RUNSAFE SECURITY

RunSafe Security Safety of Flight Approach

RunSafeSecurity.com

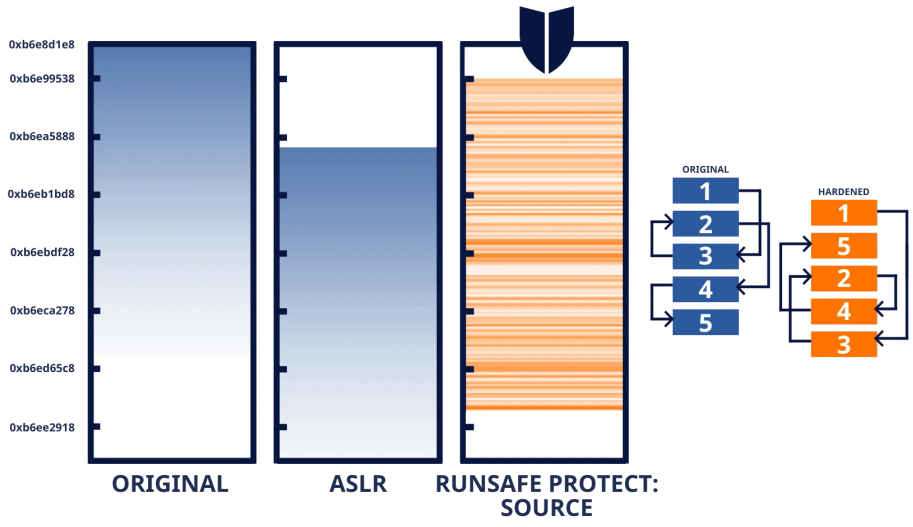
Table of Contents

Executive Summary

RunSafe's product, RunSafe Protect, will be certifiable for flight safety at the highest level through DO-178C at DAL A and qualifiable by DO-330 at TQL 1.

There are two components of RunSafe Protect. The first operates during compilation, and is used in a Computer Software Configuration Item (CSCI) development environment, while the other operates during software execution. The development tool takes the CSCI source code as input and outputs an equivalent binary with RunSafe metadata. The airborne software is the compiled binary along with the RunSafe Protect library.

When loaded, the binary's functions are relocated deterministically with a specific seed. The development component will be qualifiable as a tool at level 1, and the execution component will be certifiable at design assurance level A. The plan outlined in this document, put together with the help of avionics safety experts at AFuzion, explains how RunSafe intends to reach this goal.



Cyber Security – The Threat

Foundational software, especially those running critical infrastructures, are built on the backbones of C and C++. These languages, while powerful, have an Achilles' heel: vulnerabilities arising from their handling of memory. According to the NSA's advisory on memory safety, 70% of both Google's and Microsoft's security fixes are memory safety related[1]. Such missteps can serve as exploitable vulnerabilities for those with ill intent as evidenced by MITRE's Top 25 Most Dangerous Software Weaknesses which had memory safety as number 1 (CWE-787), 3 of the top 10, and 7 of the top 25[2].

A key concern here is the inherent predictability within these languages. Without RunSafe, when a bad actor finds memory vulnerabilities in one software binary, they are able to develop an exploit that works anywhere that binary is deployed. This isn't just a bug; it's a golden ticket for those with malicious intent. With attackers constantly improving their binary analysis tooling and the increasing reliance on open source code, attackers have more opportunities to find these vulnerabilities than ever.

To tackle this, it's not enough to just patch these memory-based vulnerabilities. Instead, the approach needs to be more strategic.

RunSafe Protect improves unpredictability by changing how software interacts with memory and introducing relocations into function loads. This makes each software instance unique, diminishing the chances of a single exploit affecting everything. It also makes it incredibly difficult to develop an exploit from a RunSafe protected binary.

RunSafe Protect is designed to address several of the common weaknesses and vulnerabilities identified by NIST/MITRE. The weaknesses and vulnerabilities addressed by RunSafe are elaborated in the tool qualification package that is provided to the CSCI developer. Not only does the package provide information to describe the anticipated threat, but it also describes how the RunSafe Protect product is designed to address these weaknesses/vulnerabilities via its relocation features. The information needed to describe RunSafe's security features will be provided as a set of shell documents. These documents provide the evidence needed to show that using RunSafe Protect satisfies all the objectives and activities called for by DO-326/356 and described in section 6 below.

[1] https://www.nsa.gov/Press-Room/News-Highlights/Article/Article/3215760/nsa-releases-guidance-on-howto-protect-against-software-memory-safety-issues/

[2] https://cwe.mitre.org/top25/

RunSafe Protect – How it Works

RunSafe Protect hardens software by relocating where functions load into memory uniquely for every software load for each instance deployed.

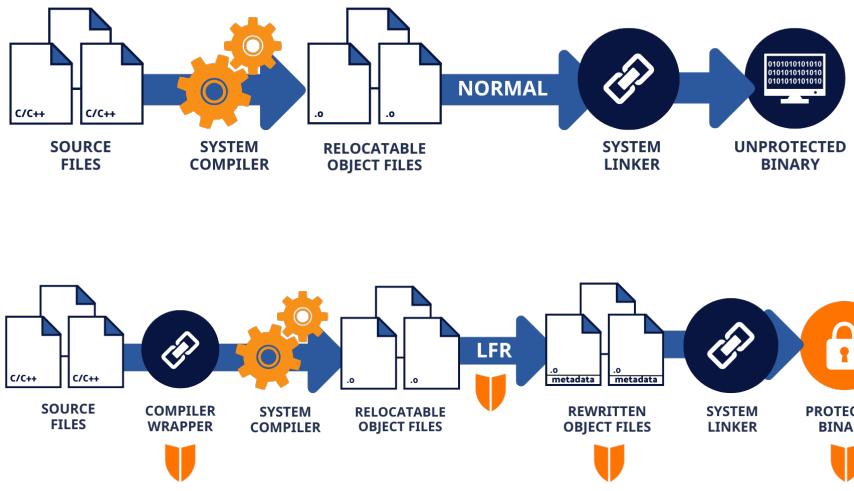
This software diversity denies attackers the ability to exploit memory-based weaknesses in software, which comprise nearly 70% of vulnerabilities in compiled code. Since RunSafe Protect does not change any lines of source code in the CSCI product, there is no change in system performance and no change in functionality.

a. Development Component:

When RunSafe Protect is built into software, the attacker cannot reliably exploit the protected software because the code they need to execute is never in the same place twice. Additionally, any failed attempts to exploit RunSafe Protect-protected software result in the program crashing. When the software is launched again, the information an attacker could gain from the previous failed exploit attempt is not useful because the program will be relocated again.

RunSafe Protect allows a user to use their existing compiler and linker. RunSafe Protect sits in front of the linker, first making the modifications needed, and then calling the existing linker. This process can be seen in the figure below. The top path through the process is what a normal build looks like, abstracting away the infinite complexities that build systems bring. Protect has been proven to work with builds as simple as a "Hello, World" example built with GCC all the way to a complex Yocto-based build using different compilers per project. With RunSafe Protect in the mix, compilation takes a slightly different path at the linking stage, where RunSafe injects information needed to accomplish our runtime relocation. The protected binary has the same control flow as the unprotected binary.

This part of RunSafe Protect is utilized during the development process in the CSCI producer's labs. Therefore, this portion of RunSafe Protect will be qualifiable as a tool under DO-330 at TQL-1.



b. Airborne Component

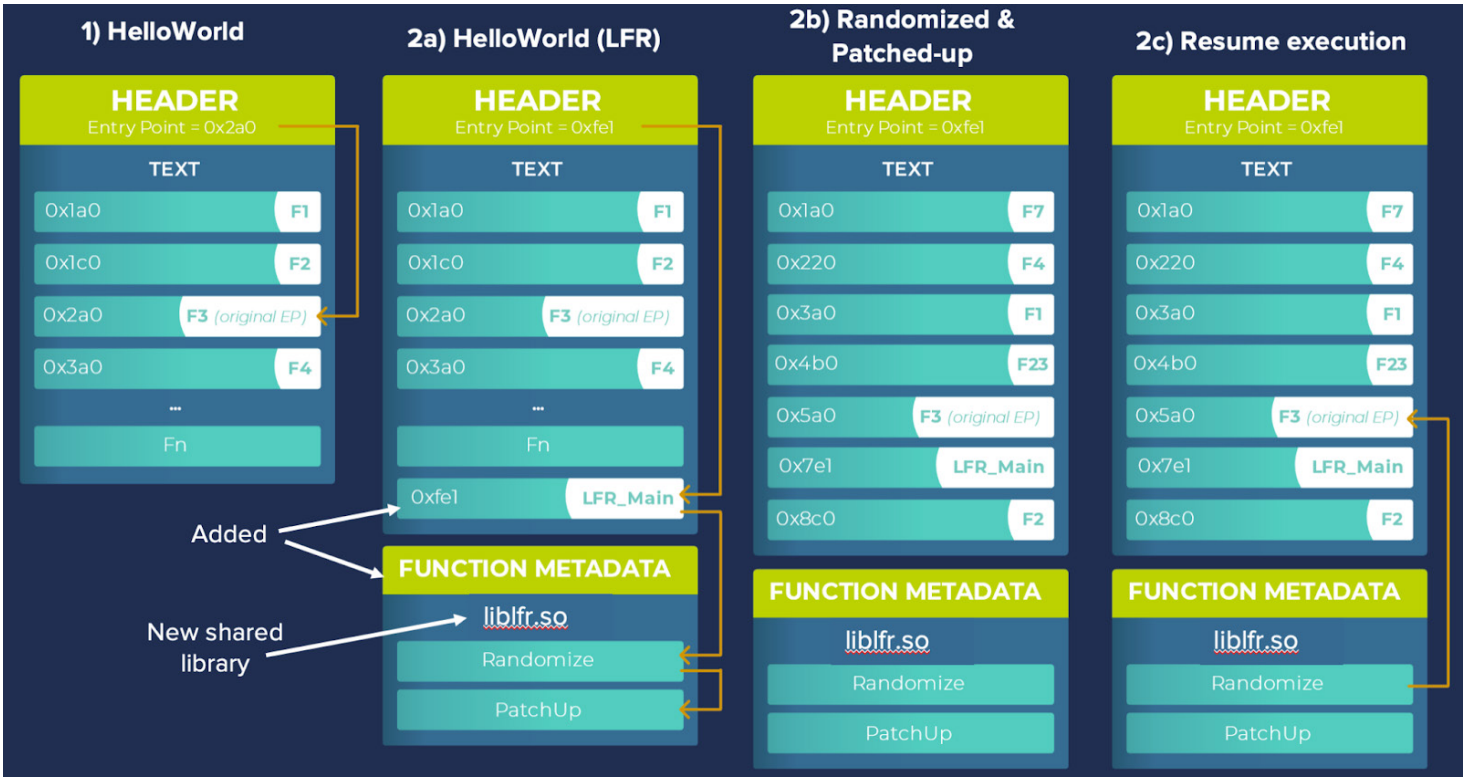
The next step of RunSafe Protect takes place when the protected software is loaded into RAM on an airborne system. The software will be certifiable under DO-178C since it is airborne software.

The diagram below details how RunSafe Protect creates a unique memory layout for each binary and shared object when integrated into a build process. The diagram walks through the process setup for a HelloWorld application without RunSafe Protect (1) and with RunSafe Protect (2a, 2b, and 2c).

- In 1, the HelloWorld application is copied into memory in the normal fashion for the operating system. The program's entry point is identified in the application's header information. Execution is then transferred to the function identified by the entry point, Function 3 in this case, located at 0x2a0.
- 2a shows the same program with RunSafe Protect included at compilation. The program has several important differences:
  - (1) A new function, RandoMain, has been added to the program. It is shown in the diagram as LFR_Main.
  - (2, 3) The application has been linked against liblfr.so. liblfr.so is RunSafe Protect's library built from a small code base (less than 20k lines of code) and loaded onto the system.
  - RandoMain is the program's new entry point.
  - Metadata has been added as a new section of the ELF file. This metadata includes:
    - Start point for each function.
    - Size of each function, in bytes.
    - Location and type of any relocations that need updating after relocation. These types of relocations are finite and testable per system.
- 2a references the first three steps of process execution, once the OS's normal process setup has completed:

1. Execution is turned over to the new RandoMain entry point, at 0xfe1. RandoMain is a stub function that jumps to the relocation routine in liblfr.so, RunSafe Protect's library.
2. The relocation routine then relocates individual functions in memory, using the metadata embedded in the file. Next, a customer-defined seed deterministically relocates the functions and places them in their new memory locations.
3. After relocating the code in memory, liblfr.so goes through all relocatable addresses and updates them to point to their new location in memory; e.g. references to Function 1 (F1) will be adjusted from 0x1a0 to 0x3a0. The diagram's 2b shows the application after all of the parts have been moved around. After the relocation, the protected software has the same memory footprint as the original software. RunSafe Protect does not make any changes to the data locations, only the function locations.

4. The final process execution step, shown in the diagram at 2c, is to transfer execution to the original entry point, Function 3. This will begin the normal, developer-intended functionality of the RunSafe Protect-protected program. Once execution transfers to the original entry point, RunSafe Protect is no longer active in the program's execution and there is no runtime performance overhead.



RunSafe Protect Qualification – Development Tool

RunSafe is certifying RunSafe Protect as a TQL-1 tool and airborne software at DAL-A. This will allow RunSafe Protect to be used as part of the development environment for any CSCI regardless of DAL (A-D).

TQL-1 is the most demanding level defined by DO-330. RunSafe Protect's developmental tool must be qualified at this level because the tool can have the most disastrous effect on flight safety. It could introduce a defect into DAL A airborne software that could cause the software to fail in flight causing a catastrophic failure of the aircraft, therefore meeting criteria 1.

Development per TQL-1 demands that all objectives (76) and activities delineated in DO-330 be satisfied, many of them with engineering independence.

Under the RunSafe strategy, the current version of RunSafe Protect is viewed as a working prototype. Because there is a demonstrated solution, no technical issues are anticipated. Development artifacts that are available will be used, but they are only used if appropriate. If the artifacts do not meet the rigorous expectations of DO-330, then the artifact will be recreated per DO-330, using current data as a guide.

Because DO-330 is more systematic and rigorous, it is possiblethat gaps in the current implementation may be discovered. These gaps will be addressed during the TQL-1 certification of RunSafe Protect.

The following expectations of DO-330 will be followed rigorously:

- Development plans defining the approach used to certify RunSafe Protect will be created.
- Plans and standards will be vetted and approved, then followed diligently.
- Work products needed to qualify RunSafe Protect to TQL-1 will be produced, verified, and controlled.
  - Tool and User Operational Requirements will be defined.
  - Design data will be captured. All required data and control flow analyses will be accomplished and recorded. Assumptions/restrictions on the use of RunSafe Protect will be identified and captured for later use (e.g., if RunSafe Protect does not address certain language constructs or assumes a specific CSCI development environment or target environment).
  - Implementation activities will translate the defined Low-Level Requirements (LLRs) into code.

- Developed code will be robustly tested to show that it satisfies all requirements, does not include dead code, and meets all DO-330 structural coverage objectives.
- Engineering independence will be used to verify all work products.
- A Conformity Review will be held/ recorded to show that all data and work products were produced as planned, all problem reports have been addressed, and the RunSafe Protect tool can be consistently produced for delivery to a CSCI developer.
- Configuration Management records will be captured to show that the integral process was consistently applied.
- Process Assurance records will be captured to show that approved plans were consistently followed.

Since RunSafe Protect is intended to satisfy a variety of development environments, it is anticipated that there will be some differences in RunSafe Protect versions. These will be addressed by configuration parameters.

If a tool used in CSCI development needs to be qualified, the CSCI developer must provide a Tool Qualification Plan (TQP) that shows that the tool meets the required Tool Qualification Level (TQL) and works as expected in the CSCI developer's environment.

Data demonstrating the rigorous TQL- 1 development will be assembled into a qualification package for use by a CSCI developer.

The RunSafe Protect tool qualification package will provide draft materials to be tailored by the CSCI developer. This will include data proving the development pedigree of RunSafe Protect described above. It also will include draft materials to be reviewed/accepted by the CSCI developer including Tool Operational Requirements (TOR), a Tool Qualification Plan (TQP), and a set of verification test cases that show RunSafe Protect satisfies the TOR in the CSCI developer's environment.

It is anticipated that draft work products will be finalized by the CSCI developer and placed under developer Configuration Management along with the other material in the RunSafe Protect qualification package.

Any issues encountered during the the application of RunSafe Protect to the CSCI developer's environment or impacts to RunSafe provided materials are to be elevated to RunSafe for resolution.

As with any other tool requiring qualification, the CSCI developer is responsible for using the tool correctly, properly controlling it, and gaining approval of the TQP. Qualification also requires that tools are used correctly. This is assured by CSCI developer quality assurance.

RunSafe Protect Certification – Airborne Software

Under DO-178C, the CSCI developer is responsible for creating a CSCI that meets allocated system requirements. The CSCI developer proposes and gains approval of the approach they will follow to satisfy DO-178C objectives (vary by DAL) in their Plan for Software Aspects of Certification (PSAC). When developing a CSCI, DO-178C calls for:

- Comprehensive detailed planning of the certification liaison, software quality assurance, configuration management, development, and verification.
- Complete definition of the high-level requirements the CSCI must satisfy and the low-level requirements that derive from those high-level requirements.
- Confirmation that all code in the CSCI has a solid rationale for being there.
- Consistent application of standards for requirements, design, and code.
- Verification of work products and systematic correction of issues identified.
- Elaboration of Parameter Data Items and confirmation that the CSCI functions correctly for normal and abnormal values of its elements.
- Qualification of any tools where the output is not verified manually.

RunSafe Protect is novel in that it alters the image of the CSCI's executable. Alterations invalidate the formal test ("run for score") that shows the CSCI meets DO-178C expectations (100% requirements coverage, structural coverage requirements, and evidence there is no unnecessary code). Therefore, the "Run for Score" testing must be performed AFTER RunSafe Protect has been applied.

RunSafe Protect also adds functionality/features to the CSCI under development. Since DO- 178C mandates that there must be a trace from system requirements to software High- Level Requirements (HLRs) to software LLRs and to test cases/procedures that prove these requirements were satisfied, the RunSafe Protect qualification package will include proposed materials to be integrated in the CSCI developer's data. This includes:

- A proposed system requirement that calls for cybersecurity attacks to be mitigated.
- A proposed set of Software HLRs defining the features the CSCI implementation will have (e.g., relocatability). The Software HLRs must be traced to the system requirement to mitigate cyberattacks.
- A proposed set of requirements, design, and coding standards for code alterations.

- A proposed set of Software LLRs that delineate the features that must be added to each CSCI unit. These LLRs must be traced to both the units affected and the Software HLRs.
- Test cases that show altered code works as expected (to relocate functions when loaded).
- Trace data.

It is expected that the seed value generated for each load, will be treated as a Parameter Data Item (PDI) under DO- 178C. The qualification package will include draft materials to be integrated into the CSCI developer's material. This material will provide a proposed CSCI plan and development artifact changes to address the altered/added code resulting from RunSafe Protect use.

As indicated above, RunSafe Protect takes a set of source code as its input and alters it so that it will relocate itself in its allocated memory space when loaded. The relocation is based on a seed. The seed (a 64-bit string) can be any value from all 0's to all 1's. Relocation is deterministic for a given seed value, but since it is randomly selected, the likelihood of identical values is small.

RunSafe Protect Certification – Airworthiness Security

The focus of DO-326 and DO-356 is an integrated approach to preventing and mitigating cyberattacks. A strategy cannot be developed without understanding the system and the environment in which it will be used and maintained. RunSafe Protect would be a part of that security strategy.

DO-326/356 is structured around a set of principles. RunSafe helps to satisfy two of those 14 principles:

- **Principle 1 - Defense in Depth:** The idea behind Defense in Depth is that if one protection fails, the second will hopefully be effective. Because RunSafe surrounds the entire product CSCI with its protective shield, it can form one layer of the Defense in Depth strategy. This allows the CSCI developer to focus attention on project's peculiar defense strategies.
- **Principle 2 - Ease of Maintenance:** RunSafe makes maintenance of the CSCI and its security features easier by eliminating maintenance of the RunSafe tool. As a COTS TQL-1 tool, RunSafe Protect maintenance responsibilities are not assigned to the CSCI developer, but to RunSafe itself. RunSafe Protect is deployed with effective tool training, support, and documentation. RunSafe Protect is designed to be integrated into the software build process. There is no action on the part of the CSCI developer to alter or configure RunSafe Protect.

RunSafe will provide documentation to satisfy objectives and activities enumerated by DO-326/356. The information can be extracted to merge into CSCI developer format, or the shell documents can become an early draft of DO 326/356 required documents, thus giving the CSCI developer a model to follow.

Shell documentation that is provided aligns with deliverables expected by DO-326/356. This includes:

- Plan for Security Aspects of Certification
- Plan for Security Aspects of Certification Summary
- Aircraft/System Security Risk Assessment
- Allocated System, High Level Software, Low Level Software, and Derived requirements
- Tool Qualification Plan, standards, test cases/ procedures, validation results, and verification results (Review, Analysis, Test) and supporting trace data
- Audit and CM records
  - These are particular to RunSafe Protect's implementation and qualification. However, the package forms a model that can be extended to other CSCIs and CSCI functionality, because it provides a comprehensive (though narrowly focused) example of how each of the objectives called for by DO-326/356 is satisfied.
- Operational guidance

RunSafe as an organization, is committed to helping CSCI developers successfully apply RunSafe Protect in their environment and address issues should they arise.

Doug Britton, EVP, RunSafe Security
Doug@RunSafeSecurity.com, 571-250-5941

Kathryn Fejer, Senior Software Engineer, RunSafe Security
Katie@RunSafeSecurity.com, 224-639-2795

Kenneth Hebert, Ph.D. CSM, AFuzion, Technical Director, Process Manager, Senior Trainer
Kenneth.Hebert@AFuzion.com, 505-226-8181

Jonathan Lynch, AFuzion, FAA DER & Commercial Pilot/Instructor
Jon.Lynch@AFuzion.com, 505-205-9800

RUNSAFE SECURITY

ABOUT RUNSAFE SECURITY, INC.

RunSafe Security is the pioneer of a unique cyberhardening technology designed to disrupt attackers and protect vulnerable embedded systems and devices. With the ability to make each device functionally identical but logically unique, RunSafe Security renders threats inert by eliminating attack vectors, significantly reducing vulnerabilities, and denying malware the uniformity required to propagate. Based in McLean, Virginia, with an office in Huntsville, Alabama, RunSafe Security's customers span the Industrial Internet of Things (IIoT), critical infrastructure, automotive, and national security industries.

www.RunSafeSecurity.com

571.441.5076

Sales@RunSafeSecurity.com