IMPORTANCE OF SECURE CODING IN EMBEDDED SYSTEMS

Kamal Kumar (Apt Computing Labs)

APT COMPUTING LABS

ACL

WHERE KNOWLEDGE MEETS INNOVATION.

TOPICS:

- **Common Vulnerabilities**
- **Coding Standards and Rust Adoption**
- **Future Trends and Conclusion**

- Embedded and automotive systems are increasingly connected: IoT devices, autonomous vehicles, and infotainment systems expose them to remote attacks.

COMMON VULNERABILITIES

CASINO FISH TANK:

- A casino deployed an **internet-connected thermostat** in an ornamental fish tank has been hacked.
- It's a classic **IoT** → **lateral movement** → **data theft** chain.

https://thehackernews.com/2018/04/iot-hacking-thermometer.html

CONT…

- **Cause:** An **internet-connected fish-tank thermometer** on the casino's network was the weak entry point
- **Impact:** The intruders **exfiltrated the casino's high-roller database**
- **Mitigation (standard):** Follow **NIST SP 800-82 / NISTIR 8228** and **IEC 62443**
- **Source:** Darktrace account reported via **The Washington Post** and **Business Insider**.

JEEP CHEROKEE:

- In 2015, hackers remotely took control of a Jeep Cherokee, manipulating its brakes and steering through a vulnerable embedded system. This incident exposed how a single coding flaw can endanger lives.

https://www.youtube.com/watch?v=MK0SrxBC1xs

CONT...

- **Cause**: Insecure C/C++ code lacked input validation and encryption, exposing network interfaces
- **Impact**: Recall of 1.4 million vehicles.
- **Mitigation**: **Standards**: ISO/SAE 21434 mandates secure communication protocols.
- **Rust**: Libraries like tokio ensure encrypted, safe networking.
- **Source**: https://www.wired.com/2015/07/hackers-remotely-kill-jeep-highway/

HEARTBLEED:

- The Heartbleed bug in the OpenSSL cryptographic software library, allowing attackers to steal sensitive information from server memory by sending malformed TLS "heartbeat" requests.

CONT…

- **Cause**: C code error in OpenSSL's heartbeat mechanism failed to validate memory access.
- **Impact**: Affected millions of devices, exposing user data and enabling server compromises.
- **Mitigation**: **Standards**: CERT C MEM00-C enforces proper memory deallocation.
- **Rust**: Ownership model prevents use-after-free errors at compile time.
- **Source**: https://heartbleed.com

HOW TO PROTECT ?

- Secure Hardware Design - FIPS 140-3, Common Criteria (ISO/IEC 15408)
- Secure Programming Practices - MISRA C/C++, ISO/IEC 27034
- Secure Boot and Firmware Integrity - NIST SP 800-193, TCG D-RTM Network
- Security and Communication - IEC 62443, IEEE 802.1X
- Regular Updates and Patch Management - NIST SP 800-147, ISO/IEC 30111
- Access Control and Authentication - OAuth 2.0, OpenID Connect, IEC 62443-4-2

SECURE PROGRAMMING PRACTICES:

PURPOSE:

- Standards enforce safe, reliable, and maintainable code in embedded systems and robotics, where errors can lead to hacks.

KEY STANDARDS AND FOCUS AREAS:

- **MISRA C**: Safe C programming
- **CERT C/C++**: To eliminate memory vulnerabilities.
- **ISO/SAE 21434**: Cybersecurity for automotive embedded systems.
- **IEC 62443**: Security for industrial control systems (e.g., robotics PLCs).

APPLICATION:

- Tools like Cppcheck enforce MISRA and CERT, catching errors early.
- Example: A robotic arm's firmware adhering to IEC 62443 avoids unauthorized control, unlike Stuxnet's exploit.

WHY RUST ?

WHY RUST ?

- Even with MISRA C and CERT C/C++ guidelines, vulnerabilities persist due to inherent unsafe constructs.

- **Key Message**: Traditional C/C++-based approaches **cannot fully eliminate memory and concurrency bugs**.

CORE FEATURES OF RUST:

- **Memory Safety without Garbage Collector**:
  - Ownership & borrowing
- **Concurrency Safety**:
  - Compiler enforces safe parallelism
- **Zero-Cost Abstractions**:
  - Performance as good as C.
- **no_std Support**:
  - Rust runs without standard library → ideal for microcontrollers, RTOS, bare-metal automotive ECUs.

BENEFITS:

- Reduce Recalls & Liabilities
- Future-proof Talent & Ecosystem
- Integration Friendly
- Regulatory Edge

- **Philosophy**:

Rust is not just a programming language; it is a **strategic enabler** for secure, safe, and reliable automotive/embedded systems.

Coding Standards and Rust Adoption

Aspect	How Rust Helps	Relevant Standards
Memory Safety	Prevents buffer overflows and use-after-free errors at compile time.	IEC 62443, MISRA C, ISO/SAE 21434
Type Safety	Catches errors early with strong type checking and pattern matching.	ISO 26262, IEC 61508
Concurrency Safety	Eliminates data races for safe multi-threaded operations.	IEC 62443, ISO/SAE 21434
Legacy Integration	Safely integrates with C/C++ code, securing legacy systems.	UNECE R156, IEC 62443
Embedded Suitability	Supports `no_std` for lightweight, predictable code on microcontrollers.	ISO 26262, ETSI EN 303 645
Secure Development	Tools like `cargo-clippy` and `cargo-audit` ensure secure coding practices.	ISO/IEC 27001, NIST CSF
Cryptography	Provides safe libraries (e.g., `rustls`) for secure communication and encryption.	IEC 62443, ISO/SAE 21434

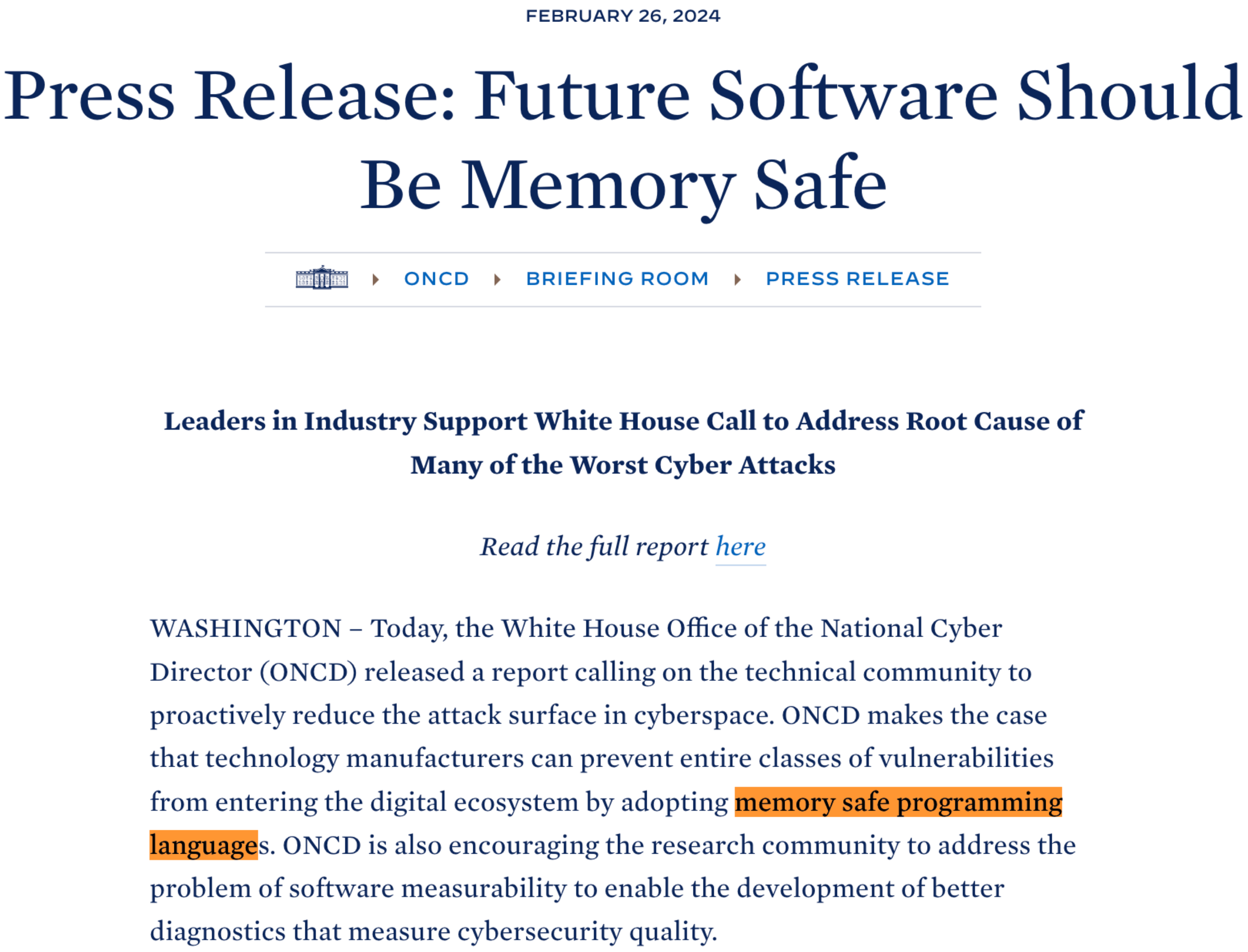
ADOPTION:

- **Microsoft**: Rust in Windows drivers for memory safety.
- **AWS**: Firecracker VM for IoT and robotics.
- **Auterion**: Drone firmware, preventing crashes in safety-critical systems.
- **Linux Kernel**: Rust support since 2022, signalling industry shift.

LINUX TORVALD'S COMMENT ON RUST ADAPTATION:

"I was expecting [Rust] updates to be faster, but part of the problem is that old-time kernel developers are used to C and don't know Rust," Torvalds said.

Coding Standards and Rust Adoption



FEBRUARY 26, 2024

Press Release: Future Software Should Be Memory Safe

ONCD ▸ BRIEFING ROOM ▸ PRESS RELEASE

Leaders in Industry Support White House Call to Address Root Cause of Many of the Worst Cyber Attacks

Read the full report here

WASHINGTON – Today, the White House Office of the National Cyber Director (ONCD) released a report calling on the technical community to proactively reduce the attack surface in cyberspace. ONCD makes the case that technology manufacturers can prevent entire classes of vulnerabilities from entering the digital ecosystem by adopting memory safe programming languages. ONCD is also encouraging the research community to address the problem of software measurability to enable the development of better diagnostics that measure cybersecurity quality.

https://www.whitehouse.gov/oncd/briefing-room/2024/02/26/press-release-technical-report/

REFERENCES

- MISRA C Guidelines: https://www.misra.org.uk
- Rust Embedded Book: https://docs.rust-embedded.org/book/
- Zero Trust in Robotics: https://youtu.be/jfPw8gH1i2I?feature=shared