C++ Security Fundamentals

Pitfalls and practices

Assaf Tzur-El

“

C makes it easy
to shoot yourself in the foot;
C++ makes it harder,
but when you do –
it blows your whole leg off.

Bjarne Stroustrup

Assaf Tzur-El

- Love to help
- Develop
- Architect
- Teach
- Robotics competition judge

The road (blocks) to working software

- Syntax errors
  - Handled by compiler
- Logical errors
  - Unit tests to the rescue!
- Memory safety
  - 😳
- Unpredictable behavior
  - 😱

C++ unpredictability

- Undefined behavior
- Unspecified behavior
- Implementation-defined behavior
- More

Safety

- Lifetime safety
- Bounds safety
- Type safety
- Thread safety
- Runtime checks

~~The answer~~

- ~~Switch to a safer language!~~
- ~~CISA, NSA, ONCD's goal~~
- ~~Not practical~~
  - ~~Existing code base~~
  - ~~Existing ecosystem~~
  - ~~Requirements~~
  - ~~No alternative~~

Solutions

- Standardization
  - C++ standard
  - Safe C++
  - ISO 26262
- Coding practices
  - OWASP Top Ten
  - Defensive programming
  - Secure by Design

Enforcement

- Rules
  - MISRA
- Tools

Motor Industry Software Reliability Association

MISRA C++:2023

Guidelines for the use of C++17 in critical systems

October 2023

Example

Rule 9.6.1 The `goto` statement should not be used

Category Advisory

Analysis Decidable, Single Translation Unit

Rationale

The use of `goto` is usually regarded as bad programming practice as it can lead to code that is difficult to understand and analyse. Restructuring code to avoid its use generally leads to code that has a lower level of complexity.

Another example

Rule 9.4.2 The structure of a `switch` statement shall be appropriate

[stmt.switch]
[dcl.attr.fallthrough]

Category Required

Analysis Decidable, Single Translation Unit

Amplification

A `switch` statement is structured appropriately when it conforms to the following restrictions:

1. The *condition* shall only be preceded by an optional *simple-declaration*;

...

7. Every `switch` statement shall have a `default` label, appearing as either the first label of the first *switch label group* or as the last label of the last *switch label group*.

Yes, but…

Rule 0.0.1 A function shall not contain *unreachable* statements

[IEC 61508-7] / C.5.9
[DO-178C] / 6.4.4.3.c
[ISO 26262-6] / 9.4

Category Required

Analysis Decidable, Single Translation Unit

Amplification

A statement is *unreachable* if the block containing it is not *reachable* from the *entry block* of the Control Flow Graph (CFG) for the function.

Why? (developer perspective)

- Our goal: Working software, not writing code
- Code quality
  - Reliability
  - Maintainability
- Safety
  - Focus on critical systems
  - Avoid dangerous patterns
- Portability
- Code reviews and audits

Why (org perspective)

- Quality
- Standards
- Cost and time
- Customer trust

“

C makes it easy
to shoot yourself in the foot;
C++ makes it harder,
but when you do –
it blows your whole leg off.

Bjarne Stroustrup

C++ Security Fundamentals


https://www.linkedin.com/in/assaftzurel/


assaf@tzurel.co.il


https://wa.me/972543330085


http://s.pashut.co.il/corecpp-2024


Eran Gilad, Gilad Darmon, Nir Dobovizki

Questions?