Policy stacking*
  - *Integration with containers*
- See JJ’s talk video from LSS 2016!

SELinux

- Android Binder IPC support (v4.0)
- Full Netlink coverage (v4.1)
- Performance improvements (v4.1)
- Fine grained ioctl coverage (v4.3)
- Export validatetrans decisions to userspace (v4.6)
- Restrict kernel module loading (v4.7)
- CALIPSO support (v4.8)
- *Upcoming: Overlayfs support (v4.9)*

Smack

- Netfilter secmark support (v4.0)
- Allow unconfined label in bringup mode (v4.1)
- Obtain security context of keys (v4.1)
- Multiple label MAC bypass via onlycap (v4.2)
- IPv6 host labeling (v4.3)
- Limited dynamic process labels (v4.4)
- Process-based permission checking for sockets (v4.5)

Integrity Subsystem

- Integration of TPM 2.0 authorization policies with kernel keys, allow hash algorithm selection (v4.5)
- EVM support for x.509 kernel certificates (v4.5)
- Measurement & appraisal of IMA policy (v4.6)
- Support for kernexec image & initramfs (v4.6)
- Support for mknotat syscall (v4.7)
- Per-rule specification of PCRs (v4.8)
- *Upcoming: extend measurment to command line, BPF etc., fine grained signatures, directory measurement, namespacing.*

Platform Security

- TPM 2.0 chip support (v4.0)
- Intel Memory Protection Keys (v4.6)

- *Upcoming:*
  - *Sparc: SSM (Silicon Secured Memory)*
  - *AMD: SME, SEV (memory encryption)*
  - *Intel: CET (Control-flow Enforcement Technology)*

Audit

- Add support for auditing by executable file, rather than just PID (v4.3)
- Add ioctl device and command info to LSM audit data (v4.3)
- Add tty field to Login event (v4.7)

Seccomp

- ptrace options for suspend/resume (v4.3)
- powerpc and tile support (v4.3)
- Dump seccomp filters via ptrace (v4.4)
- um and parisc support (v4.5)
- Remove 2-phase API (v4.8)
- ptrace before seccomp (v4.8)
- *Maybe upcoming: deep argument inspection*

Keys

- Support for kernel module signing (v4.3)
  - Explicit file for x.509 trusted keys
  - Sign modules with external key
- Support for TPM 2.0 (v4.5)
- Userspace access to DH computation using stored keys (v4.7)
- Encrypt big keys saved to shm (v4.7)
- Key blacklisting and rejection (v4.7)
- Runtime addition of secondary system key (v4.7)
- *Upcoming: key revocation*

Crypto API Users

- ext4 filesystem encryption (v4.1)
- Kernel module signing (v4.3)
- MACsec/IEEE 802.1AE (v4.6)
- Migrate ext4 to vfs crypto API (v4.8)
- *Upcoming: btrfs encryption*

Kernel Self Protection

- Kernel Address Sanitizer (KASan) (v4.0)
  - SLAB support (v4.6)
- Always enable RODATA checking (v4.6)
- KASLR for ARM64 (v4.6), MIPS (v4.7)
- Page zero-poisoning (v4.6)
- X86 execute-only memory (v4.6)
- SLAB freelist randomization (v4.7)
- BPF JIT constant blinding (v4.7)

KSP (cont.)

- Freelist randomization for SLUB (v4.8)
- KASLR:
  - Full physical memory on x86_64 (v4.8)
  - Kernel memory base on x86_64 (v4.8)
- gcc plugin infrastructure (v4.8)
- Hardened usercopy (v4.8)

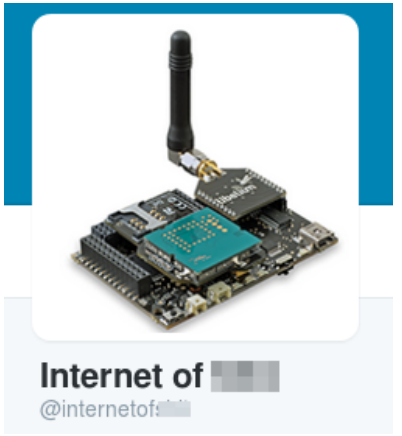
KSP (cont.)

- Predictions for v4.9 from Kees Cook
  - latent_entropy gcc plugin
  - vmalloc stack on x86
  - List hardening
  - PAN emulation for arm64

- For more detail:
  - https://outflux.net/blog/ (Kees' blog)

Future Challenges

- IoT
- KSP arms race
  - Need more original research in mainline!
- Evolving threat models
- Security architecture vs. features



Resources

- Linux Security Module mailing list
  - http://vger.kernel.org/vger-lists.html#linux-security-module
- Linux Security Summit (Aug 2016, Toronto)
  - http://events.linuxfoundation.org/events/linux-security-summit/program/slides
- Kernel Self Protection Project
  - http://kernsec.org/wiki/index.php/Kernel_Self_Protection_Project
- LWN Security
  - http://lwn.net/Security