Section: CS

Lowering Avionics Bus Trust: Moving ARINC 429 Bus Architecture Towards Zero Trust

Matthew Preston

Problem Statement:

ARINC 429 bus architecture, like most bus architectures is insecure. Any command on the bus will be inherently trusted and executed by receiver LRUs. This means that compromising legacy systems and software can potentially cause catastrophic effects. Since 2022, the United States government has stated intent to securing its assets with Zero Trust principles. Therefore, moving ARINC 429 to zero trust without having to design and implement a whole new architecture or standard would help closer align the cybersecurity posture of critical infrastructure to this intent.

Solution Statement:

This research will simulate an ARINC 429 bus (as a digital twin) as well as implement an ARINC 429 traffic/rules-based intrusion detection system to engineer a defense system for ARINC 429 that helps move the ARINC 429 bus architecture towards zero trust. It will have the following deliverables:

- A simulation of an ARINC 429 bus architecture of and airplane
- A rules-based IDS that can log and flag ARINC 429 words.

Completed Tasks (Last 2 Week):

- I drafted my project proposal.
- I revised and refined my project proposal.
- I did the following research (all properly cited in the references section):
    - Research on understanding how the ARINC 429 protocol works.
        - "ARINC-429 RX Implementation in Labview FPGA." – This helped me see some of the ways a real reciever ARINC 429 LRU is programmed and works.
        - "PyARINC429" on GitHub – this is a utility I plan to use to craft ARINC 429 words.
        - "ARINC-429 with Cyber and Wirefault Protection" – this is another example of an LRU for ARINC 429 and how it is designed in the real world.
        - "ARINC 429 Specification Tutorial" – this is a document that outlines the ARINC 429 specification, concisely.
        - "ARINC-429 tutorial: A Step-by-Step Guide" – this is another guide to the ARINC 429 specification.
        - "ARINC-429 Tutorial & Reference" – this is also a guide to ARINC 429, and has an example of another LRU working with a scheduler for words.
        - "ARINC-429 Tutorial: A Step-by-Step Guide" – Concise tutorial for ARINC-429.
        - "Avionics Databus Tutorials" – this is a very simple overview of many data bus architectures and helped me narrow down my focus from proposal 1.
        - "Interfacing Electronic Circuits to Arduinos" – this is a tutorial on building a homebrew ARINC 429 TX LRU board. While I won't be able replicate it because he got a free chip from Device Engineer Inc., it does help me understand more on how an LRU would work on the data bus.
        - "ARINC Specification 429 Part 1-17: Mark 33 – Digital Information Transfer System (DITS)" – this document matches hex to English on many things

encapsulated for the ARINC 429 word, so I can make my simulated data accurate.

- Research on various bus architecture security solutions, to see what has been developed.
    - “Episode 64: Zero-Trust Cybersecurity for Vehicles.” – this is an interview with a technician from the Southwest Research Institute, a federally funded research and development center, on how they implemented zero trust for the CAN bus.
    - “Understanding Cyber Attacks on MIL-STD-1553 Buses” – this lists details for 1553 attacks.
    - “1553 Network and Cybersecurity Testing” – this paper explores test/design of 1553 buses with cybersecurity.
    - “Why You Need to Secure Your 1553 MIL-STD Bus and the Five Things You Must Have in Your Solution” – this is a short article that says for better cybersecurity for 1553 bus, you need to have real-time monitoring, active mitigation, user-defined threat definition, simple integration, and on platform and test and maintenance protections. This article helps guide a thought process for similar solutions for other bus architectures but is somewhat vague.
    - “Exploiting the MIL-STD-1553 avionic data bus with an active cyber device” – this is a research paper detailing a cyber-attack on MIL-STD-1553.
- Research on understanding zero trust cybersecurity
    - “Executive Order on Improving the Nation’s Cybersecurity” – this is an executive order from the POTUS in which, he talks more about his vision for what the US’ cybersecurity posture should be.
    - “Zero Trust Architecture.” – the definition of Zero Trust in cybersecurity.
    - “Executive Order on Improving the Nation’s Cybersecurity” – the more strategic level view of what the executive branch wants for zero trust in their systems.
- Research on any cybersecurity that has been done on ARINC 429, and if any of those help ARINC 429 implement zero trust principles/tenets.
    - “Hardware Fingerprinting for the ARINC 429 Avionic Bus” – As far as I’m aware this is the only ARINC 429 cyber security research I could find. Additionally, they claim to be the first ARINC 429 research paper.

- I started the simulation of the ARINC 429 bus by using PyARINC429 to build one component of the bus, the electronic engine control in python.
- I started the flight simulation software, by adding math for various forces and orientation on the plane. It is still rough. It also has a way to plot the plane’s positional coordinates in 3D using matplotlib.

Tasks for the Next Project Report:

- Finalizing the flight simulator to have positional data to feed into LRUs in the future.
- Create the code that will be the Flight Management Computer LRU simulation.
- Create the code that will be the weight and balance system LRU simulation.
- Create the code that will be the GPS LRU simulation.
- Create the code that will be the radio management system LRU simulation.
- Test data connection hookup between the EEC and W&BS LRUs and the flight simulator.

Questions I have or Issues I’m running into:

I was wondering if my project is too ambitious in its timeline or number of deliverables. Most of the coding happens on the front end in short succession. I was also wondering if any of the parts of the simulation seem frivolous, vague or unobtainable.

Methodology Paragraph Summary:

The process I will be employing is:

1. Plan structure of next component I need to build
2. Build component based on plan
3. Test component functionality by making test cases
4. Rework component based on test
5. Test component working with other components

Repeat until each component is finished

For evaluation, I will be using a data-driven approach:

1. Gather data during evaluation phase.
2. Use data to see if the solution is robust, impacts functionality, and is accurate.

Timeline:

Week #	**Description of Task**	**Status**
Week 1 Monday, 13 May 2024 – Sunday 19 May 2024	Research topic by reading articles. Focus the research on understanding how the ARINC 429 protocol works.	Completed
	Research topic by reading articles. Focus research on various bus architecture security solutions, to see what has been developed. This research has helped narrow down the architecture from the list of ARINC 429, CAN bus, 1553 bus, and ARINC 629 to just ARINC 429.	Completed
	Research topic by reading articles. Focus the research on understanding zero trust cybersecurity, and if any zero trust implementation has been done for the above architectures. This helps inform a template for ARINC 429	Completed
	Draft Initial Proposal	Completed
Week 2 Monday, 20 May 2024 – Sunday 26 May 2024	Finalize Proposal	Completed
	Research topic by reading articles. Focus the research on any cybersecurity that has been done on ARINC 429, and if any of those help ARINC 429 implement zero trust principles/tenets.	Completed
	Simulate a receive LRU, the electronic engine control, that would be taking ARINC 429	In Progress

	commands and outputting actuations (for simulator).	
	Start flight simulation software that should interface with LRU outputs. It should at this point just be an outline of the following functionality: - A tick/step-based time system that calculates the airplanes positional data from the last tick. Given the following attributes: altitude, x/y position, roll, yaw, pitch, forward velocity, and jet engine thrust, it should calculate the next x, y, altitude positional data. - Start at position 0, 0, 500 - Plot continuously the plan's positional data - A way to take in data from the actuators LRUs (jet engines, balancing LRUs for fins, etc) and translate that to the calculations. An outline here consists of creating the python file, putting in the math equations for steps/ticks for this, outlining a function that should take in data from an actuator and setting up the plotting given 3-d coordinates. This will be the codebase to test the ARINC429 simulation actuators from.	In Progress
Week 3 Monday, 27 May 2024 – Sunday 2 June 2024	Simulate a transmitter LRU, the Flight Management Computer. It should have the following features: - Multiple TX channels - Multiple RX channels - Basic flight functionality software that generates ARINC 429 words based on desired direction to go	Not started
	Finalize functionality for the flight simulator above.	Not started
	Simulate a receiver LRU that will be the weight and balance system on the plane. It should output actuator data for the simulation.	Not started

Week 4 Monday, 3 June 2024 – Sunday 9 June 2024	Simulate a transmitter LRU, the GPS that reports actual positional x/y/altitude data. It should get this back from the simulator above.	Not started
	Simulate a receiver LRU, the radio management system.	Not started
	Test interaction and hookup between the electric engine control LRU, and weight and balance system to the simulator to make sure they can input and influence the motion of the plane.	Not started
Week 5 Monday, 10 June 2024 – Sunday 16 June 2024	Simulate a transmitter LRU, the ADIRU, and test its hookup to the flight simulator to make sure it can input data correctly.	Not started
	Create vulnerable program for FMC and a simple buffer overflow / rop hack for it to gain system access to the FMC.	Not started
	Create the orange ARINC429 bus.	Not started
	Create the blue ARINC429 bus.	Not started
Week 6 Monday, 17 June 2024 – Sunday 23 June 2024	Create final report outline.	Not Started
	Create the purple/channel B ARINC 429 bus	Not Started
	Create the green/channel A ARINC 429 bus	Not Started
	Add functionality to the attack above that when system access is gained on the FMC, start transmitting ARINC 429 words from the FMC to the EECs that pitch the plane into a downward trajectory.	Not Started
	Start the rules-based IDS system. Implement the functionality to it: - Create syntax for IDS rules - Create functionality for logging bus traffic	Not Started
	Start logging ARINC 429 traffic for data collection.	Not Started
Week 7 Monday, 24 June 2024 – Sunday 30 June 2024	Create first draft of final report	Not Started
	Add the following functionality to the IDS: - Ability to generate alarms based on transmission ID - Ability to generate alarms based on parity & parity correctness	Not Started

	- Ability to generate alarms based on sign/statis matrix -> normal operation (N/S, E/W), functional test, failure warning, no computed data - Ability to generate alarms based on data (hopefully also granulate that based on flight directional data, etc. Combine with previous words to see if there is a suspicious word). - Ability to generate alarms based on source/destination field - Ability to generate alarms based on label (data type)	
Week 8 Monday, 1 July 2024 – Sunday 7 July 2024	Revise the first draft of the final report into second draft.	Not Started
	Create a mode in the IDS that can reset the bus upon any of the alarms from IDS	Not Started
Week 9 Monday, 8 July 2024 – Sunday 14 July 2024	Create final demo presentation. It should include a demo of the simulation, attack, and defense implementations working with the attack.	Not Started
Week 10 Monday, 15 July 2024 – Sunday 21 July 2024	Create / post final demo video.	Not Started
Week 11 Monday, 22 July 2024 – Thursday 25 July 2024	Finish project final report from second draft.	Not Started

Evaluation:

Evaluation plan:

Test and report on the following.

1. For IDS: how fast can it run?
2. How many rules can the IDS handle?
3. What percentage of ARINC 429 words can the IDS categorize correctly as malicious?

Further evaluation shall be fleshed out in future progress reports.

Report Outline:

To be delivered by progress report 5.

References:

R. Vincent. "ARINC-429 RX Implementation in Labview FPGA." *Arinc-429 RX Implementation in LabVIEW FPGA*, NI Community, 28 Nov. 2023, https://forums.ni.com/t5/Example-Code/Arinc-429-Rx-Implementation-in-LabVIEW-FPGA/ta-p/3507624

aeroneous. "PyARINC429." *Discover PyARINC429, a simple Python module for encoding and decoding ARINC 429 digital information.* 17 Jul. 2018, https://github.com/aeroneous/PyARINC429

Peña, Lisa; and Shipman, Maggie. "Episode 64: Zero-Trust Cybersecurity for Vehicles." *Technology Today Podcast*, Southwest Research Institute, Feb. 2024, https://www.swri.org/podcast/ep64

"ARINC-429 with Cyber and Wirefault Protection" *ARINC-429 Solutions*. Sital Technology, https://sitaltech.com/arinc-429/

"Understanding Cyber Attacks on MIL-STD-1553 Buses" Sital Technology, https://sitaltech.com/understanding-cyber-attacks-on-mil-std-1553-buses/

"1553 Network and Cybersecurity Testing." Alta Data Technologies LLC, 19 Jan. 2021, https://www.altadt.com/wp-content/uploads/dlm_uploads/2020/10/1553-Network-and-Cybersecurity-Testing.pdf
Tilman, Bill. "Why You Need to Secure Your 1553 MIL-STD Bus and the Five Things You Must Have in Your Solution." Abaco Systems, 14 Dec. 2021, Original Link: https://abaco.com/blog/why-you-need-secure-your-1553-mil-std-bus-and-five-things-you-must-have-your-solution, Accessible Here: https://web.archive.org/web/20240223161240/https://abaco.com/blog/why-you-need-secure-your-1553-mil-std-bus-and-five-things-you-must-have-your-solution

Waldmann, B. "ARINC 429 Specification Tutorial." *Avionics Databus Solutions*, Version 2.2, AIM Worldwide, Jul. 2019, https://www.aim-online.com/wp-content/uploads/2019/07/aim-tutorial-oview429-190712-u.pdf, https://www.aim-online.com/products-overview/tutorials/arinc-429-tutorial/

"ARINC-429 tutorial: A Step-by-Step Guide." KIMDU Technologies, 26 Jun. 2023, https://kimdu.com/arinc-429-tutorial-a-step-by-step-guide/

"ARINC-429 Tutorial & Reference" *Understanding ARINC-429*, United Electronic Industries/AMETEK, https://www.ueidaq.com/arinc-429-tutorial-reference-guide

Biden, Joesph R. Jr. "Executive Order on Improving the Nation's Cybersecurity." *Briefing Room, Presidential Actions*, The White House, 12 May 2021, https://www.whitehouse.gov/briefing-room/presidential-actions/2021/05/12/executive-order-on-improving-the-nations-cybersecurity/

Rose, Scott; Borchert, Oliver; Mitchell, Stu; and Connelly, Sean. "Zero Trust Architecture." *NIST Special Publication 800-207*, National Institue of Standards and Technology, U.S. Department of Commerce, Aug. 2020, https://doi.org/10.6028/NIST.SP.800-207

Young, Shalanda D. "Moving the U.S. Government Toward Zero Trust Cybersecurity Principles" *MEMORANDUM FOR THE HEADS OF EXECUTIVE DEPARTMENTS AND AGENCIES*, Version M-22-09, Executive Office of the President; Office of Management and Budget, 26 Jan. 2022, https://whitehouse.gov/wp-content/uploads/2022/01/M-22-09.pdf

"Avionics Databus Tutorials." *Ballard Technology*, Astronics AES, https://www.astronics.com/avionics-databus-tutorials

maewert. "Interfacing Electronic Circuits to Arduinos." *Circuits; Arduino*, Autodesk Instructables, https://www.instructables.com/Interfacing-Electronic-Circuits-to-Arduinos/

Airlines Electronic Engineering Committee. "ARINC Specification 429 Part 1-17: Mark 33 – Digital Information Transfer System (DITS)." *ARINC Document*, Aeronautical Radio Inc. 17 May 2004, Original Link: https://read.pudn.com/downloads111/ebook/462196/429P1-17_Errata1.pdf , Accessible here: https://web.archive.org/web/20201013031536/https://read.pudn.com/downloads111/ebook/462196/429P1-17_Errata1.pdf

D. De Santo, C.S. Malavenda, S.P. Romano, C. Vecchio, "Exploiting the MIL-STD-1553 avionic data bus with an active cyber device." *Computers & Security*, Volume 100, 2021, 102097, ISSN 0167-4048, https://doi.org/10.1016/j.cose.2020.102097. (https://www.sciencedirect.com/science/article/pii/S0167404820303709)

Gilboa-Markevich, N., Wool, A. (2020). "Hardware Fingerprinting for the ARINC 429 Avionic Bus." In: Chen, L., Li, N., Liang, K., Schneider, S. (eds) Computer Security – ESORICS 2020. ESORICS 2020. Lecture Notes in Computer Science(), vol 12309. Springer, Cham. https://doi.org/10.1007/978-3-030-59013-0_3

Kiley, Patrick. "Investigating CAN Bus Network Integrity in Avionics Systems." Rapid7, 30 Jul. 2019, https://www.rapid7.com/research/report/investigating-can-bus-network-integrity-in-avionics-systems/

Appendix

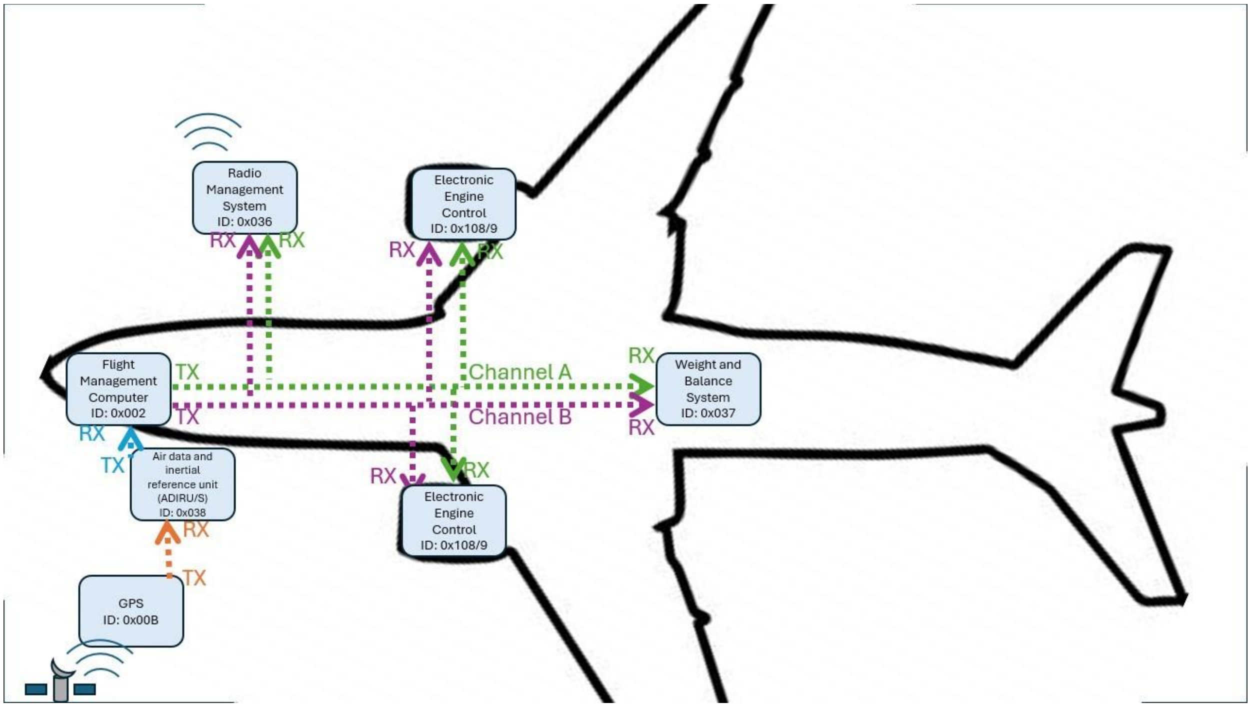
Radio Management System ID: 0x036
RX
RX
Electronic Engine Control ID: 0x108/9
RX
RX
Flight Management Computer ID: 0x002
TX
TX
RX
Channel A
Channel B
RX
Weight and Balance System ID: 0x037
RX
TX
Air data and inertial reference unit (ADIRU/S) ID: 0x038
RX
RX
RX
Electronic Engine Control ID: 0x108/9
TX
GPS ID: 0x00B

Figure 1: System Model

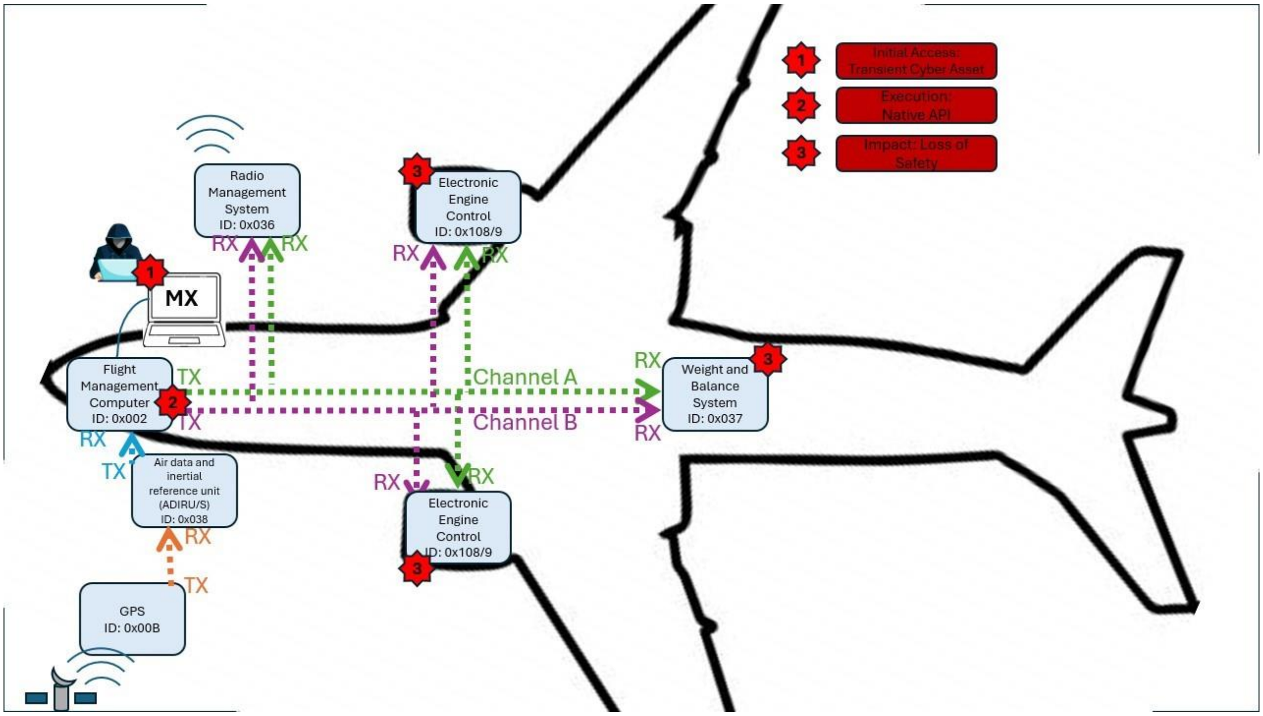
1
Initial Access: Transient Cyber Asset
2
Execution: Native API
3
Impact: Loss of Safety
Radio Management System ID: 0x036
RX
RX
3
Electronic Engine Control ID: 0x108/9
RX
RX
1
MX
Flight Management Computer ID: 0x002
2
TX
Channel A
TX
Channel B
RX
Weight and Balance System ID: 0x037
3
RX
RX
TX
Air data and inertial reference unit (ADIRU/S) ID: 0x038
RX
RX
RX
Electronic Engine Control ID: 0x108/9
3
TX
GPS ID: 0x00B

Figure 2: Threat Attack Model