1. debug\_assert!()

The debug\_assert macro is used to compare an expression to true or not. This macro gets disappeared when build in release mode (“cargo build –release”)

1. unsafe {} block

Rust does not check code written under unsafe{} block and thus developer can bypass rust protections and leads to any memory leaks/overflows.

Eg: allows us to execute assembly instructions with asm! macro and this can lead to memory corruptions if not coded properly

1. CVE-2021-29922 – leading zeros does not get truncated in std::net library allows attackers to bypass X-Custom-IP header checks.

Link: <https://sick.codes/sick-2021-015/>

1. Command Execution via std::process::Command struct – allows to execute OS commands
2. Path, open, read\* functions by default accepts any file path. The path should be sanitized before passing on to these functions