# BPF to bridge Cloud and IoT Linux Security

## Linux-IoT based devices

Djalal Harouni @tixxdz

August 18-19th, 2021

# BPF to bridge Cloud and IoT Linux Security

Agenda:

- Motivation Linux-IoT / Embedded Linux
- Linux security mechanisms
- BPF cloud security
- BPF for Linux-IoT security

Djalal Harouni @tixxdz

August 18-19th, 2021

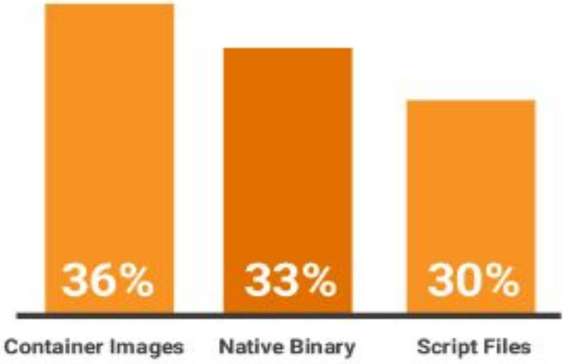
# Motivation: Linux-IoT / Embedded Linux

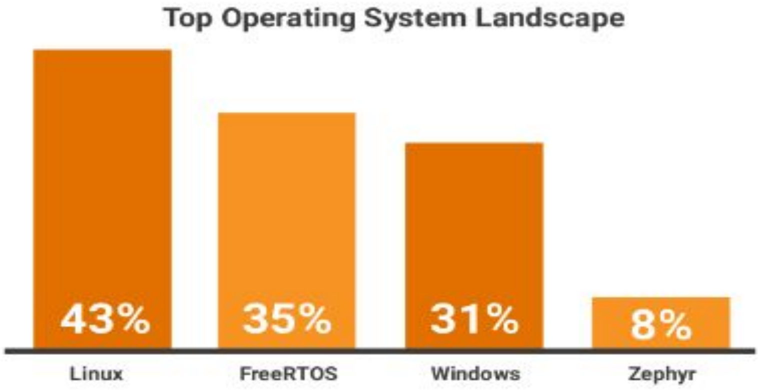
Smart connected hardware everywhere

Devices are more powerful

Devices run modern software stacks

Top Edge Computing Artifacts for IoT Solutions 2020





Eclipse 2020 IoT Developer Survey Key Findings [1]

# Linux security mechanisms

1. Auditing / Monitoring
2. Mitigation

1. Auditing / Monitoring
   - Audit framework (incompatibilities and performance issues)
   - perf
   - **BPF based tracing, etc**

# Linux security mechanisms

2. Metigation
    - Builtin: easy to deploy / handled by applications
        - namespaces (mnt), cgroups, etc
        - Seccomp, Yama LSM, LoadPin LSM, Lockdown LSM, etc
    - Builtin: hard to deploy
        - MAC LSMs hard with complex rules.
            - Not adapted to dynamic environment (k8s)
            - Hard to deploy on Linux-IoT, needs proper engineering.
    - Out of tree: easy to configure
        - grsecurity patches
    - New builtin:
        - **BPF LSM [2],** landlock

# BPF cloud security

eBPF use cases:

- Container runtime security
- Auditing and monitoring
- Performance tracing
- Network security and observability
- load-balancing (k8s, services)
- Forensics analysis
- etc

# BPF for Linux-IoT security

IoT attack signals:

- File system / storage operations: mounting filesystems or external storage
- File system modification : arbitrary file access, hiding files, etc.
- Execution of in memory ELF binaries from network (memfd, etc)
- Access to special kernel files, device files, etc
- Attaching devices: usb, etc
- Loading kernel modules, hiding modules, Loading eBPF programs
- Arbitrary network connections
- Resource exhaustion, network data consumption, logins, brute force attacks, etc

# BPF for Linux-IoT security

Mitigations:

- BPF LSM to filter filesystem operations:
  - Mount new filesystems or remount to change mount flags.
  - Mount kernel virtual filesystems or filter access to special files.
  - But for files modification better with: **multiple mounts + mnt namespace, etc.**
- BPF LSM to block execution of memory ELF binaries or any other suspcious binary, etc.
- BPF Kprobes+LSM for tracing and blocking resource exhaustion, logins, brute force attacks.
- BPF LSM to block attaching devices like usb, loading modules, other BPF programs.
- BPF for IP accounting, network security, data consumption, L7 network flow inspection, etc
- etc

# BPF for Linux-IoT security

BPF for Linux-IoT advantages:

On devices:

- Combine BPF auditing / tracing and mitigation mechanisms to do realtime protection.
- Minimal performance impact and cannot crach the host.
- Low maintenance burden if small BPF programs with stable helpers.
- Avoids subsystems potholes related to cgroups, namespaces, containers and sandboxes aspects.
- Avoids the tedious journey of upstreaming security changes.
- Possibility to have security metigations without a full blown policy rules.
- No LSM conflicts
  - LSM stacking (v5.1): “Nearly 20 years of missing LSM-based innovations because this functionality wasn’t available” grsecurity - 10 Years of Linux Security [3].

Cloud:

- Ability to have observability platforms to detect suspicious activities on devices, flag devices, etc.

# BPF for Linux-IoT security

BPF for Linux-IoT why ?

- More embedded / Linux-IoT compatibility with libbpf, libbpf-go, etc
- BTF (BPF Type Format) is the metadata format which encodes the debug info related to BPF program/map, as structure offsets are built in BTF sections.
- CO-RE (Compile Once, Run Everywhere) portable BPF. Allows compiled BPF bytecode to be relocatable and portable between different kernel versions, and removes the need to have Clang/LLVM runtime being deployed to target machines.

This allows to save storage space by not having to install LLVM, Clang and kernel header dependencies. [4] [5]

# BPF for Linux-IoT security

bpflock - eBPF driven security for locking Linux machines

https://github.com/linux-lock/bpflock

# References:

- [1] https://iot.eclipse.org/community/resources/iot-surveys/assets/iot-developer-survey-2020.pdf
- [2] “Kernel Runtime Security Instrumentation” https://lwn.net/Articles/798918/
- [3] “grsecurity - 10 Years of Linux Security” https://grsecurity.net/10_years_of_linux_security.pdf
- [4] https://facebookmicrosites.github.io/bpf/blog/2020/02/19/bpf-portability-and-co-re.html
- [5] https://www.brendangregg.com/blog/2020-11-04/bpf-co-re-btf-libbpf.html

# Thank you!

## Special thanks to eBPF Summit organizers