New proposal:
Securing IoT Devices with Diversification

Prof. Ville Leppänen,
Software Engineering & Software Security

May 2018 S²ERC SHOWCASE, Ball State University
Motivation

**Situation**
There are a few different execution platforms

*Software monoculture*

platforms are using **identical internal structure**!

**Result**
Single malware works on **millions of machines**!

**What to do**
Break the monoculture *(how?)*
Proactive solution: Diversification

**Change the inner workings of a system** to be based on changed internal interfaces (changing done after development at deployment phase)

**Car analogy**

The same feeling, the same performance, but engine room parts and their connections are different + engine hood welded shut
Diversification continued

- Goal: Deployed systems are made unique from each other
  - System = software, platform, libraries, OS, network protocols, …
  - Unique = Internal structures changed, no changes to user
- Some resemblance to cryptography
  - A secret key is used…
  - …but there is no decryption in the traditional sense
Diversifying a system

Your code

Your application

Diversification Method

A diversified secret interface definition between the platform and application

Diversified release

Application platform

Interfaces with

Application - platform interface

Diversified software platforms
MALWARE on a diversified system

Your application interfaces with the Application platform. Injected code can use already known interfaces for compromising the system.

Diversified release and Diversified software platforms. Injected code or foreign application cannot call platform features on a diversified system.
For internet of things

Advantages
- Low or no performance/storage penalty
- No hardware changes necessary
- Multilayer defense: Can be used with other security solutions
- A bug discovered later: attacker can’t use the security hole …

Add diversification as additional defense against mass exploitation

Reading material
Proposal

- Build **tools** for **diversifying internal interfaces** in **microkernel-based IoT OS** (e.g. RIOT, but something else also possible)
- Tools apply to **applications** and the **OS** (propagation) in **binary** form
- Phases: (1) Designing diversification solution for e.g. system call layer and selected higher API layers (depending on the selected OS); (2) implementation for the lowest layer; (3) demonstration of the implementation; (4) implementations for selected higher layers + demonstration
Benefits

- When IoT devices are diversified having unique device-wise software, then a single malware no longer works in millions of machines (as just happened with Mirai for IoT-like devices)
- Prevent large scale attacks against IoT devices
- Tools for diversifying your IoT devices!
- Not yet available for microkernel-based IoT’s.
Previously applied diversification by us for:

- Linux OS kernel, system call layer
- Library layer (and transitive propagation to all ELF files)
- SQL layer
- Web layer (JavaScript, HTML, ajax-style)
- Command shell layer
- Rasperry Pi (no micro-kernel based structure; code + mem layout)
Sources of diversification related research (source: an SLR)
Questions?  (Some of our related works)