Elena Reshetova

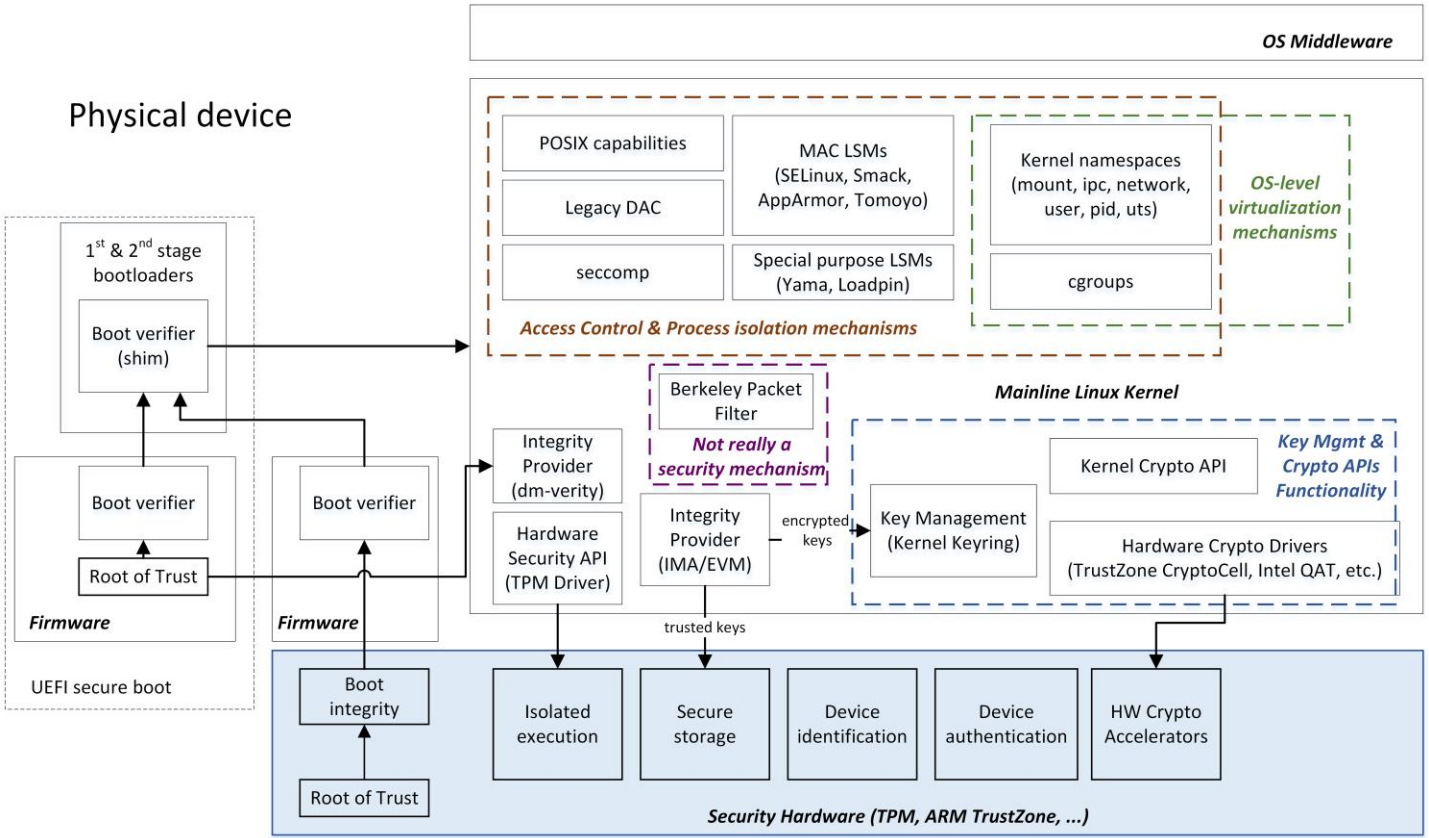
# Towards (better!) Linux kernel security: the journey of past 10 years

# About myself

- Security researcher and architect
- Ms. Degree in Information Security from SUAI University, St. Petersburg
- Doctoral degree in Information security from Aalto University, Espoo, Finland
- Intel Open Source Technology center from 2011
- Linux mobile and embeeded platform security
- Recently: Linux kernel hardening and cryptography

# Goal of this talk

# Linux kernel security mechanisms



# What this talk is not covering

- Non-mainline solutions
- Key management (Keyring)
- Linux Crypto API
- Crypto HW and their drivers
- Platform Security Features
- Netfilter
- Classical virtualization
- ...

# Linux Security Summit: 2010 – 2019

- 2010 – Boston, 1 day, 20-30 participants, 11 talks, 1 panel
  - Brad Spengler, “Linux Security in 10 Years”
  - Kees Cook, “Widely Used But Out-Of-Tree”
  - Mimi Zohar, “Using EVM to protect security extended attributes”
  - 6 access control-related talks
- 2019 – LSS NA (San Diego) & LSS EU (Lyon), 5 days, 420+ people, 42 talks, 4 tutorials, BoF & lightening sessions
  - Kernel hardening is actually happening in mainline
  - Plain AC (MACs) is not a major focus (working relatively well already)
  - Much more diversity in topics

# Main tendencies for the past 10 years

Security does matter!

- Performance vs. Security
- Real world deployment in wide range of scenarios and use cases
- More variety and more complexity

# Use cases and areas that drive Linux kernel security

# Mobile and handheld devices

- Very active in 2010-2015
- Mobile security architectures for various OSes
  - Symbian, Maemo/MeeGo, Android, iOS, …
- Higher reliability requirements & legislations
- Usable security as a new challenge
- Many elements and components are similar
  - mandatory access control, per-application permissions, secure/authenticated boot, …
  - different design decisions

# Virtualization, clouds & distributed workloads

OS-level vs. traditional virtualization

OS-virtualization goes back to 2000

- Virtualize kernel resources vs. HW
- A “container”: application or system
- Checkpoint and restart
- Many OS-specific implementations
  - Free BSD jails, Linux V Server, Solaris Zones, Open VZ,  Linux Containers (LxC), Cells, …
  - Linux mainline converges to “kernel namespaces”

# Embedded and IoT

- Many different small OSes
  - Build using “OS building tool”: Open Embedded/Yocto, Buildroot, …
- No big resources for security architecture development
- Initially very limited in HW and SW and fully unprotected
- Real time OSes
  - NuttX, Zephyr, FreeRTOS, …
  - Its own set of challenges

# Old railway wagons

# Access Control

LSMs

- Gradual improvement
- MAC for mobile OSes
  - SELinux for Android
  - Smack for MeeGo/Tizen
  - AppArmor for Ubuntu Touch
- Stacking
  - Casey Schaufler, “Multiple Concurrent Security Models? Really?” – 2013
  - Well on its way by now
- Namespacing
  - Smack, AppArmor, …

LSS NA 2019 Keynote: Retrospective: 26 Years of Flexible MAC - Stephen Smalley, National Security Agency

# Integrity & Secure/Authenticated/Verified boot

- Essential component for many systems
  - High end to low (and very low) end
- **Integrity Measurement Architecture (IMA)/EVM**
  - Policy-based filesystem measuring
  - Root of Trust is in TPM or Security HW
- **dm-verity**
  - Transparent & lightweight integrity checking of block devices
  - Filesystem is read-only
  - Root of Trust is in RO FW
- **fs-verity**
  - “dm-verity” per file-basis
  - Integration with IMA

- Deployments on a scale
- Immutable data protection for mobile device
- Runtime mechanisms for ensuring immutability in containers

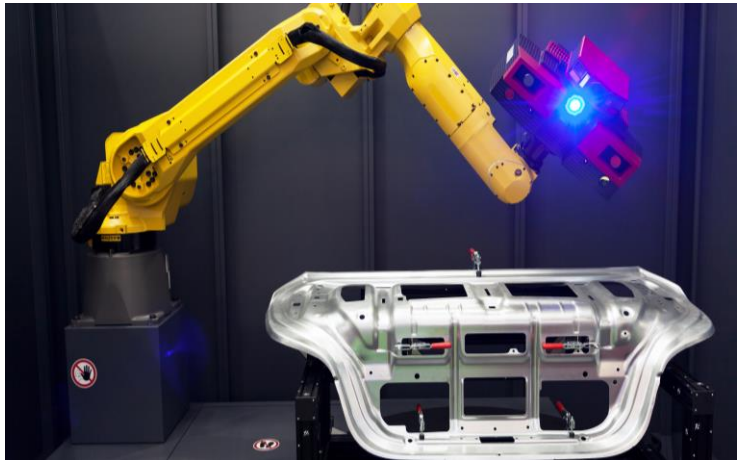
**Open challenges:**

- Integrity of mutable data
- Policy namespacing
- Smooth integration with Security HW root of trusts (non-TPM based)

# Newcomers

# Narrow focused mechanisms

- Minor LSMs
  - Narrow scope (fixed) MAC restrictions
  - Yama, LoadPin
- Seccomp
  - Programmatic kernel attack surface reduction



From Yama submission patch in 2010: https://lwn.net/Articles/393012/

```
@@ -0,0 +1,72 @@
+config SECURITY_YAMA
+       bool "Yama NAC Support"
+       depends on SECURITY
+       select SECURITYFS
+       select SECURITY_PATH
+       default n
+       help
+         This selects Yama, the NAKed Access Control system which
+         provides additional global security settings above regular
+         Linux discretionary access controls.
```

A thread on link restrictions on lkml in 2005 (see 800179c9 for info):

> *Of course we can't always rely on programmers to get it right, so the*
> *idea here is to make sure we ask broken code to behave nicely, and stab*
> *it in the face if it doesn't. Please try to examine this in that scope.*

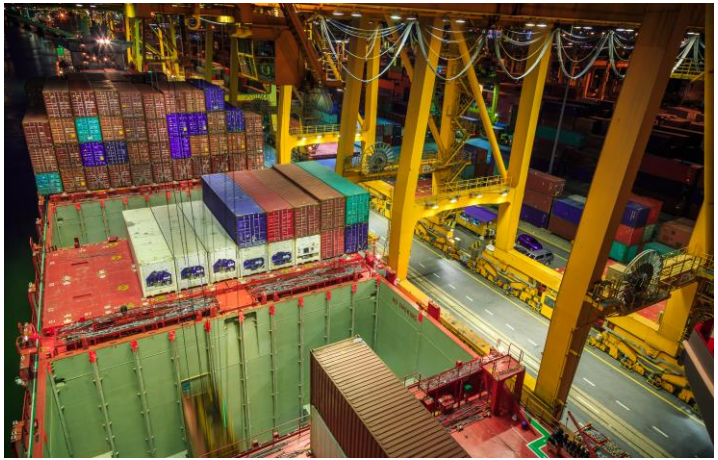
It's fine for hardened distro. But still inappropriate for mainline.

# Namespaces, containers & capabilities

- Initially not driven by security requirements (like chroot())
- Linux basic kernel namespaces: mount, pid, user, networking, ipc, ...
- Others are desired: time, device, keyring, LSM, IMA policy, ... seccomp policy namespaces?
- Initially - strong opposition:
  - Involved complexity & high risk, controversial
  - Not mature technology & not enough knowledge
- Now – slow future direction:
  - Many LSMs experiment with namespaces: Smack, AppArmor, ..
  - IMA policy & keyring namespacing proposal exists
- Ambient capabilities and capabilities per namespace
  - Overlap and escalation problems are not solved

Capabilities challenge: “False Boundaries and Arbitrary Code Execution”
https://forums.grsecurity.net/viewtopic.php?f=7&t=2522

***Meta question: ”What is a container from a kernel perspective?”***



# Kernel hardening – KSPP project

Aims to eliminate:

- Classes of bugs
- Methods of exploitation

“Slow but steady” motto

A lot of good work is coming out

- Hardening features
- Found bugs
- Mind-set changes

## Specific TODO Items

Besides the general work outlined above, there are number of specific tasks that have either been asked about frequently or are otherwise in need some time and attention:

### Kernel items

- Split thread_info off of kernel stack (Done: x86, arm64, s390, powerpc. Needed on arm and others?)
- Move kernel stack to vmap area (Done: x86, s390. Needed on arm, arm64, powerpc and others?)
- Implement kernel relocation and KASLR for ARM
- Make CONFIG_STRICT_KERNEL_RWX and CONFIG_STRICT_MODULE_RWX mandatory (done for arm64 and x86, other archs still need it)
- Further restriction of perf_event_open (e.g. perf_event_paranoid=3)
- Extend HARDENED_USERCOPY to split user-facing kmalloc()s and in-kernel kmalloc()
- split short-lived kmalloc()s from long-lived kmalloc()s
- split user-size-controlled kmalloc()s from regular kmalloc()s
- protect ARM vector table as fixed-location kernel target
- disable kuser helpers on arm
- add constant-blinding tests to lib/test_bpf.c
- rename CONFIG_DEBUG_LIST better and default=y
- create defconfig "make" target for by-default hardened Kconfigs
- expand use of __ro_after_init, especially in arch/arm64
- restrict autoloading of kernel modules (like GRKERNSEC_MODHARDEN) (Timgad LSM)

...

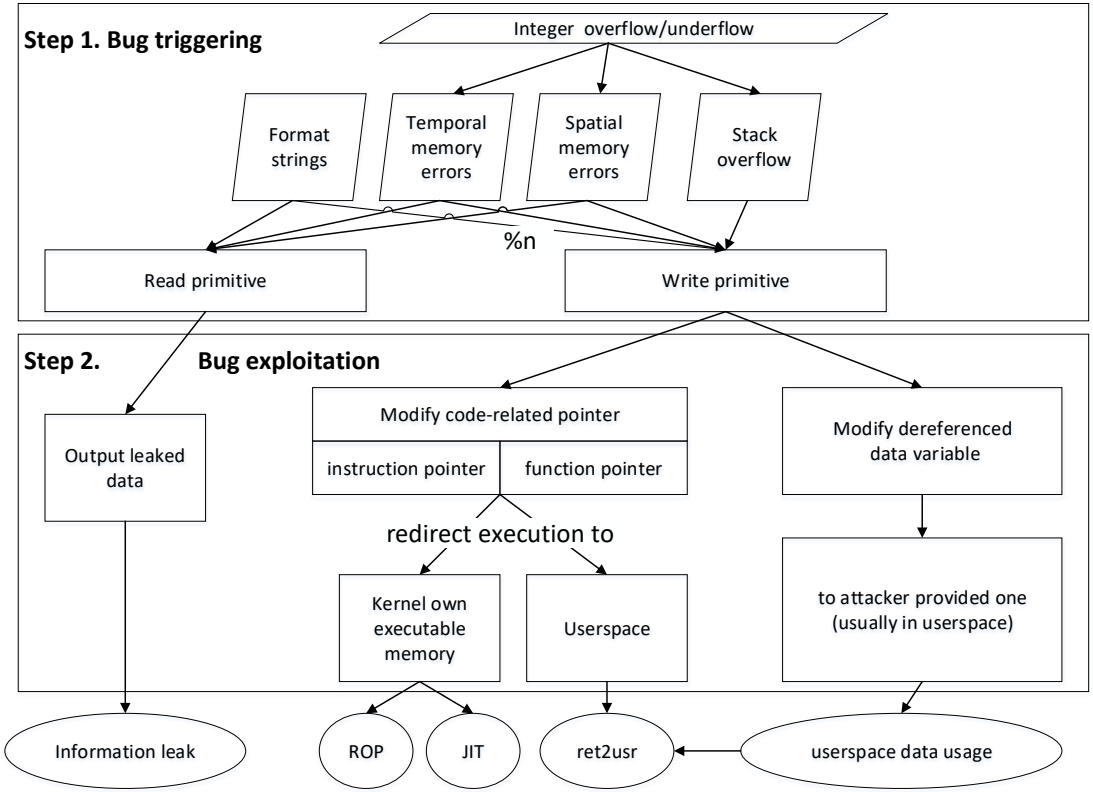


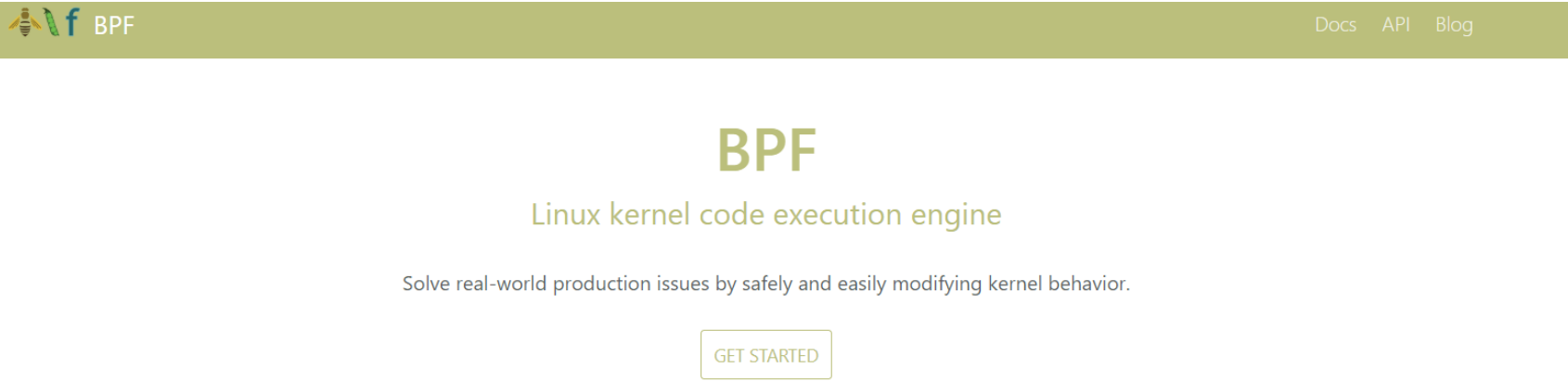
https://kernsec.org/wiki/index.php/Kernel_Self_Protection_Project

# Kernel exploitation stages – KSPP focus areas

- Different KSPP features target different exploitation stages and different primitives

- Detailed mapping on KSPP features by Alexander Popov: github.com/a13xp0p0v/linux-kernel-defence-map




BPF
Docs API Blog
BPF
Linux kernel code execution engine
Solve real-world production issues by safely and easily modifying kernel behavior.
GET STARTED
Extend the Linux kernel
Extend Linux kernel behavior for a variety of purposes, like load balancing, container networking, kernel tracing, monitoring, and security.
Run code in the kernel
Run user-space code in the kernel to solve production issues where user-space solutions alone aren' t enough.
Tools built on BPF
Use open source tools built on BPF that are now in production, including tools for resource control, traffic control, monitoring, troubleshooting and more.

# BPF – Berkeley Packet Filter

- In-kernel VM for unprivileged execution from userspace
  - generic mechanism
- Original paper for BSD packet filter from 1992
- Very wide applicability
- Recently very interesting for both attackers and security researchers

**LSFMM 2019 gains a BPF track**

The call for proposals for the 2019 Linux Storage, Filesystem, and Memory-Management Summit has been updated with an important addition: this year's event (April 30 to May 2, San Juan, Puerto Rico) will include a BPF track. The submission deadline has been extended to February 22 to allow BPF developers to put together their proposals.

**Linux Plumbers Conference 2019**
**BPF Microconference**

A BPF Microconference will be featured at this year's Linux Plumbers Conference (

**Linux Kernel Developers' bpfconf 2019**

# BPF security observations

BPF motto :
safely and easily modifying kernel behavior.

- Cons: Breaks standard security assumptions about the kernel
  - Linux Kernel is (important) part of TCB
  - Now this TCB has (quite arbitrary) execution possibility with code from OUTSIDE of TCB
  - Execution is limited (Language limitations, BPF Verifier, …), but …
  - Powerful attack surface
- Pros: Is a useful mechanism that can be used for Security
  - Landlock LSM – use BPF to create tailored security sandboxes
  - KRSI LSM – use BPF to create dynamic MAC policies
  - Cilium - use BPF to do API-aware network security filtering
  - …

# Outlook

# Hopes and expectations for future

People's hopes might differ, here is one:

- LSM stacking, later LSM namespaces
- IMA policy & Keyring namespacing
- More confidence in and more use of BPF for security
- Hardened kernel with each release
- Be more proactive

# References

- Kernel security project page: https://kernsec.org/wiki/index.php/Projects
- Overview of the Linux Kernel Security Subsystem, James Morris, Microsoft
  - https://www.youtube.com/watch?v=L7KHvKRfTzc
- Linux Security Summits: https://kernsec.org/wiki/index.php/Events
- Kernel Self Protection Project: https://kernsec.org/wiki/index.php/Kernel_Self_Protection_Project
- BPF and XDP Reference Guide: https://cilium.readthedocs.io/en/latest/bpf/

**Upcoming**: Linux Security Summit Europe, Lyon, France, 31 Oct – 1 November

https://events.linuxfoundation.org/events/linux-security-summit-europe-2019/

INTEL OPEN SOURCE TECHNOLOGY CENTER | 01.org

elena.reshetova@intel.com
elena.reshetova@gmail.com