**Compiler Selection**

Compiler selection is important to ensure a system operates safely and securely. Compilers are important as they are the intermediary between the human readable source code and the machine readable binary code. This crucial step is often overlooked and compilers, unless coming from a trusted source with digital signature, should be treated as any other commercial off the shelf software that has an unknown pedigree.

Often, developers analyze the source code to detect any code that can negatively impact security or safety. This aims to solve one part of the problem. After the source gets compiled, we need to be sure that the compiler did not insert any logic (maliciously or inadvertently) into the binary that compromises the systems security or safety. This is especially important because this type of vulnerability will be inserted into every piece of software that the compiler is used for compilation.

To combat against this, developers of security or safety critical systems should only use compilers from a trusted source with a digital signature. The trusted source should also provide evidence that the compiler is free from anomalous behavior, similar to the way RTCA’s DO-178B defines qualifiable tools. In addition, developers of critical software can perform source to binary traceability to ensure the compiler has not inserted any undesired logic into the binary code.

# Bibliography

Thompson, K. (1984). Reflections on Trusting Trust. *Communications of the ACM* , 761-763.