Deja vu Security™ - Avionics Testing

Whitepaper

Avionics equipment services a wide range of functions in aircraft, satellites, and spacecraft. The equipment ranges from general-purpose flight computers to highly specific instruments and controls. At their core, every avionics system is an embedded computer that serves flight enabling and safety critical functions for communications, navigation, monitoring, weather management, and aircraft control. The testing of avionics equipment creates many unique challenges and Déjà vu Security applies a variety of techniques to fully enumerate the risk inherent in an avionics systems and embedded devices. These techniques include but are not limited to Fuzzing Avionics, Embedded System Testing, Manual Interconnect Testing, Redundancy Testing, and Flight Safety Effect Enumeration.

Fuzzing Avionics

Avionics equipment is implemented from a cornucopia of embedded processors, real time operating systems, software, and custom firmware. This firmware implements various functions inside an avionics system and is low level code that accesses hardware resources directly. The function of firmware in avionics systems varies widely, depending on the Line Replaceable Unit (LRU) or component in question. The one constant of the firmware is the ability to consume and process data, making avionics equipment interesting targets for fuzzing.

The interfaces used to communicate with firmware also vary widely, ranging from standard low-level bus protocols such as 429, 485, 1553, 1394, WiFi, SPI, I2C, USB, proprietary hardware interfaces and custom protocols. As such, this scenario is not one solution fits all. When required, Déjà vu Security creates a custom solution for communicating with firmware internal or external to the device.

The goal of fuzzing avionics is to identify security and reliability issues in systems and protocols automatically. Fuzzing utilizes the fact that machine time is cheap and can run hundreds of thousands or millions of test cases, enumerating faults automatically. Fuzzing uses known good data to seed messages to a fuzzing engine. The fuzzer engine intelligently mutates the data by changing the lengths, values, types, and checksums in an intelligent and automated manner. The mutated data is sent to a monitored avionics system. Shells, serial connections, debugging terminals, debuggers, and JTAG can be used to detect a fault in the system. The mutated data and messages can target various COTS and ARINC protocols, such as ACARS, 429, TCP, DHCP, TELNET, WiFi, and many others.

Fuzzing targets various onboard networked systems and embedded controllers to cause reproducible faults such as:

- Memory Corruption
- Code Execution
- System Denial of Service
- Safety Effects
- System Reboot
- Firmware Corruption
- Cascading System Failures

Once faults are collected, they are bucketed to remove redundant issues and reviewed for severity before reporting.

Embedded System Testing

Embedded System Testing attempts to identify, classify, and close risk in the design and implementation of the avionics system. Embedded System Testing is performed on the running LRU firmware and software. It utilizes software-based and hardware-based attacks against the processor, firmware, and running services. It can include the review of each component for logical issues, backdoors, weak access controls, authentication issues, memory corruption, firmware loading issues, and enumeration of hard-coded secrets.

Often times during the embedded system review process, a tester will identify systemic issues such as reuse of vulnerable libraries and poor coding practices that can cause reliability and security issues. Additionally, testing attempts to enumerate the security maturity of the software and operating system.

Additionally, as LRUs are designed to be field upgraded, embedded system testing focuses of the data load process and verification of patches and updates. The tester identifies data loading in various system states such as maintenance vs flight time. They inspect the protection mechanisms deployed for ensuring against malicious firmware update via a maintenance laptop, USB stick, or WiFi. If a 615A Data Loader or other aircraft transfer mechanisms are present, they are inspected for verification post load.

Manual and Automated Interconnect Testing

It has been said an aircraft is a large number of parts flying in formation. As such, no LRU is an island. The goal of system interconnect testing is to identify issues in component to component interactions. Identify sensitive or critical data passed across busses to emulate and test messages. Send messages via various onboard busses: 429, Ethernet, I2C, etc. The tester then intelligently manipulates traffic until an intended effect is observed. The tester may attempt manually or use a tool in an attempt to:

- Create network congestion that causes critical messages to be dropped or corrupted
- Send malicious messages to crash systems and services
- Send tailored messages to manipulate other embedded onboard systems
- Send exploit messages to take full control of a system

Redundancy Testing

Onboard systems deploy multi redundancy and switch to a backup system in the event of a failure. A common design decision is to make these systems identical; and a software defect in one affects them all. Targeting the redundancy requires a corruption message found via fuzzing, reversing, and the proper timing of sending the messages for each bus. Redundancy testing enumerates the systems' overall resilience and cascading effects to a well-positioned attacker in the event they can cause a temporary or permanent denial of service to a system of the aircraft.

Flight Safety Effect Enumeration

Much like corporate network systems, attackers target low hanging fruit and work up to more important systems. The goal of flight safety effect enumeration is to help prioritize system defense efforts by understanding an attacker's sphere of influence. Many onboard systems have a low safety classification of D or E. However, these systems can often communicate and interact with system that have a higher level, A - C. Due to their interconnectivity, the control of a lower level system (D or E) can allow an attacker to communicate or influence a higher level system (A, B, C). Flight safety effect enumeration creates attack trees and performs highly targeted testing to identify potential cascading safety effects. If an attacker has control of an edge or a central system, how much additional influence can they cause? Starting at E level systems and working one system at a time towards A to C level with the ultimate goal of being able to influence or confuse a system to potentially causing a safety event.

Contact Us

Visit our website	Talk to a representative	Email us
dejavusecurity.com	Call toll free 1 (855) 333 5288	sales@dejavusecurity.com