Zephyr® Project
Developer Summit

Insights from Zephyr Security Audit and Vulnerabilities Experiences

Flavio Ceolin (flavio.ceolin@intel.com)
David Brown (david.brown@linaro.org)

THE LINUX FOUNDATION

#EmbeddedOSSummit

@ceolin
@d3zd3z

Who we are

Agenda

- Introduction
- Overview of past vulnerabilities on Zephyr
- External code audit
- Lessons learned
- Strategies implemented to enhance Zephyr’s security
- Conclusion

Embedded System Security

“The Zephyr OS is based on a small-footprint kernel designed for use on resource-constrained and embedded systems“

- Security in Embedded Systems is critical !
  - Embedded systems are increasingly connected to networks, making them vulnerable to cyberattacks.
  - Security breaches in embedded systems can have serious consequences, including data breaches, system malfunctions, and safety hazards.

Zephyr: A framework for Secure Embedded Development

- Zephyr offers features such as memory protection, secure boot, and trusted firmware.
  - Modules provide additional features
- Groups dedicated to continuously improving the security of the framework.
  - Security Committee
  - Security Working Group
- Regular security updates and patches are released to address vulnerabilities and enhance security features.

CNA

- **Registered with MITRE** in 2017
  - We issue our own CVEs
- **Zephyr Project Security Incident Response Team** (PSIRT)

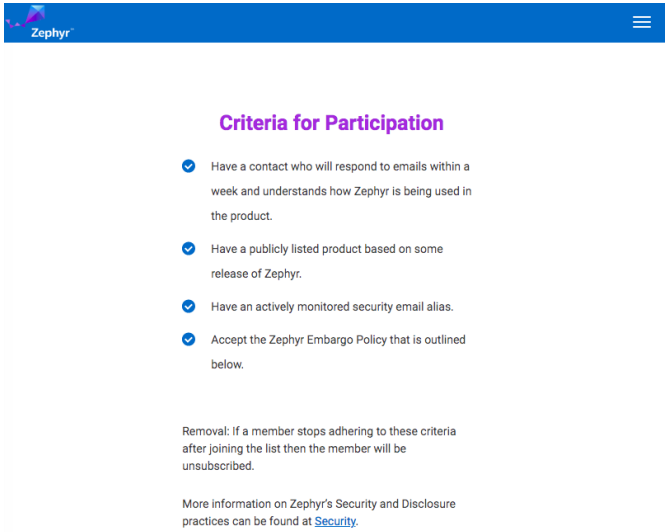
Zephyr Project

The majority of the links on this page redirect to external websites ⧉; these links will open a new window or tab depending on the web browser used.

Scope	Zephyr project components, and vulnerabilities that are not in another CNA's scope
Root	MITRE Corporation
Security Advisories	View Advisories
Program Role	CNA
Organization Type	Vendors and Projects
Country*	USA

Vulnerability Alert Registry

- For an embargo to be effective, product makers need to be notified early so they can remediate
- Goal: Zephyr to fix issues within 30 days to give vendors 60 days before publication of vulnerability
- Product makers can register to receive these alerts for free by signing up at Vulnerability Alert Registry



Overview of past vulnerabilities

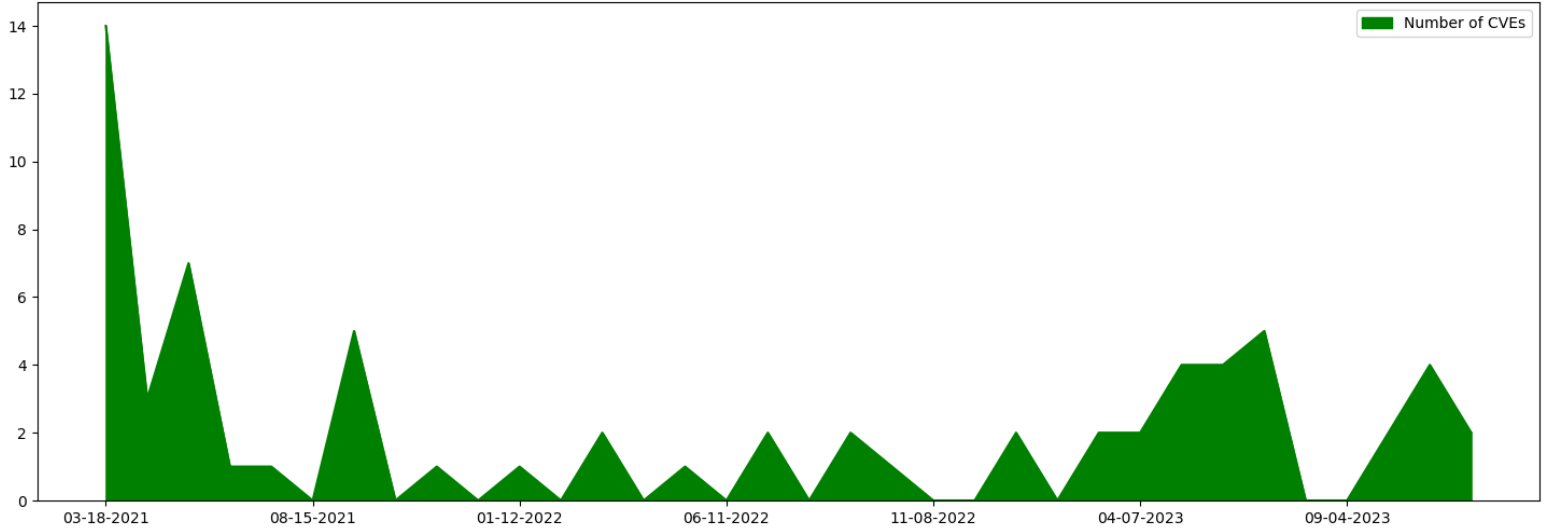
Public Vulnerabilities

Data is from CVEs published in the last three years !

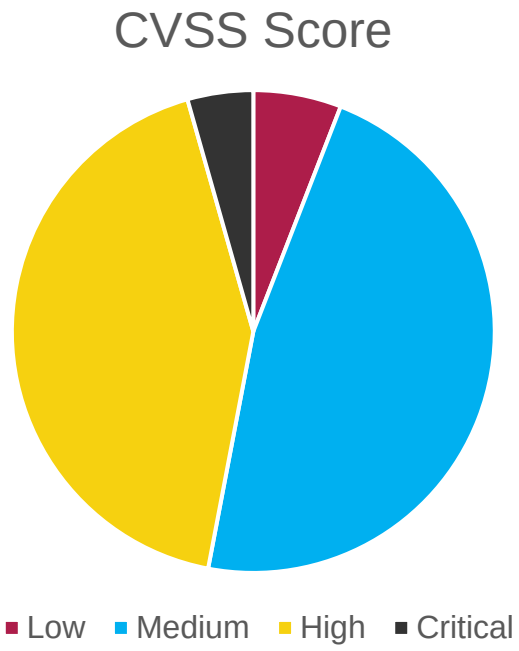
*CVE - **Common Vulnerabilities and Exposures**

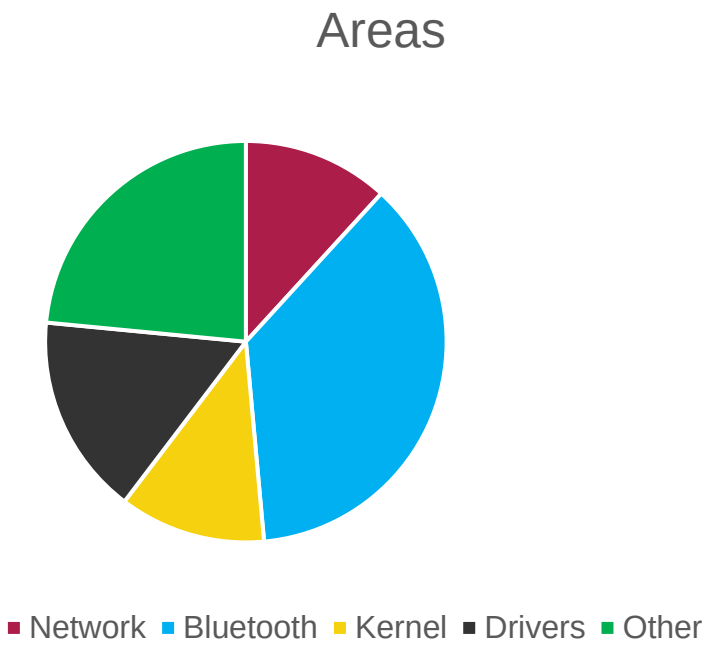
Published Vulnerabilities

Total of CVEs published : 68

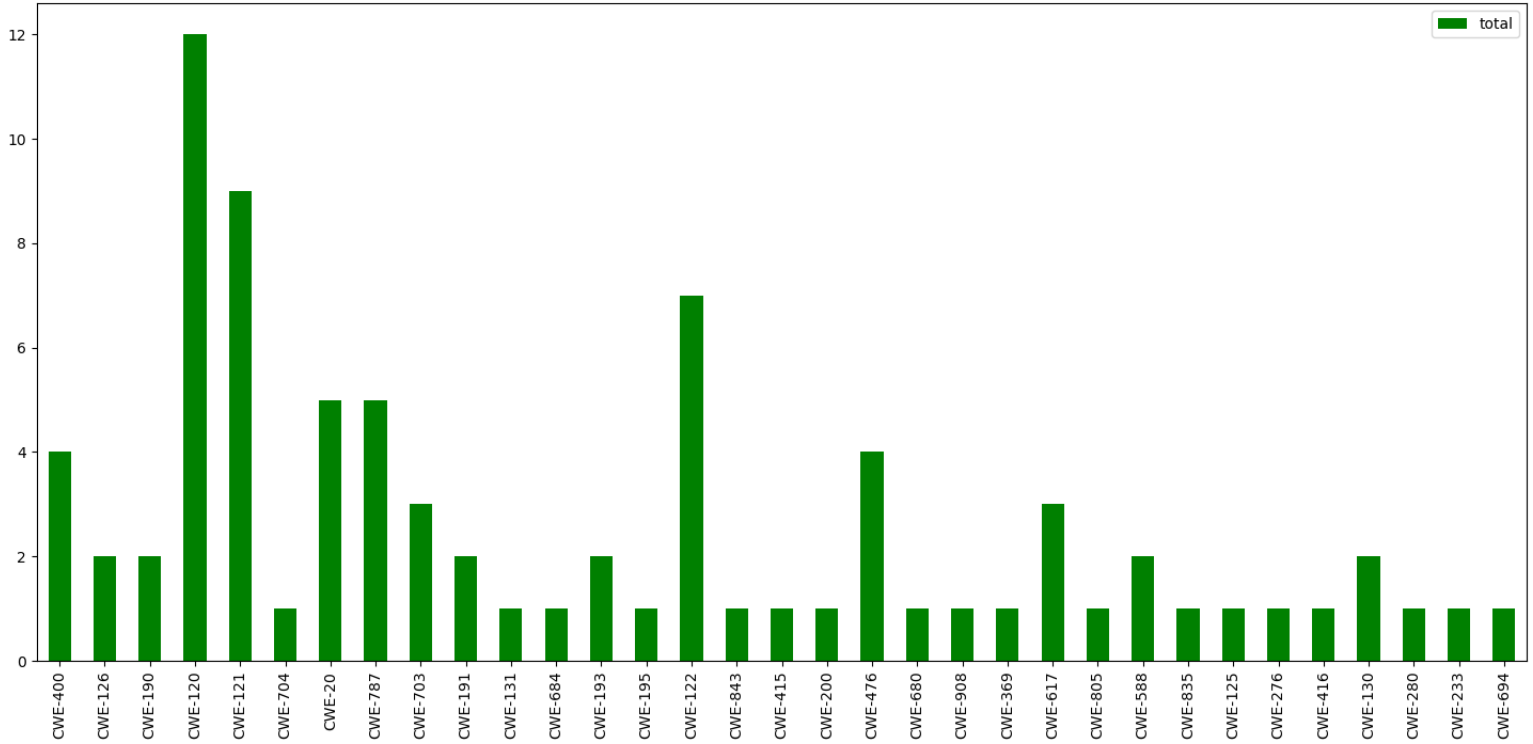


Published Vulnerabilities





Published Vulnerabilities



CWEs

- CWE-120: Buffer Copy without Checking Size of Input ('Classic Buffer Overflow’)
- CWE-121: Stack-based Buffer Overflow
- CWE-122: Heap-based Buffer Overflow
- CWE-20: Improper Input Validation
- CWE-787: Out-of-bounds Write

Some conclusions and questions

- Unsafe programming language
  - C is prone to buffer overflow issues
- Lack of awareness and training
- Excessive optimization and performance concerns ?
  - Insufficient validation and sanitization
- Network / Bluetooth -> Easier to fuzzy

External code audit

Why an external audit ?

- Identifying Vulnerabilities
- Independent Assessment
- Best Practices
- Community Trust
- Reputation

How we choose the auditor

- Expertise in Embedded Systems
- Reputation
- Communication
- Cost
- Experience with Zephyr RTOS

How we have defined the scope

- Security Objectives
- Components
  - Narrow to something doable and that benefits most users
- Depth of Analysis
- Threat Model

Scope

The primary focus is centered around Zephyr’s core kernel features

- User mode support
  - memory management and protections, user and supervisor threads
  - System calls
- Inter-process communication and process scheduling
- Exploit mitigations
  - Stack canaries
  - Stack guard
  - Stack pointer randomization

Findings

- NCCGroup
- Target Zephyr 3.6 / 3.7
  - 02/2024 ~ 03/2024

- Three issues found
  - Two low severity caused by integer overflow and TOCTOU
  - One informational caused by integer overflow

Lessons learned

Lessons

- Defining the scope is hard
  - Resource Constraints
  - Depth and Breadth
  - Future-Proofing
  - Stakeholder Agreement

Lessons

- Threat model is worth
  - Guiding the Audit Process
  - Validating Security Controls
  - Facilitating Communication

- Comprehensive Testing
  - The audit make it clear the importance of comprehensive testing

Strategies implemented to enhance Zephyr’s security

Strategies

- Security Training
- Improve automated Security Checks
- Monitoring vulnerabilities in third-party components and dependencies used in Zephyr RTOS
- Community Engagement

Conclusion

Outcome

- Enhanced Security
  - The identification and subsequent remediation of even low-severity issues contribute to a more secure system
- Increased Confidence
  - Third-party auditor validated the security and quality of the code base increasing confidence among developers, stakeholders, and users
- Recommendations aligned with Zephyr plans
  - Guided Fuzzing of Libraries and Subsystems

Positive findings

"The overall design and documentation of Zephyr's kernel demonstrated a well understood attack surface and threat model, especially in regards to maintaining user thread privilege separation and isolation."

"Strong defensive programming practices were employed holistically across the kernel's codebase"

Questions ?

Thank you !

Zephyr® Project

Developer Summit

Published Vulnerabilities