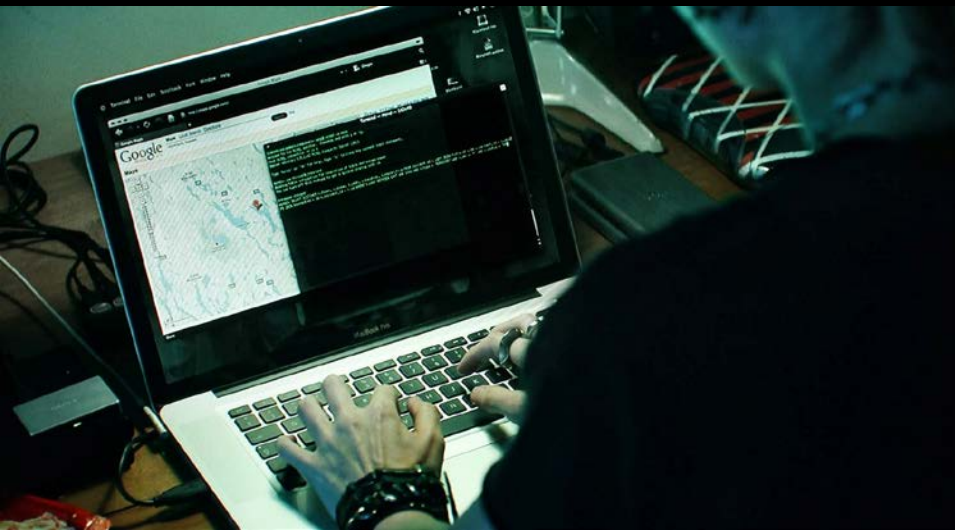




**The
Industrial Internet of
Sitting Ducks**

Jos Wetzels





Jos Wetzels



Midnight Blue

Independent
Security Researcher

ICS
IoT
Automotive
Medical
Access Controls
Networking & Firewalls

**UNIVERSITY
OF TWENTE.**

(Former)
Security Researcher

Critical Infrastructure
Embedded BinSec
HIDS / NIDS



@s4mvertaka



www.midnightbluelabs.com
[samvertaka.github.io](https://github.com/samvertaka)

THE INTERNET OF THINGS

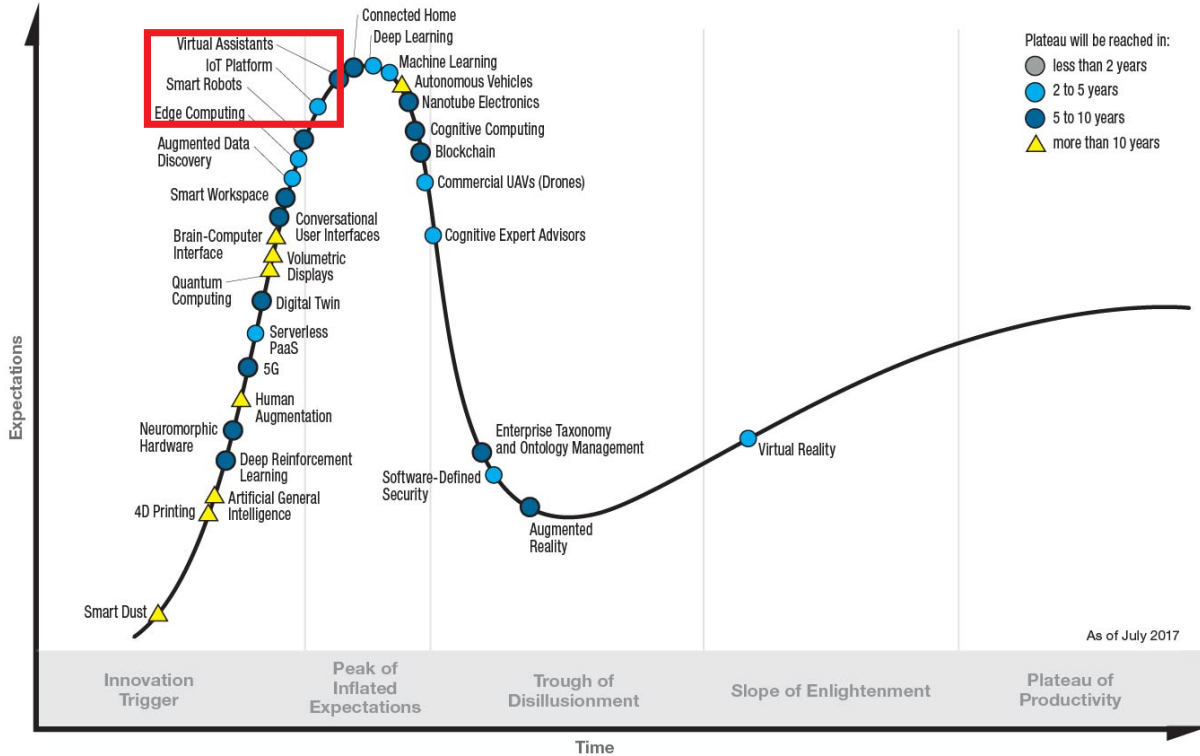
AN EXPLOSION

Gartner Hype Cycle for Emerging Technologies, 2017

2020
50.1 BILLION
Taking population predictions into account, there will be about 5.6 devices per human on the planet.



BILLIONS OF DEVICES



1002
1,000,001
About the equivalent of the pop of San Jose.



'80 '92 '94

'18 '20

YEAR

“

We are seeing a market failure for cybersecurity and privacy. (...) Currently there is no basic level, no level zero defined for the security and privacy of connected and smart devices.

- ENISA / Infineon / NXP / STM

Pizza Hut made shoes that will order pizza



Parents warned over exploding fidget spinners powered by Bluetooth

You Can Only Wash Google And Levi's New \$350 'Connected' Jacket Ten Times

Moxie: Showerhead with wireless speaker



★★★★★ Scdragon · 8 days ago

Wifi does not connect

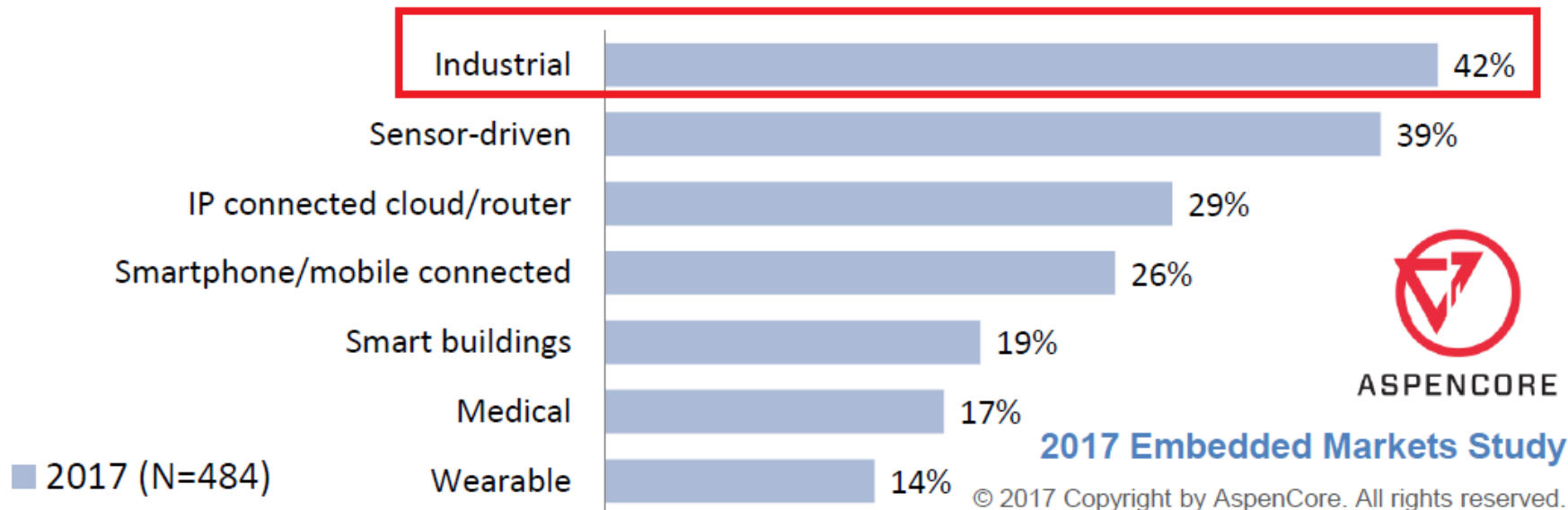
Wifi connectivity does not work. Stuck in setup mode.



Maid: Smart Microwave Oven



If you are developing Internet of Things (IoT) applications, please indicate the type of application.

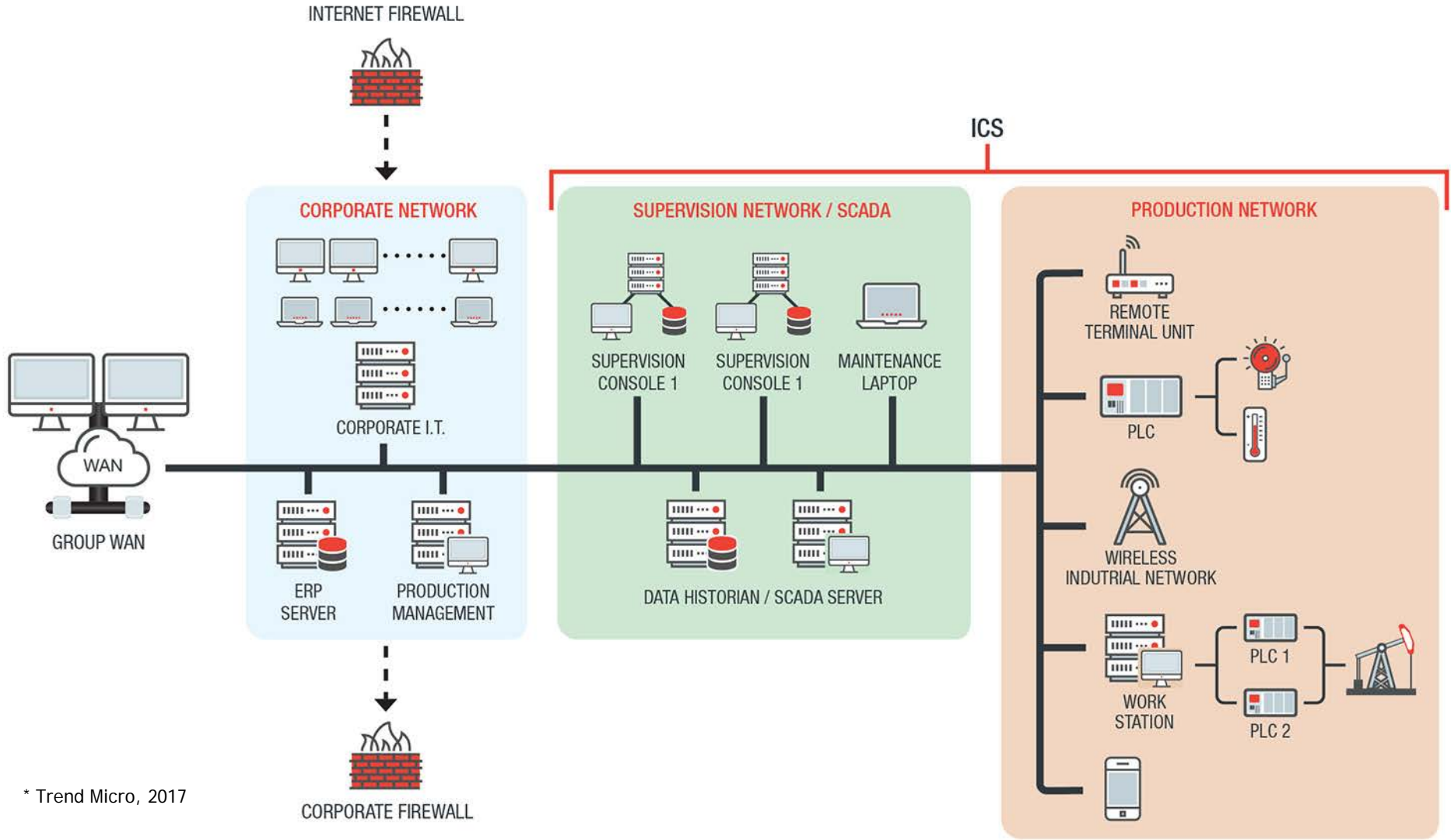


Smart Factories

The image depicts a futuristic industrial setting, likely a smart factory. In the foreground, a hand holds a smartphone displaying a digital dashboard with various data visualizations, including a line graph, a bar chart, and several circular icons. The dashboard is labeled 'STAGE 1' at the top. The background shows a large, modern factory interior with a complex network of white structural beams and machinery. A worker in a blue uniform is visible in the distance, interacting with a piece of equipment. The scene is overlaid with glowing blue digital lines and nodes, suggesting a networked or data-driven environment. The overall color palette is dominated by blues and greys, giving it a high-tech, industrial feel.



Critical Infrastructure



PLCs



RTUs

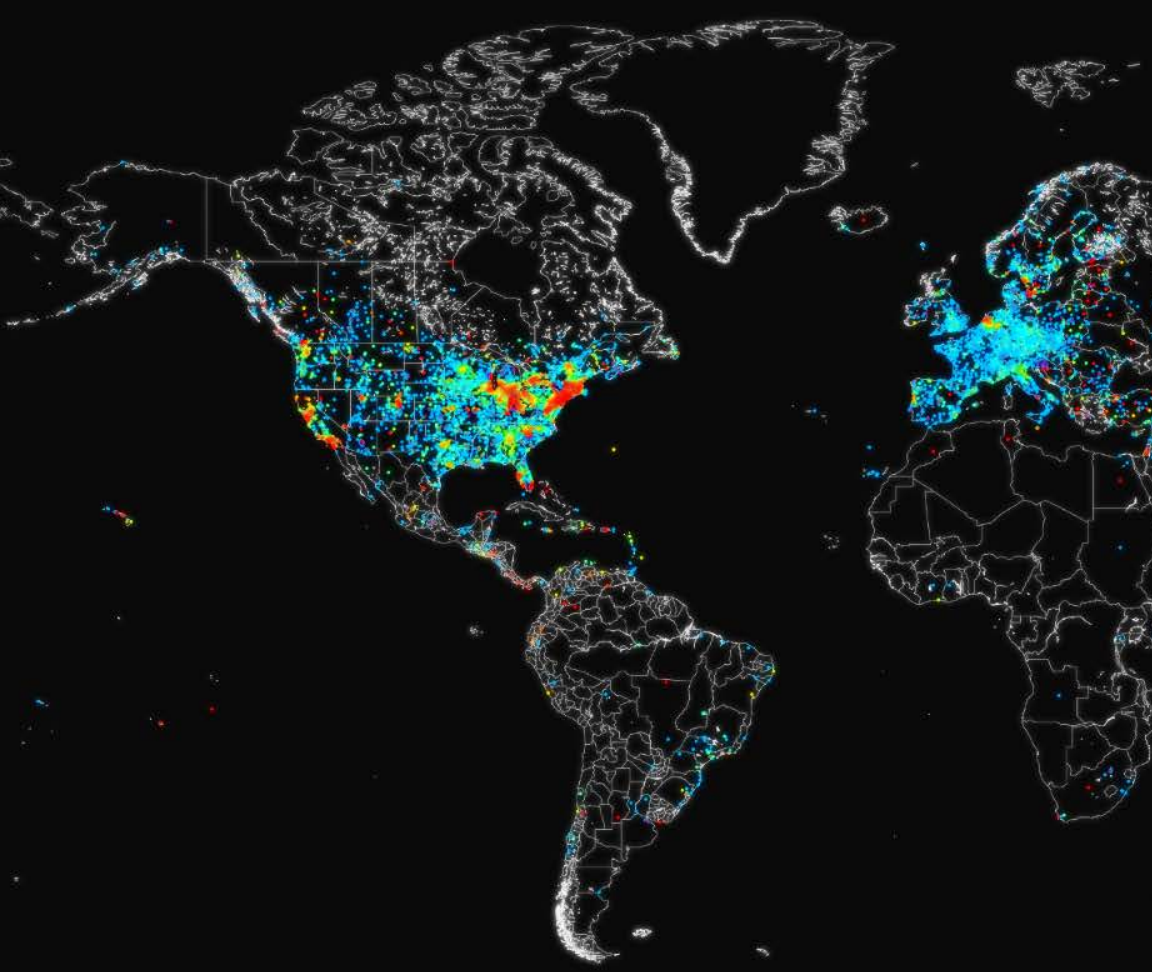


Gateways, Modems



Sensors, Actuators





envac Station 5 Main Active Alarm Auto Ready Seq 0 SE 0 AV 0 DV 0 H 3 L 5 D pr 3

Frequency ref: 5.0Hz

HV

E1: 0.0Hz
E2: 0.0Hz
E3: 0.0Hz
E4: 0.0Hz

Ext. pipe pres [kPa]: 0.2

Frac 1

Valve Status:
DV 16: 1 L LEV
DV 22: 1 H LEV
DV 23: 1 L LEV
DV 24: 1 L LEV
DV 30: 1 H LEV
DV 32: 1 H LEV
DV 33: 1 H LEV
DV 36: 1 H LEV
DV 37: 1 H LEV
DV 38: 1 H LEV
DV 38: 1 L LEV
DV 39: 2 H LEV
DV 42: 1 H LEV
DV 45: 1 L LEV

Filter Pressure [Pa]: 1 [Pa]

System Events:
15-12-11 00:07:33 033 Stop terminal
15-12-11 00:04:18 272 Low air speed AV 39
15-12-11 00:14:18 015 Any low air speed AV
15-12-11 00:03:08 258 Start terminal

Login Service Start / Stop Settings Valve status Logout Disconnect Alarm Tx Alarm reset

ABB Melk control unit PRODAPT MID

Visualization Operations Reports Faults Events Measurements Master data System Help

Overview - all furnaces

Furnace energy counter: 33131070 kWh	Twin Power - Circuit	Furnace energy counter: 37548756 kWh
Melting energy counter: 25237814 kWh		Melting energy counter: 28981748 kWh
Stirring energy counter: 71177823 kWh	Auto Off	Stirring energy counter: 8013638 kWh
Furnace connected to: DCU 2	Furnace On	Furnace connected to: DCU 1

Power: 501 kW	1 State: Active Operation mode: Melting Material: ST Current: 1909 A Frequency: 56 Hz	2 State: Active Operation mode: Keeping warm Material: ST Current: 0 A Frequency: 0 Hz	Power: 0 kW
Weight: 22629 kg	Duration/Remain.: 5:38:41:50 Energy set/act.: 0kWh / 19756kWh sp Energy set/act.: 0kWh/kWh / 2746.1kWh/kWh Weight set/act.: 27100kg / 22629kg Temp. set/act.: 1360°C / 1300°C	Duration/Remain.: 5:38:45:39 Energy set/act.: 0kWh / 76513kWh sp Energy set/act.: 0kWh/kWh / 29578kWh/kWh Weight set/act.: 24775kg / 7226kg Temp. set/act.: 1360°C / 669°C	Weight: 7226 kg
Temperature: 1300 °C			Temperature: 669 °C

2015-07-30 00:47:50 Information: PLSrv - PIC confirmed operation mode manual

2 Furnace off 30.07.2015 02:20:13

ABP INDUCTION

F1 All Furnaces	F3 Operation	F5 End	F7 Power	F9 Treatment	F11
F2 Furnace 2	F4 Heat	F6 Melting	F8 Faults	F10 Furnace	F12

* <https://icsmap.shodan.io/>

A photograph of a laboratory or industrial facility. In the center, several men in white lab coats are walking through a narrow aisle. On either side of the aisle are tall, cylindrical metal structures composed of many stacked, horizontal rings. The floor is marked with blue circular labels containing numbers like 16, 17, and 18. The background is filled with complex machinery, pipes, and electrical equipment. The word "Stuxnet" is overlaid in large white text across the middle of the image.

Stuxnet



Sandworm

The Economics of Ransomware: How SCADA/ICS Changes the Equation



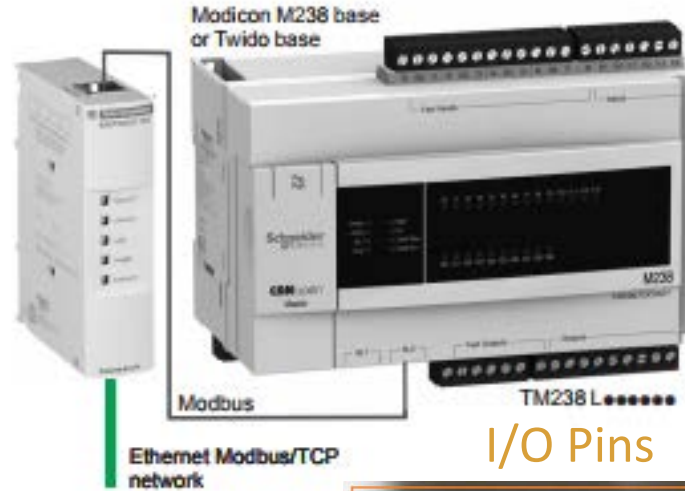
Researchers Create PoC Ransomware That Targets ICS/SCADA Systems

Take Down: Hackers Looking to Shut Down Factories for Pay



=

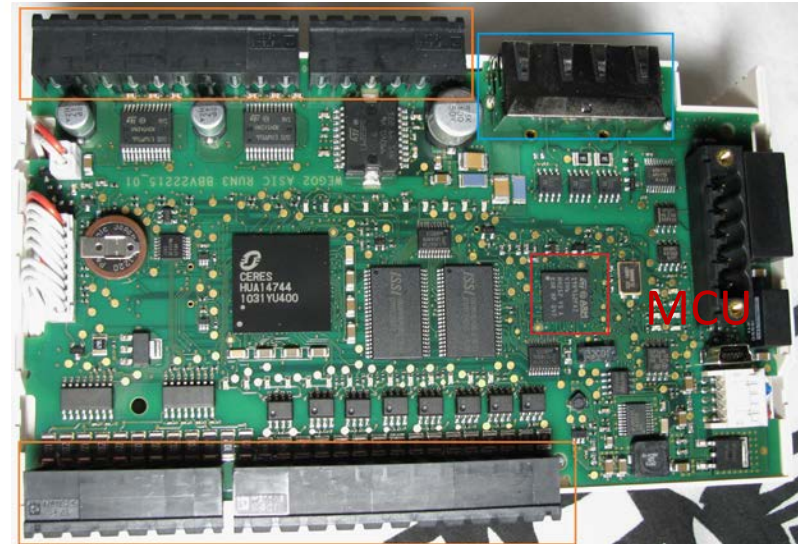




* <https://www.eevblog.com>, 2014

I/O Pins

Serial Link



PLC

Firmware

Applications
(eg. Web, FTP, Telnet, ..)

Operating System
(usually small RTOS)

MCU / SoC

Programmable Logic
(eg. IEC 61131-3)

Logic Handler
(MCU/(C)PLD/
FPGA/ASIC)



Typical ICS Security #1: Opto 22 OPTEMU-SNR-DR2*

- Energy Monitoring & Control Device
- Managed over Ethernet
FTP, SNMP, OptoMMP
(unauthenticated), PAC Control
(unauthenticated)
- Use OptoMMP to
disable IP filtering, enable FTP, get
FTP credentials
- Upload firmware & reflash over FTP
(no firmware signing)

Typical ICS Security #2: Modicon Quantum PLC*



- Large PLC for process applications
- FTP with hardcoded backdoor
Read/Write Access to configuration,
firmware, passwords, etc.
- Telnet with hardcoded backdoor
Is actually a C interpreter...
- Unauthenticated Modbus Extension
Start/Stop PLC
Overwrite programmable logic
Etc.

Open Secret: ICS Security Sucks



- Unauthenticated Plaintext Protocols
- Unauthenticated firmware & logic uploads
- Default Backdoor Passwords
- Absent or Infrequent Patching
- Etc.



(Some) Reasons Why

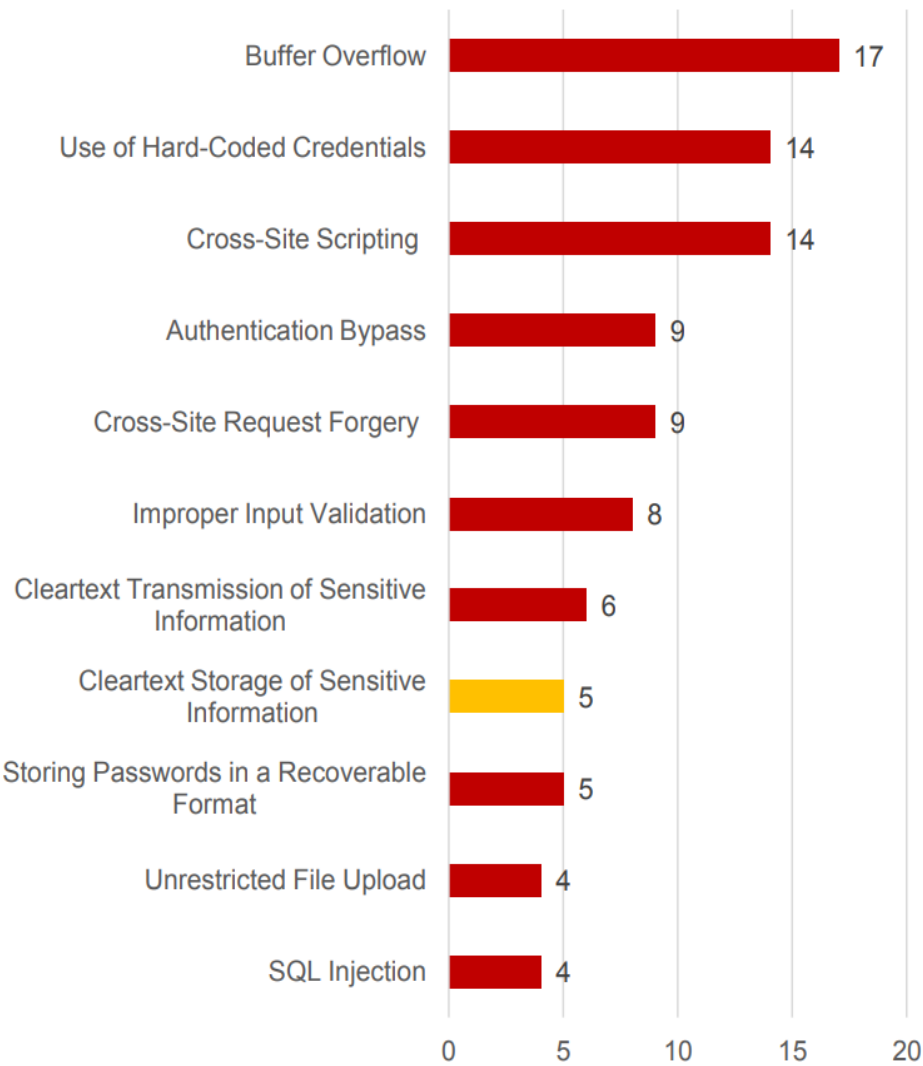
- **Insecure by Design**
Not designed for 'open' networks
Security not considered
- **Device Lifespan**
10+ years

Age-old designs

Ceased vendor support
- **Legacy / Backwards Compatibility**
Adhere to old, insecure standards

The image shows two industrial workers in a refinery or chemical plant. They are wearing dark blue work jackets, white hard hats, and safety glasses. One worker is holding a clipboard and a pen, and the other is looking at it. In the background, there are tall industrial towers, pipes, and a clear blue sky with some light clouds. A semi-transparent dark grey rectangle is overlaid on the image, containing the text.

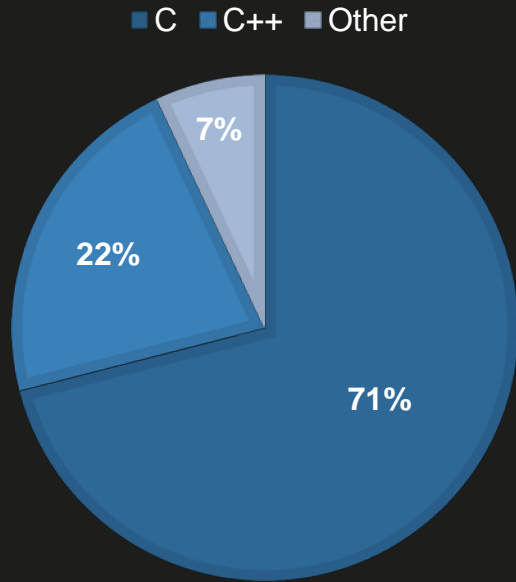
Let's say we fix all this...
What's next?



Memory
Corruption is a
Big Deal™ in
Embedded

* 2016 ICS Vulnerabilities Statistics, Kaspersky

PRIMARY EMBEDDED PROGRAMMING LANGUAGE



Unsafe Languages Are Here To Stay

- Ideally use safe languages
Java, Go, Erlang, Rust
- But unsafe continues to dominate
C, C++



Exploit Mitigations

General Purpose Exploitation Has Been Getting Harder



2010

VS



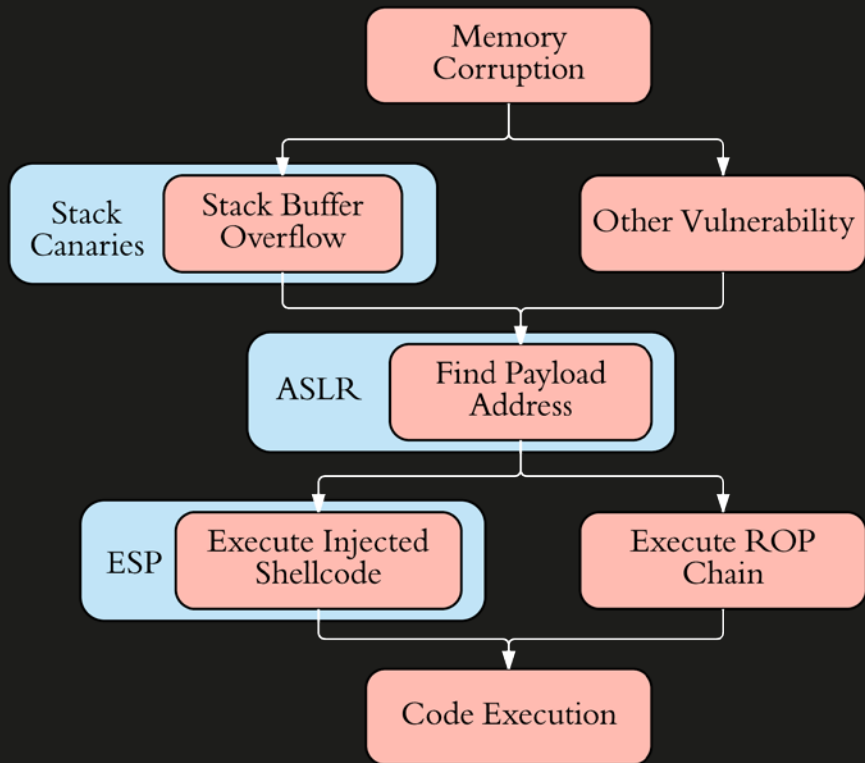
2016

A black and white photograph of a rural landscape. In the foreground, a dirt road leads towards the horizon. A large, round hay bale sits on the road. To the left, a utility pole stands with power lines stretching across the sky. The background shows a flat, open field under a cloudy sky. The text "What About Embedded?" is overlaid in the center in a bold, white font.

What About Embedded?



Quantitative Analysis

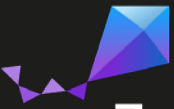


Minimum Mitigation Baseline

- ESP / DEP / NX / W^X Non-exec. data memory
- ASLR Address Space Layout Randomization
- Stack Canaries / SSP Stack buffer overflow protection



OpenWrt
Wireless Freedom



Zephyr™

RIOT ARMmbed

Windows Embedded

QNX



iOS



NUCLEUS



μC/OS

NetBSD

WIND RIVER
PULSAR LINUX



Green Hills
SOFTWARE



TinyOS

VxWorks

Operating System Selection

- Selected 45 Popular Embedded Oses
- High-end, Low-end, Linux/Windows/BSD-based, proprietary, etc.
- Evaluated Support For Mitigation Baseline

Optimistic Assesment

Mitigation Support

— ESP — ASLR — Stack Canaries



What's Going On Here?



Usual Embedded Suspects



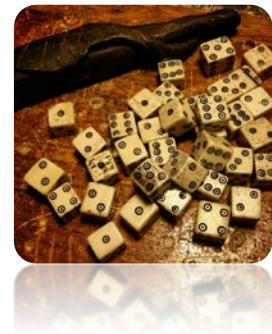
Resource
Constraints



Hardware
Limitations



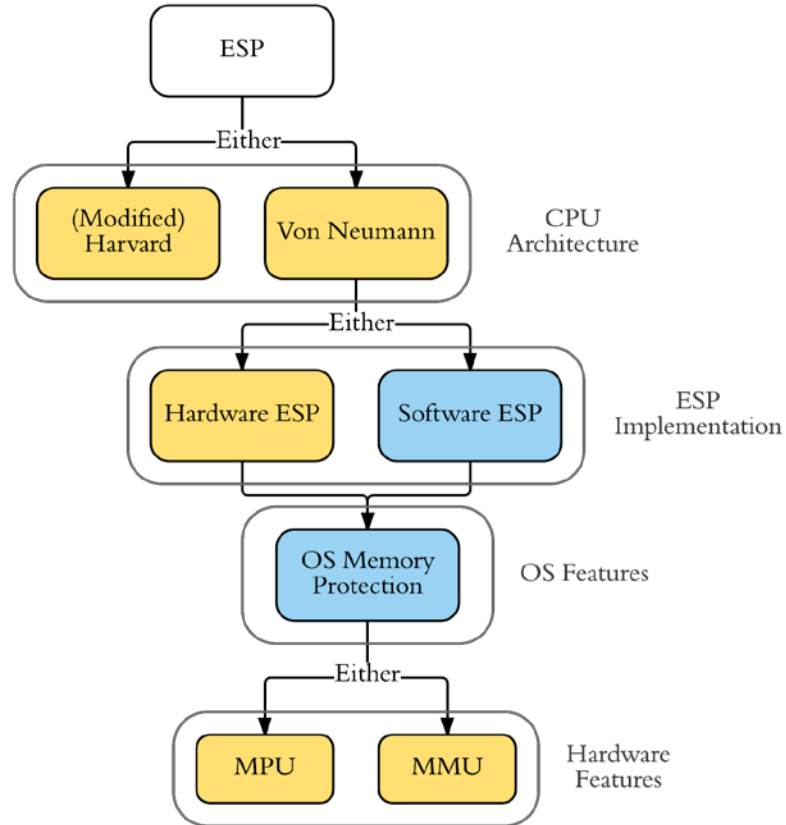
Cost
Sensitivity

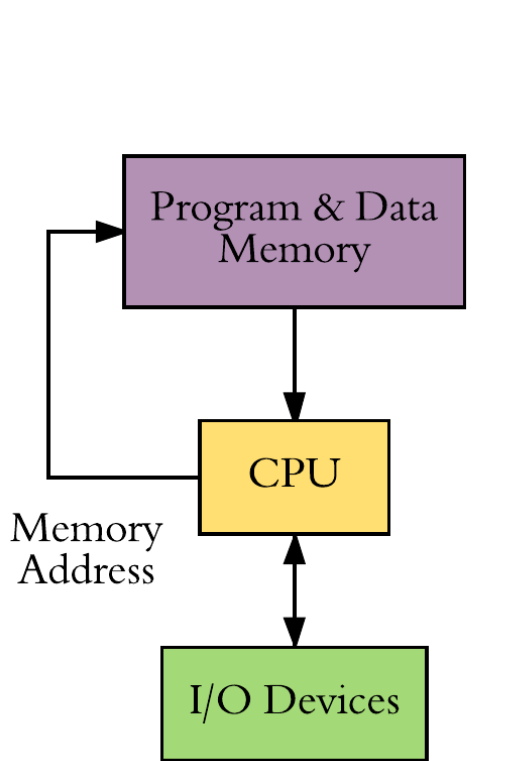


Random Number
Generator Issues*

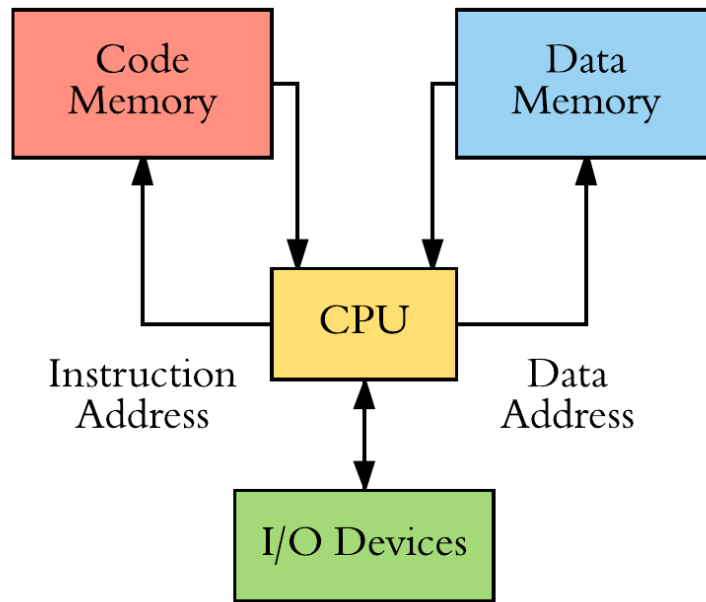
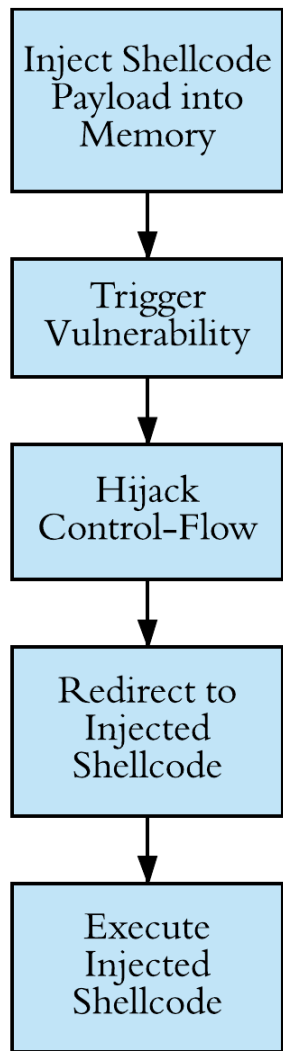
* 'Wheel of Fortune' - Jos Wetzels, 33C3, 2016

Hardware & Software Dependencies

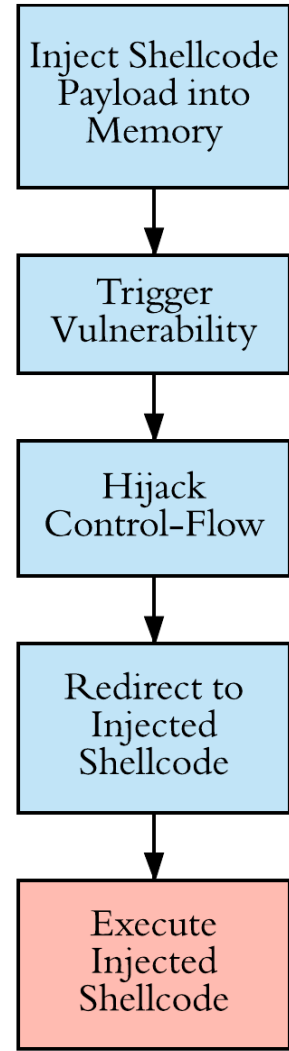




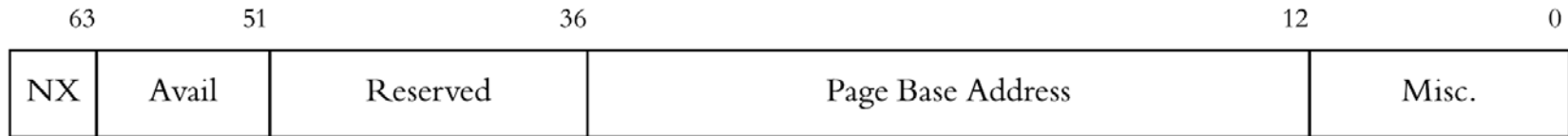
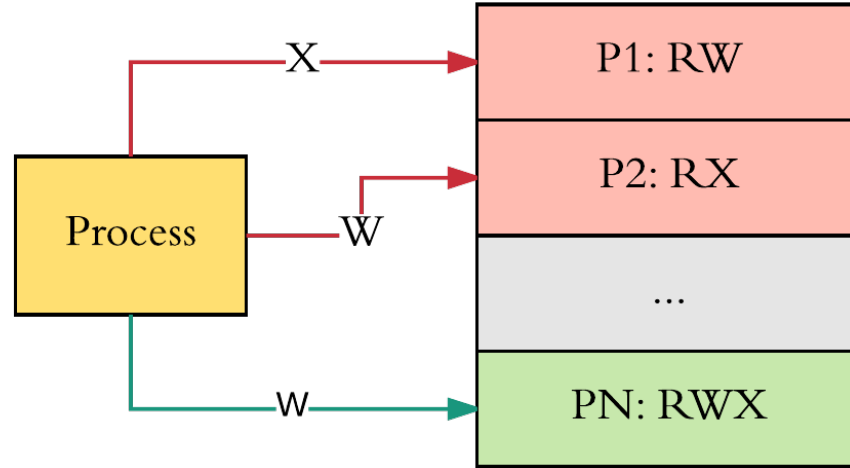
Von Neumann Architecture (VNA)



Harvard Architecture (HA)

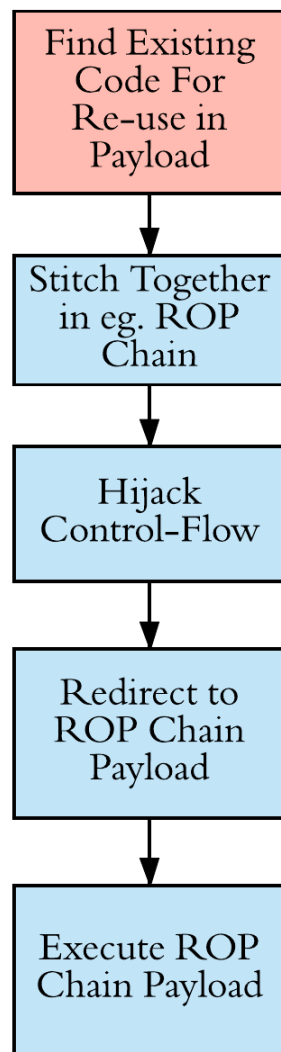
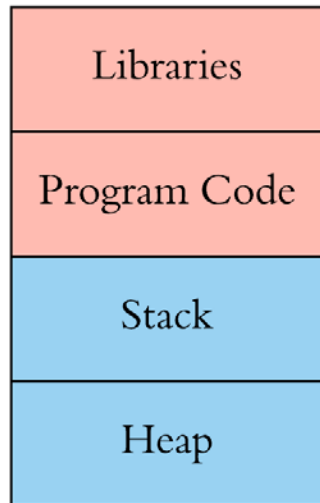
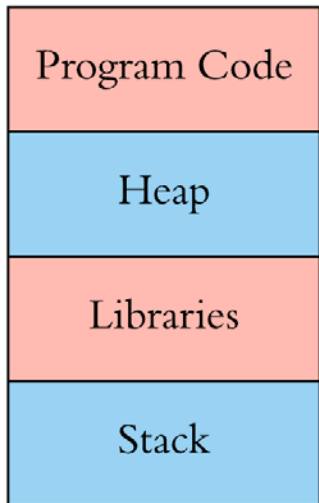
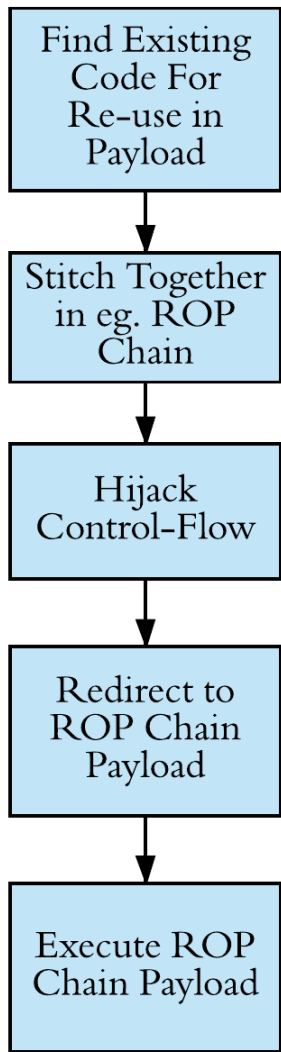


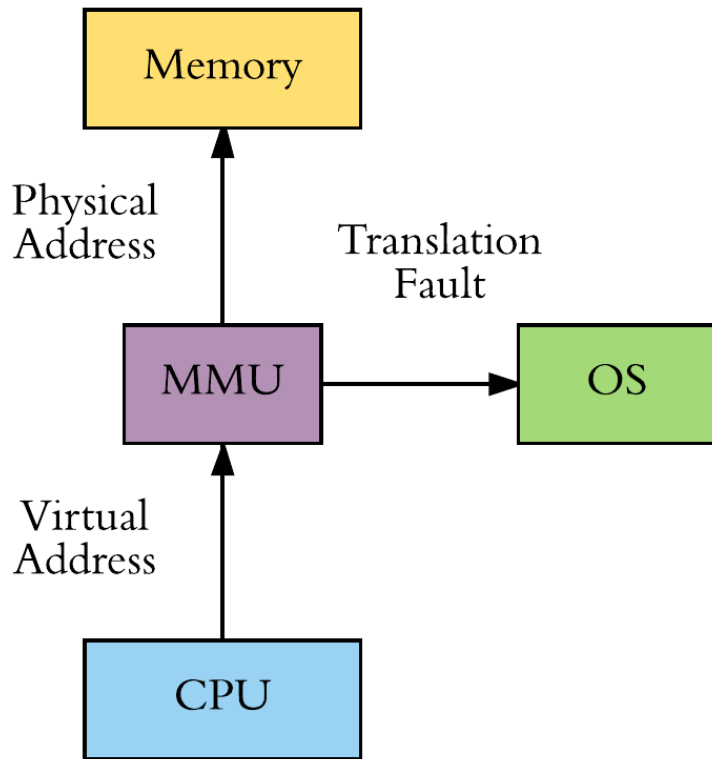
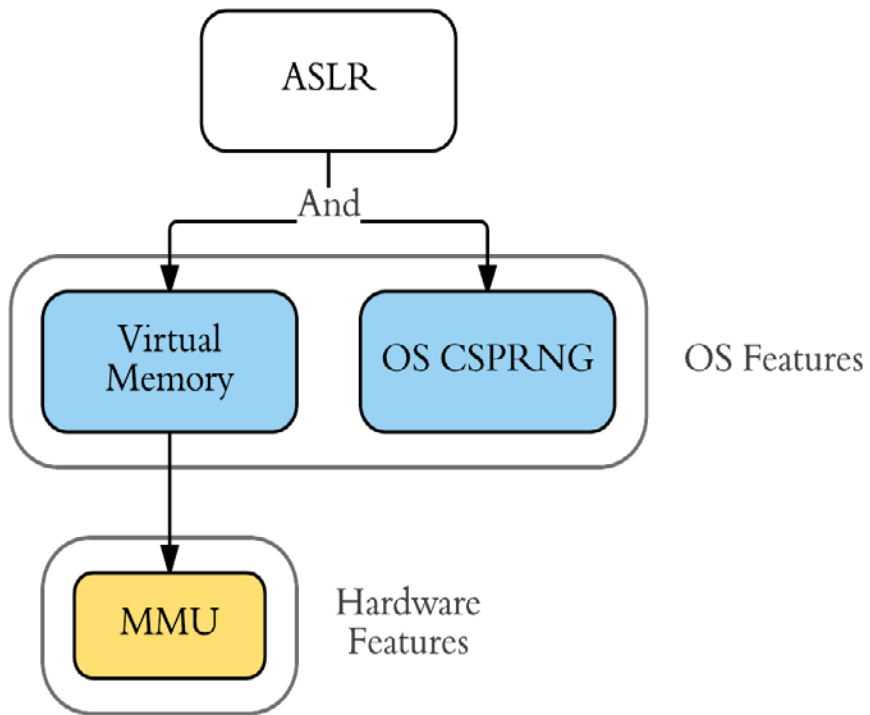
MPUs/MMUs & Hardware ESP Support

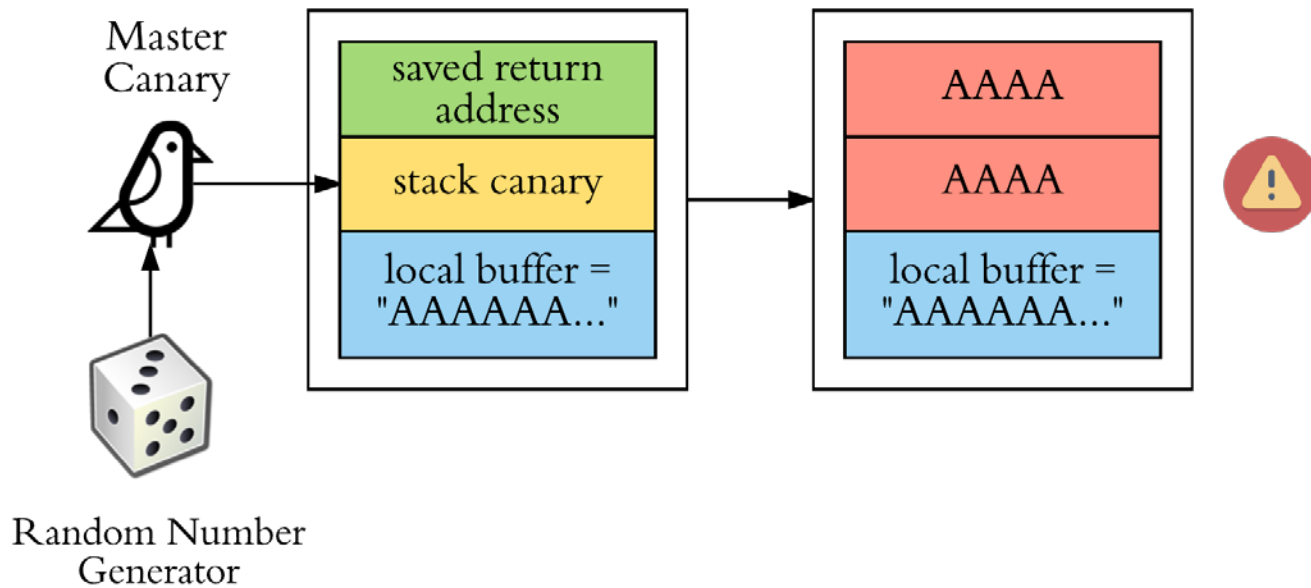
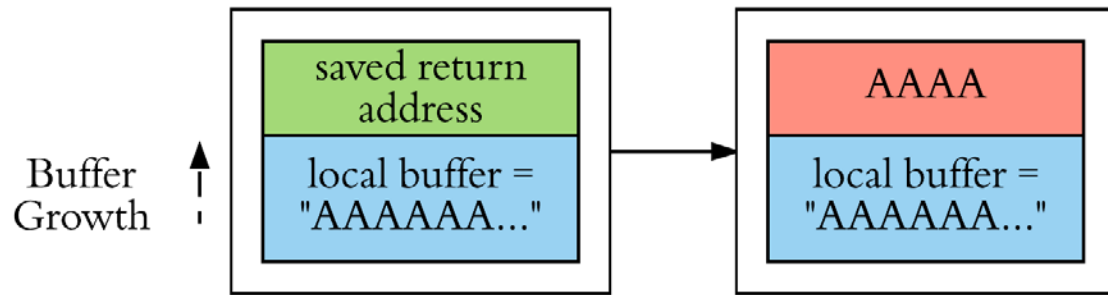


64-bit PTE (PAE mode)

return oriented programming

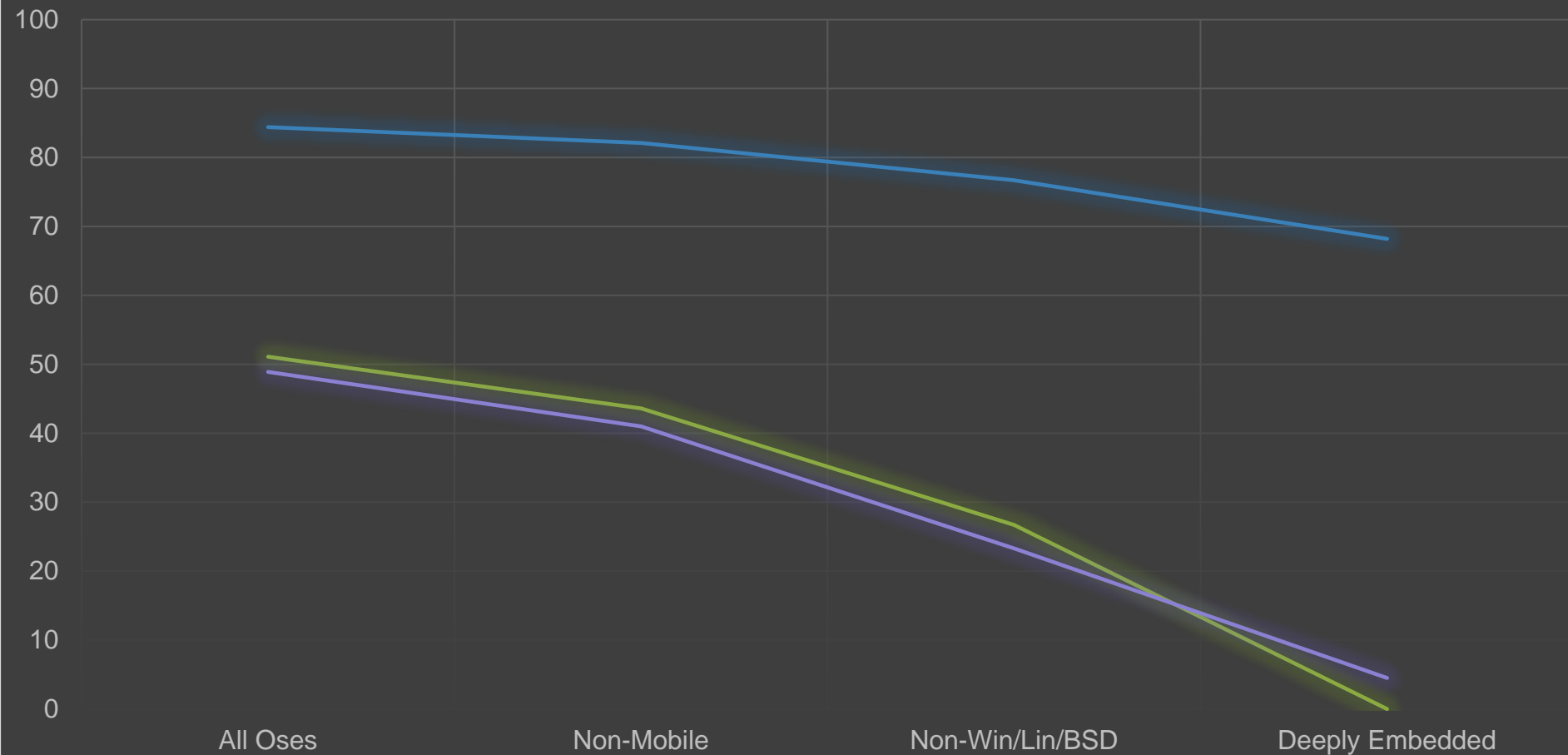






Software Dependency Support

Memory Protection Virtual Memory OS CSPRNG



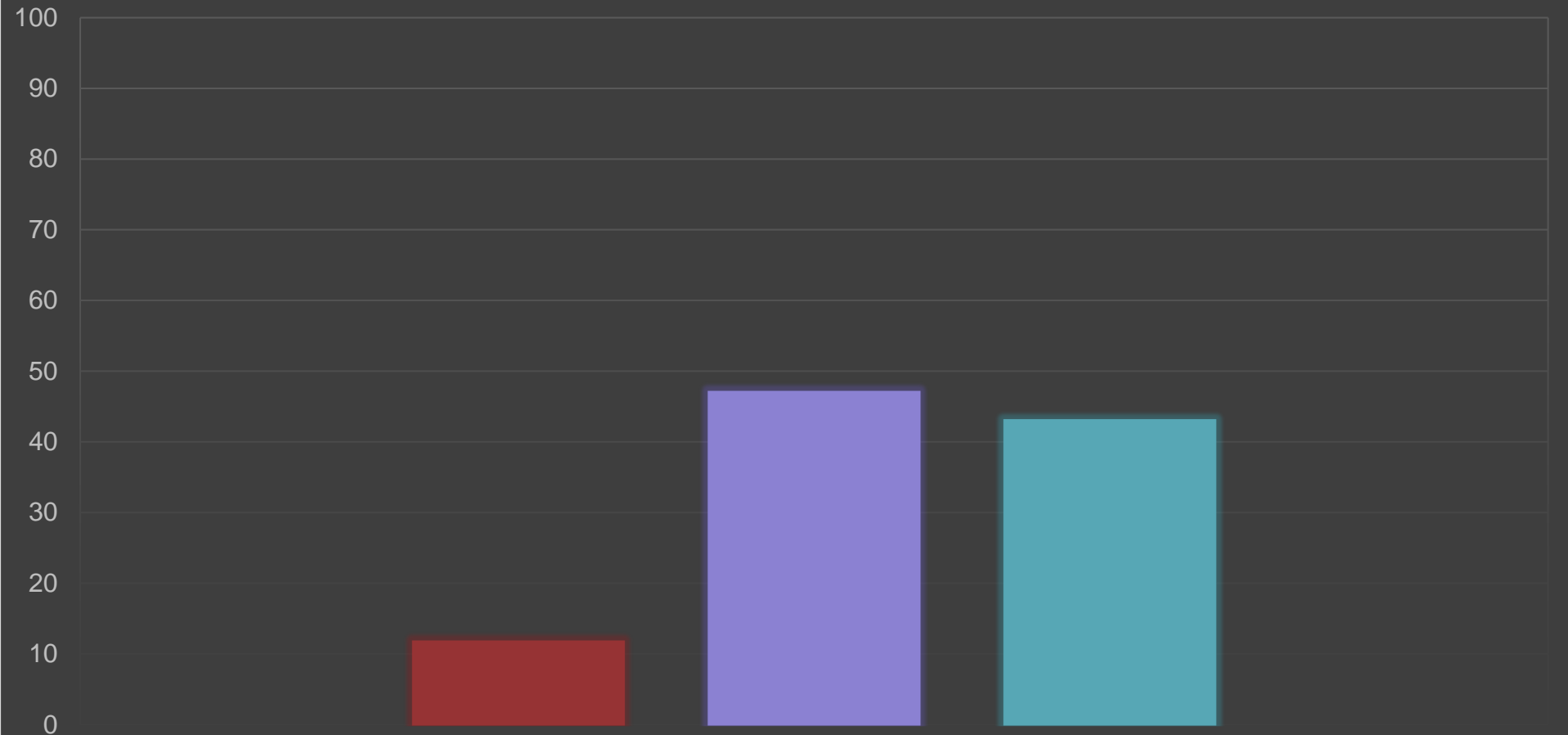


Hardware Selection

- Selected 78 Popular Embedded '*Core Families*'
- Evaluated for Hardware Dependency Support

Hardware Dependency Support (VNA)

MPU MMU Hardware ESP



A control room with multiple large monitors displaying data and charts, and several smaller monitors on a desk in the foreground. The text "Practical (I)IoT Examples" is overlaid in the center.

Practical (I)IoT Examples



Typical (Entry-Level) PLC: Modicon Momentum Unity

- ST SPEAr 320s (ARM926EJ-S)

Von Neumann

MMU (No XN Bit)

No TRNG

- VxWorks RTOS

No Mitigations

Memory Protection

Virtual Memory

No OS CSPRNG

ARM



VxWorks

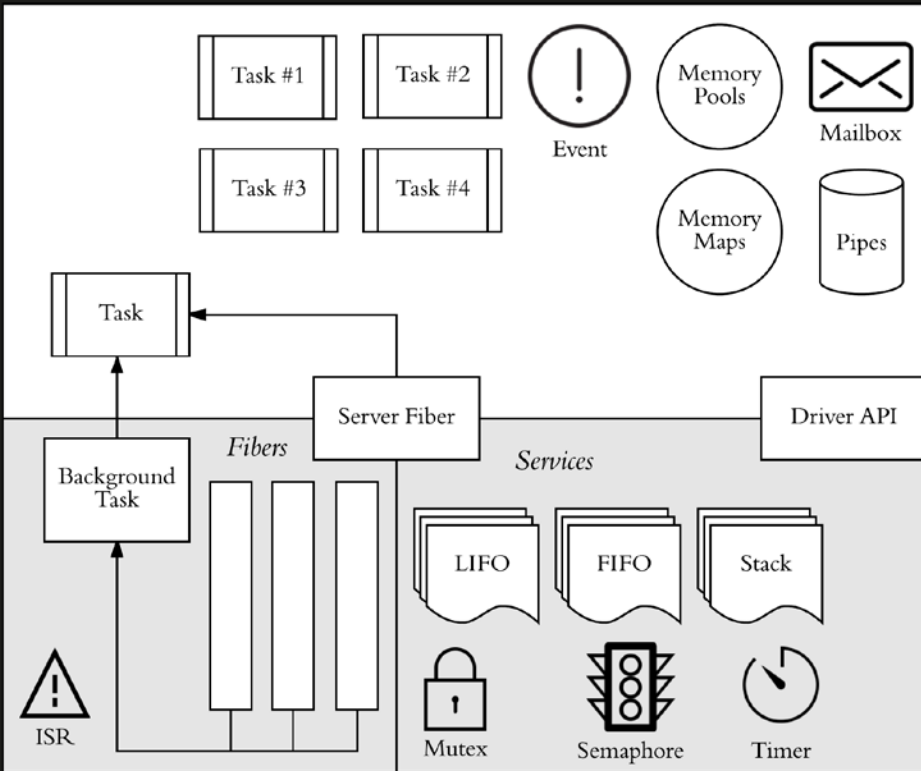


MSP430
Inside

Typical Wireless Sensor Node: Advantech WISE-1021

- MSP430F5419A
Von Neumann
No MPU / MMU
No TRNG
- TI-RTOS
No Mitigations
Memory Protection
No Virtual Memory
No OS CSPRNG

Microkernel



Nanokernel



- Library based RTOS
Based on Wind River Rocket
- OS Linux Foundation Project
Aimed at resource-constrained IoT
- Young (2016) but promising
Input from major chipmakers

Explicit focus on security



```

349 FUNC_NORETURN void _Cstart(void)
350 {
351 #ifdef CONFIG_ARCH_HAS_CUSTOM_SWAP_TO_MAIN
352     struct k_thread *dummy_thread = NULL;
353 #else
354     struct k_thread dummy_thread_memory;
355     struct k_thread *dummy_thread = &dummy_thread_memory;
356 #endif
357
358     /*
359      * Initialize kernel data structures. This step includes
360      * initializing the interrupt subsystem, which must be performed
361      * before the hardware initialization phase.
362      */
363
364     prepare_multithreading(dummy_thread);
365
366     /* perform basic hardware initialization */
367     _sys_device_do_config_level(_SYS_INIT_LEVEL_PRE_KERNEL_1);
368     _sys_device_do_config_level(_SYS_INIT_LEVEL_PRE_KERNEL_2);
369
370     /* initialize stack canaries */
371     #ifdef CONFIG_STACK_CANARIES
372         __stack_chk_guard = (void *)sys_rand32_get();
373     #endif
374
375     /* display boot banner */
376
377     switch_to_main_thread();
378
379     /*
380      * Compiler can't tell that the above routines won't return and issues
381      * a warning unless we explicitly tell it that control never gets this
382      * far.
383      */
384
385     CODE_UNREACHABLE;
386 }

```

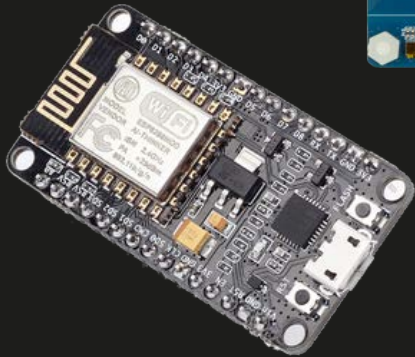
Zephyr Stack Canaries

- Based on Clang/GCC SSP
- One master canary for entire address space
Generated once at system boot
- Generated using RNG API
Implementation depends on chosen *random* driver

Zephyr Random API

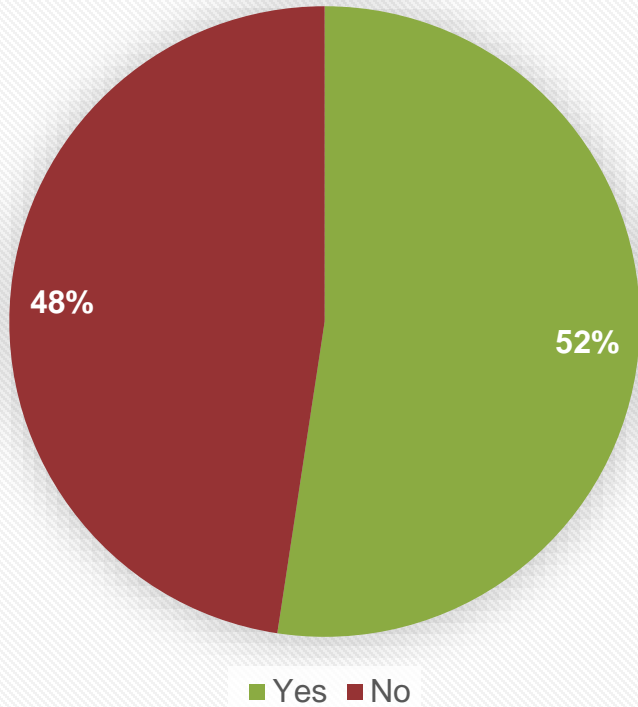
RANDOM_HAS_DRIVER (TRNG)

- RANDOM_MCUX_RNGA
NXP Kinetis K64F
- RANDOM_MCUX_TRNG
NXP Kinetis KW40Z & KW41Z
- RANDOM_STM32_RNG
STM32 Boards
- RANDOM_ESP32_RNG
ESP32 Boards
(requires Wi-Fi & Bluetooth enabled)

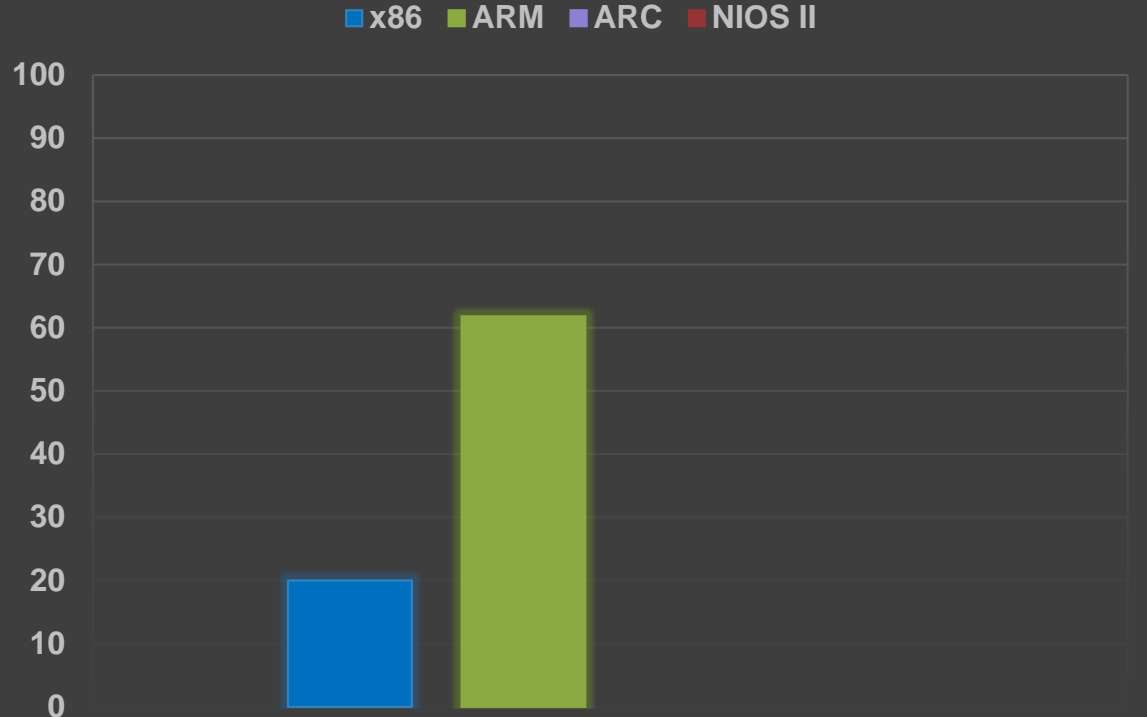


TRNGs Among Zephyr 1.8 Supported Boards

TRNG Support



TRNG Support Per Architecture





Zephyr Random API

TEST_RANDOM_DRIVER (PRNG)

- X86_TSC_RANDOM_GENERATOR /
TIMER_RANDOM_GENERATOR
Uses timestamp (eg. x86 RDTSC)

Directly, not pulled through PRNG

Canary 1st value drawn from API ->
little difference between bootruns

Infoleaks everywhere

- Had contact with Zephyr team,
plans to integrate OS CSPRNG

A black and white historical photograph showing a city street during a battle. In the foreground, a complex defensive structure is built from numerous wooden beams and logs, arranged in a crisscross pattern. In the background, a large, multi-story building with many windows is visible, appearing somewhat hazy or smoky. The word "Defense" is overlaid in white text on a dark rectangular background in the center of the image.

Defense

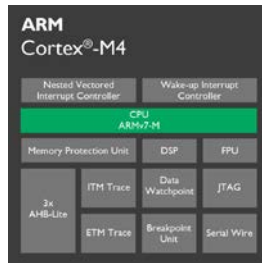
(Some) IoT Binary Security Tips



Harvard CPU



MCU with TRNG



VNA CPU
with MPU/MMU
& XN



Security-Oriented (RT)OS
(Warning: Still early days)



But: Reality Strikes (& Sucks)

- Lifespan & Legacy
What we buy *today* was designed years ago

What we design *today* might still be used in *10+ years*
- Mitigations are a stop-gap
Need to start work on long-term now



 **Jorge Aparicio**
@japaricious Volgen

[@rustlang](#), stability without stagnation, taken to robotics

Probably the first WIP self-balancing robot coded in 100% Rust

[#RustyRobots](#) 1/
Vertalen uit het Engels

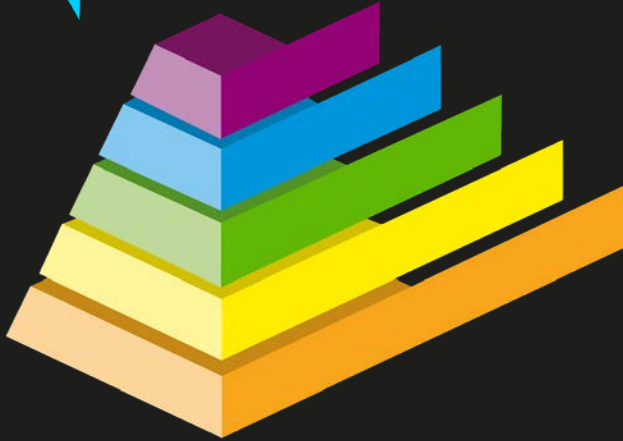


0:17

Safe Languages

- Can't emphasize this enough
- Move to safe embedded development as soon as possible

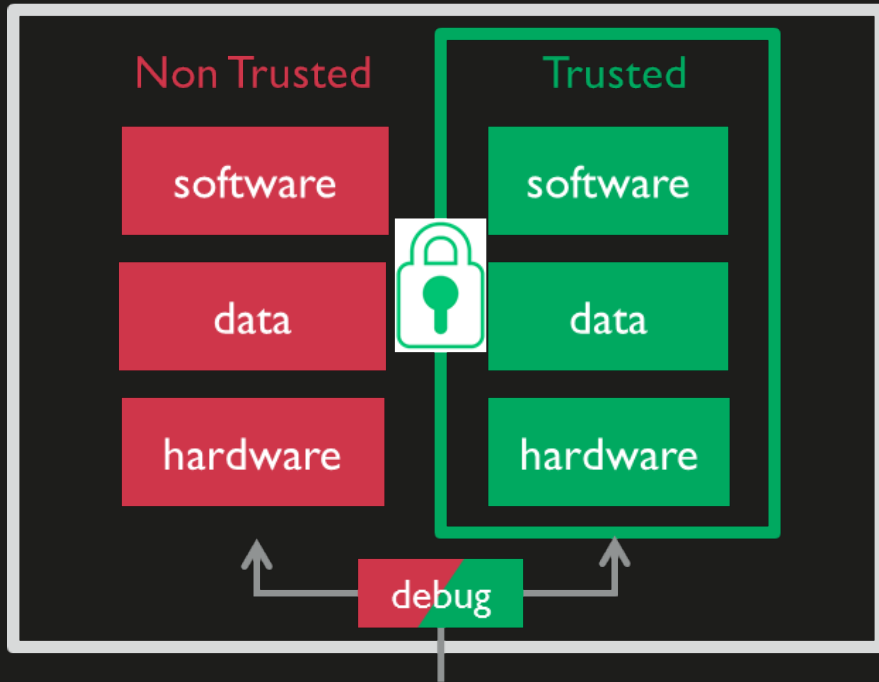




Renewable Security & Defense in Depth*

- Make your device *patchable*, ensure an *update infrastructure* is in place
- Layer your defenses, avoid single points of failure

Compartmentalization & Small TCB*



- Use hardware to enforce barriers between software components (MMU, TrustZone)
- Principle of Least Privilege
Keep your TCB Small



Failure Reporting*

- If possible, ensure failures (eg. segfaults) are reported to cloud-based backend for later analysis
- Windows Error Reporting
The Inside Story Behind MS08-067, John Lambert, 2015



Questions?



@s4mvertaka

✉ j.wetzels@midnightbluelabs.com