Strength and Weaknesses of the C Programming Language

Danial Tariq

January 20, 2025

Advanced Systems Programming

Strengths

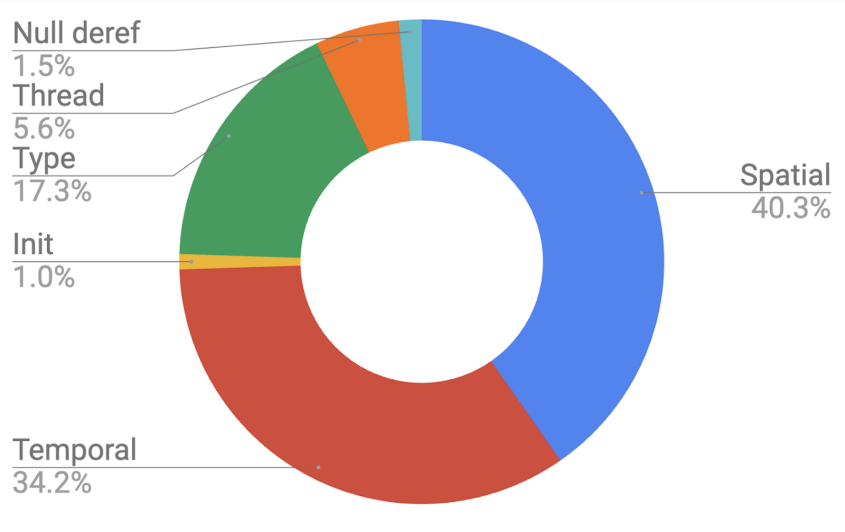
- Control over layout of data structures down to individual bits.
- Manual memory management – Makes it easier for developers to create more efficient memory allocation strategies.
- Highly portable – pretty much every platform has a C compiler.

```
struct rtp_header {
uint16_t cc:4
uint16_t x:1;
uint16_t p:1;
uint16_t v:2;
uint16_t pt:7;
uint16_t m:1;
uint16_t seq;
uint32_t ts;
uint32_t ssrc;
}

char *buffer = malloc(BUFLEN);
// ...
```

Weaknesses - Memory Safety

- Incredibly easy to introduce memory safety vulnerabilities.
- Lack of bounds checking – this has been the cause of some of the most severe security vulnerabilities, e.g. Heartbleed, Crowdstrike outages.
- Lack of temporal safety – use-after-free, double-free, etc.



Breakdown of memory safety zero-day exploits by vulnerability class

Source: Google Security Blog. `https://security.googleblog.com/2024/11/retrofitting-spatial-safety-to-hundreds.html`

Weaknesses - Undefined Behavior

- There are many cases where the C standard deliberately leaves the behavior of certain operations undefined.
- This is often done for performance reasons, but can lead to subtle bugs that are difficult to diagnose.
- This becomes even more challenging when you consider the behaviour of optimising compilers in the presence of undefined behaviour.
- This can also interfere with the programmer's mental model of the layout of data structures in memory.