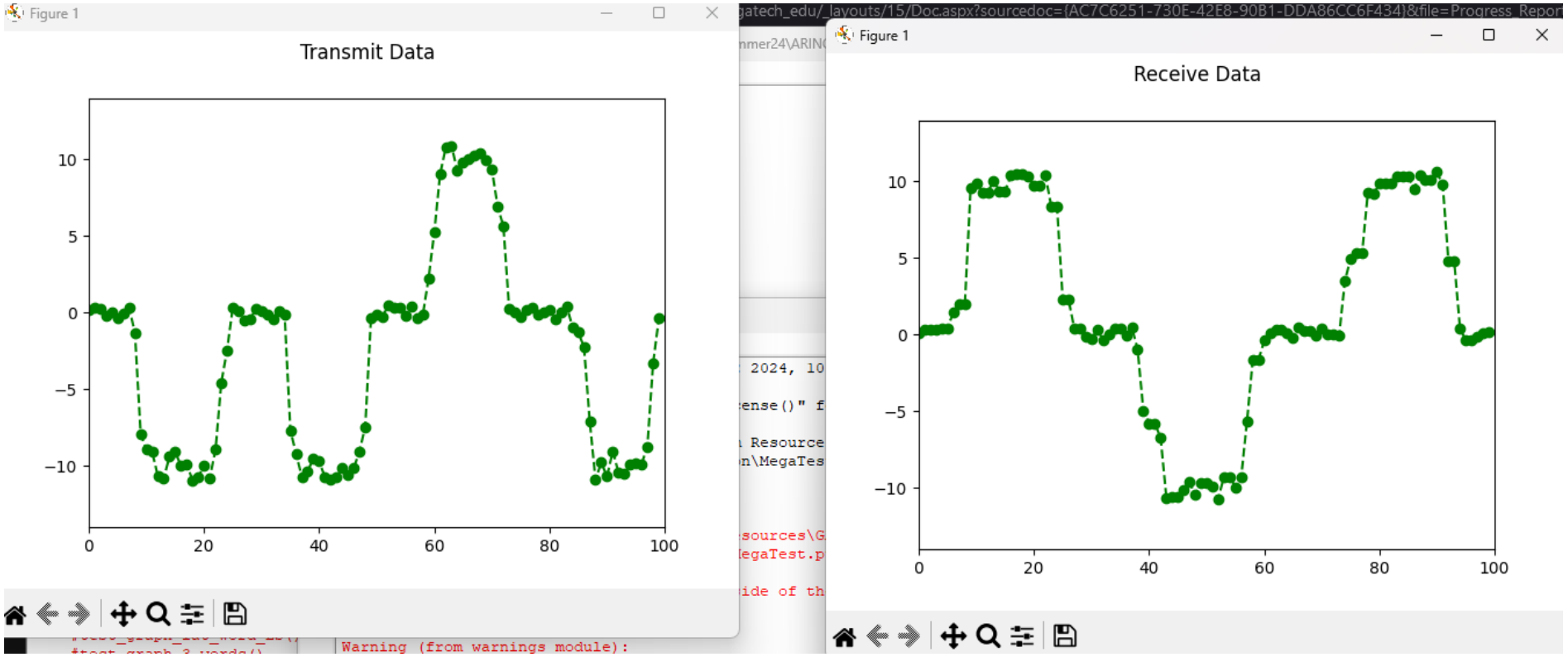
Lowering Avionics Bus Trust: Moving ARINC 429 Bus Architecture Towards Zero Trust.

Matthew Preston Final Presentation

Overview

- Problem Description
- Solution
- Demo
- Evaluations
- Limitations
- Possible Next Steps / Future Work



Problem Description

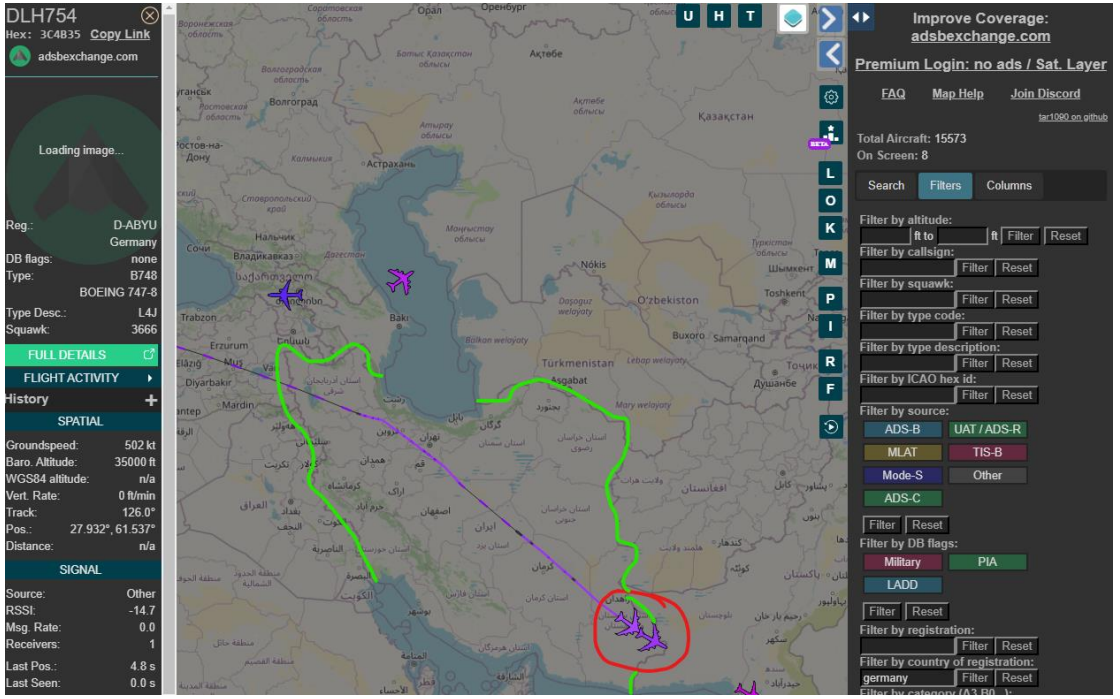
- ARINC 429 bus architecture is insecure
- Any command is trusted and executed by receiver LRUs
- Compromising legacy systems and software can be catastrophic

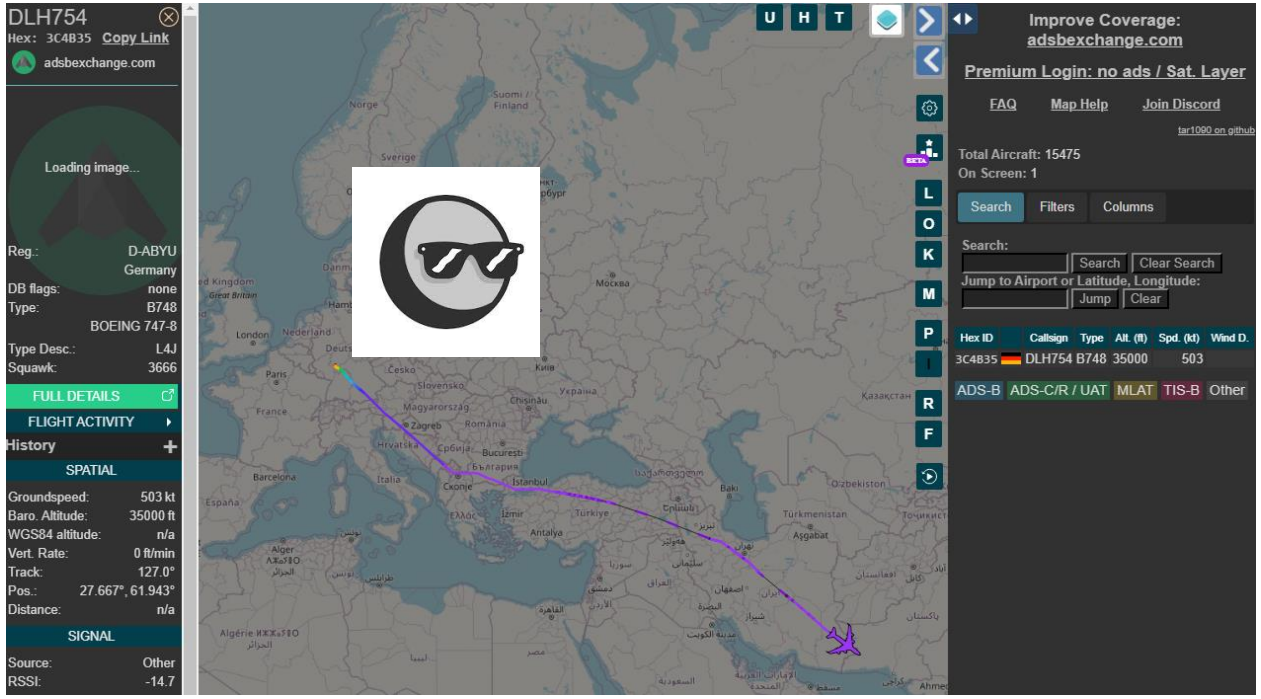
Problem Description

- ARINC 429 bus architecture is insecure
- Any command is trusted and executed by receiver LRUs
- Compromising legacy systems and software can be catastrophic

Problem Description

Attack Narrative – Person of Interest Capture





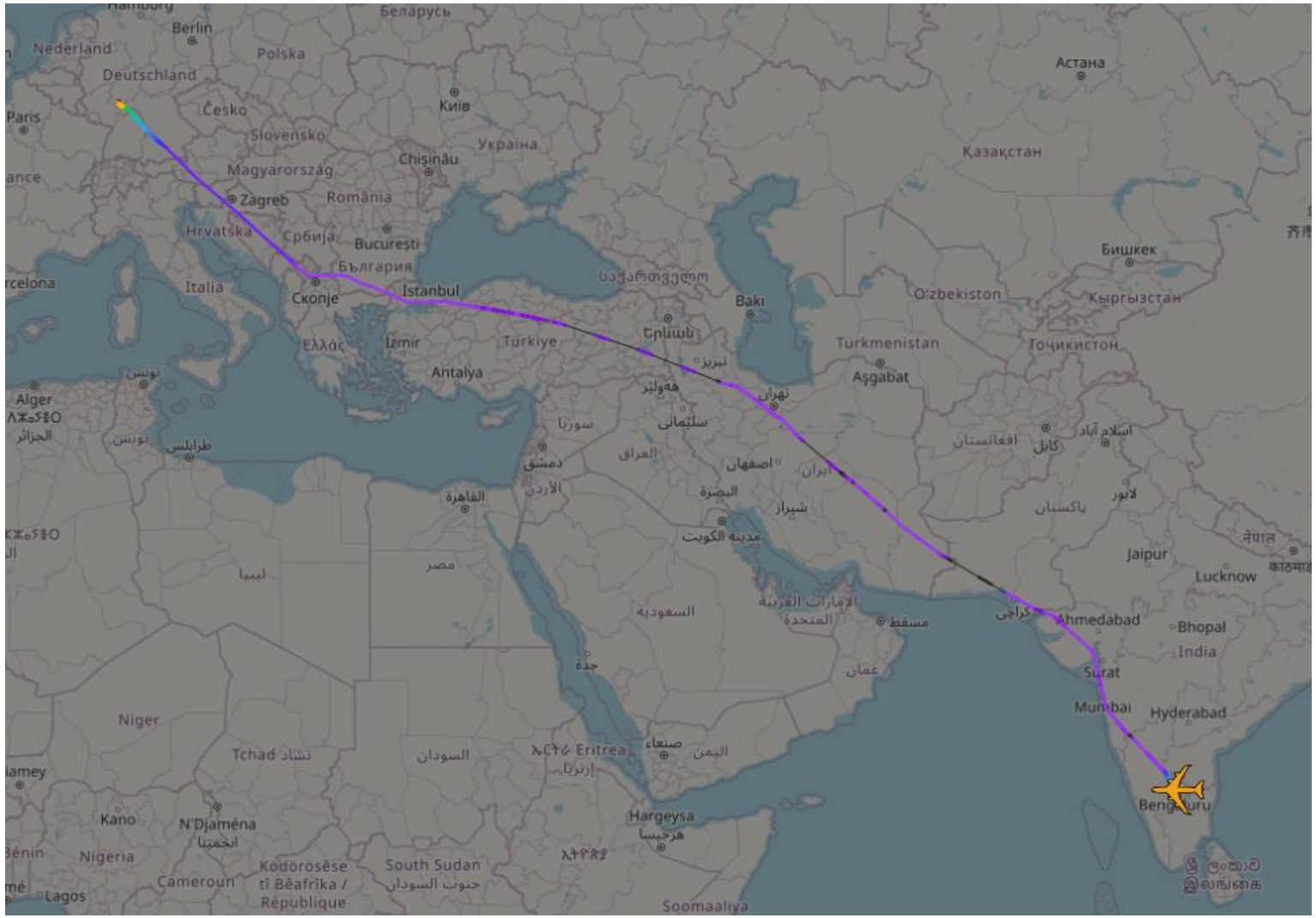
Source: ADS-B Exchange, Flight DLH754 (https://globe.adsbexchange.com/?icao=3c4b35) 1429 EST 2 July 2024
Flight from Frankfurt Germany to India.

Passenger of Interest outspoken against & refugee from Iranian Govt living in Germany.

Can the Iranian regime force the plane to land while over their airspace?

Problem Description

Attack Narrative – Full Flight



Problem Description
Attack Path

Pseudocode Idea:

```
if(lat, long in Friendly Airspace):
  for x in range(2):
    sleep(15 min)
    send(FMC.dive_word)
```

Quantas Flight 72

Mayday issued

Solution

- ARINC429 Simulator/ LRUs
- Attack PoC
- IDS

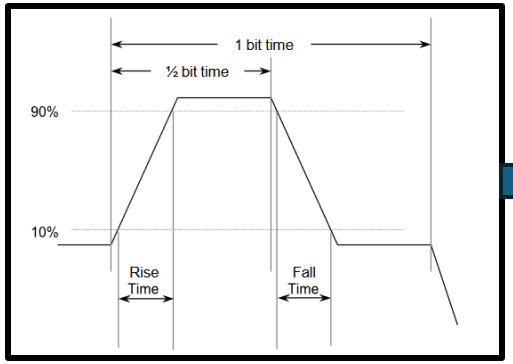
```
!Outfiles
alerts = C:/Users/mspre/Desktop/Practicum Resources/GATech_MS_Cybersecurity_Practicum_InfoSec_Summer24/ARINC429 Simulation/IDS_Rules_test_files/IDS_EVAL3_RULES_FILES/Alerts_Logs/Alerts
logs = C:/Users/mspre/Desktop/Practicum Resources/GATech_MS_Cybersecurity_Practicum_InfoSec_Summer24/ARINC429 Simulation/IDS_Rules_test_files/IDS_EVAL3_RULES_FILES/Alerts_Logs/Logs_EV

!Channels
Channel1: Global_Positioning_System -> ADIRS
Channel2: ADIRS -> Flight_Control_Computer

!SDI
# Do not identify any TX LRUs SDI here.
Channel2: ADIRS -> 00
Channel1: Flight_Control_Computer -> 11

!Rules
alert/log Channel2 0o310 ADIRS data:41.88331858083409 C "Longitude is 41.88331858083409 Degrees"
alert Channel1 0o311 ADIRS "GPS Longitude word sent."
log Channel2 0o015 ADIRS data:123 00 C "Wind Speed is 123 Knots"
alert Channel2 0o221 ADIRS data:70.45 "WARNING!!! ANGLE OF ATTACK DANGEROUS!"
# Should not pop there are no dangerous Indicated AoAs.
log Channel2 0o221 ADIRS data:1.010735034942627 time "Indicated Angle of Attack signals the plane is climbing in elevation."
log Channel2 ADIRS I "ADIRS is sending incorrect parity words."
alert Channel2 0o230 Encoding:BCD "BCD Words for label 0o230"
alert Channel1 0o066 Flight_Control_Computer bits[15:20)=1111 time "FMC Sending Spurious downward dive words?!!"
```

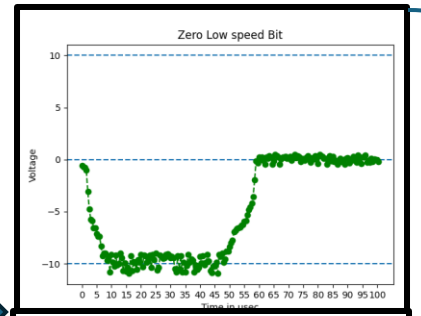
Example of IDS Rules

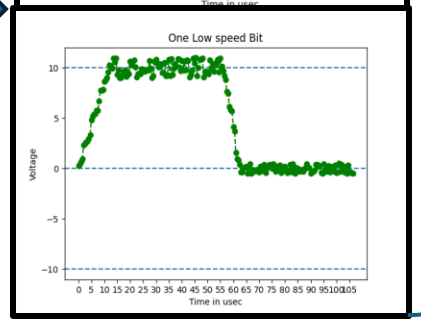
```
def alert_or_log(self, word: str):
    flag_this_tuple = False
    this_word_alerted_or_logged = False
    a_o_r = "None"
    with open(self.log_filepath, "a") as log_fd:
        with open(self.alert_filepath, "a") as alert_fd:
            for _tuple_ in self.rules:
                parity = _tuple_[3]
                time_notate = _tuple_[4]
                flag_this_tuple = False
                # Part 1 Check if you should flag this word.
                if(_tuple_[2] == "0"*31 and parity == None):
                    flag_this_tuple = True
                if (_tuple_[0].__contains__("alert") or _tuple_[0].__cor
                    #Check channel? -> done by caller
                    psuedo_word = word[:-1]
                    partity_calc = lru_txr()
                    correct_parity_bit = partity_calc.calc_parity(psuedo
                    #p_bitmask = _tuple_[2] + correct_parity_bit
                    bitmask = _tuple_[2]  #+ lru_txr.calc_parity(_tuple_

                    if (bitmask[0:10] == psuedo_word[0:10] and (int(bitm
                        if ((parity == True and word[-1] == correct_parity_bit)
                                or (parity == False and word[-1] != correct_parity_bit)):
                            # alert
                            flag_this_tuple = True
                        elif (parity == None):
                            # alert
                            flag_this_tuple = True
                # Part 2: If the word is flagged, the log it appropriately.
                if (flag_this_tuple and time_notate):
                    a_o_r = _tuple_[0]
                    this_word_alerted_or_logged = True
                    if (_tuple_[0].__contains__("alert")):
                        alert_fd.write(f"{ctime()}: Alert! {_tuple_[5]}\n")
                    if (_tuple_[0].__contains__("log")):
                        #input(_tuple_)
                        log_fd.write(f"{ctime()}: {word} {_tuple_[5]}\n")
                elif (flag_this_tuple and time_notate == False):
                    a_o_r = _tuple_[0]
                    this_word_alerted_or_logged = True
                    if (_tuple_[0].__contains__("alert")):
                        alert_fd.write(f"Alert! {_tuple_[5]}\n")
                    if (_tuple_[0].__contains__("log")):
                        #input(_tuple_)
                        log_fd.write(f"Logged word #{self.n}: {word} {_tuple_[5]}\n")
        alert_fd.close()
    log_fd.close()
    return((this_word_alerted_or_logged, a_o_r))
```

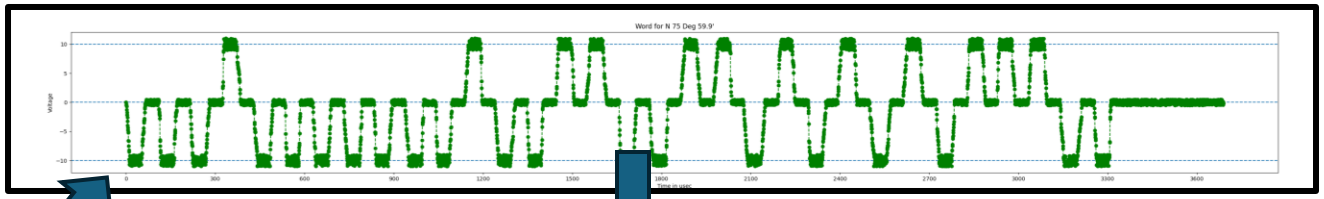
Snapshot from IDS

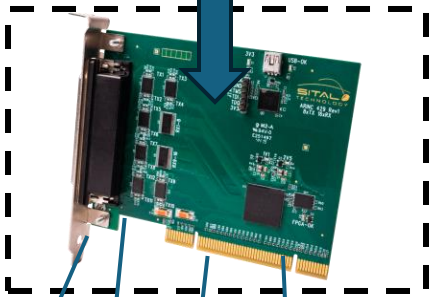
Demo


1 bit time
½ bit time
90%
10%
Rise Time
Fall Time

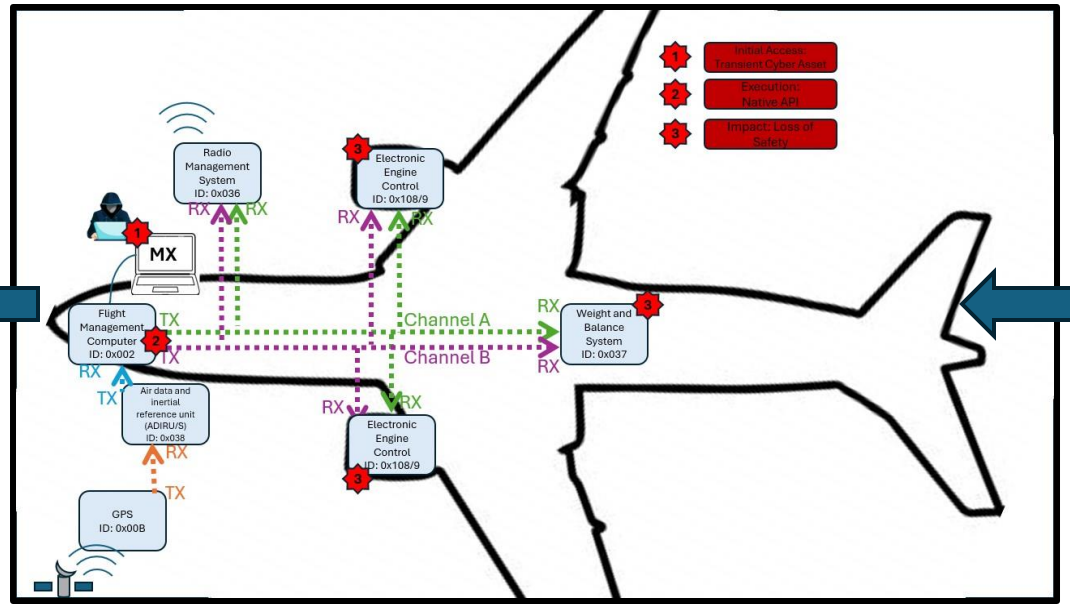

Zero Low speed Bit
Voltage
Time in usec

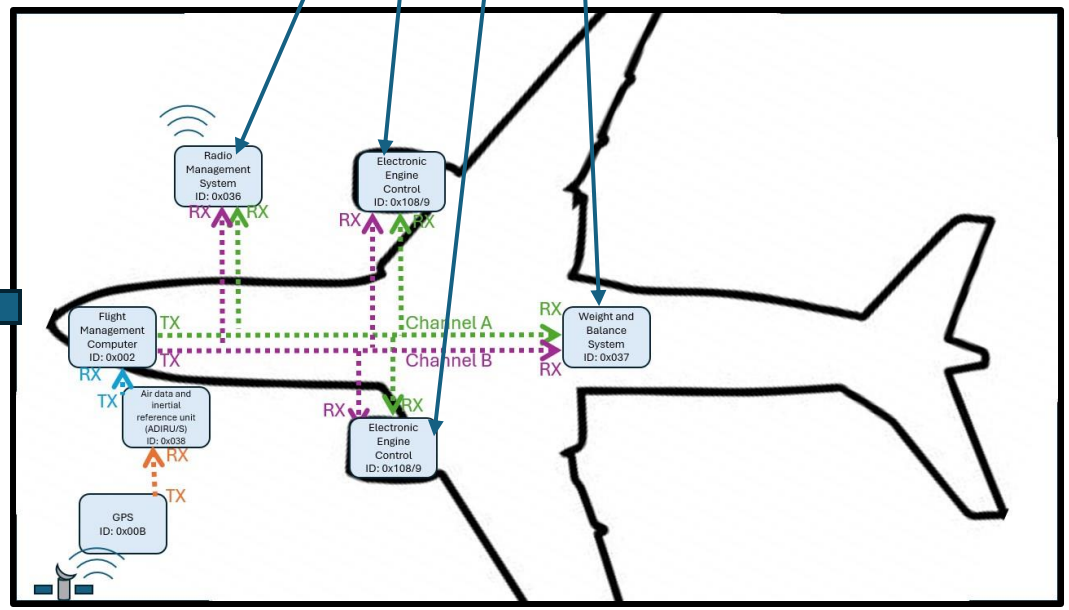

One Low speed Bit
Voltage
Time in usec





+IDS


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


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

Evaluations

1. MegaTest.py → Simulation Correctness
2. IDS_EVAL1.py → IDS Robustness
3. IDS_EVAL2.py → IDS Correctness
4. IDS_EVAL3.py → IDS Detection of Attackers
5. IDS_EVAL4.py → Time added by IDS

Evaluation 1:

MegaTest.py → Simulation Correctness

```
def test_rules_AllDataTypes():
    current_dir = getcwd()
    filename = current_dir + "\\IDS_Rules_test_files\\" + "rules_data_stress_test.txt"
    IDS_test_default = IDS(rules_file=filename)
    sdi = IDS_test_default.sdis
    print(sdi)
    rulez = IDS_test_default.rules
    assert(rulez == [('alert/log','Channel1','00010000110000000000010000000000',True,False,'Longitude is -40.0 Degrees'),
                ('alert/log','Channel1','01010000110000000000000011000000',True,False,"Plane's speed is 6000 Knots"),
                ('alert/log','Channel1','11010000111001100110010010000000',None,False,"Plane's Track Angle is 259.9 Degrees"),
                ('alert/log','Channel1','00110000111100000110010000000000',None,False,"Plane's Magnetic Heading is 98.3 Degrees"),
                ('alert/log','Channel1','10110000111100010010000000000000',True,False,'Wind Speed is 98.3 Knots'),
                ('alert/log','Channel1','11111001111000100100010010001000',True,False,'Baro Correction (ins. Hg) '),
                ('alert','Channel2','00000010000000101111100000000000',None,False,'Selected Course '),
                ('alert','Channel2','00100010000101101000000000000000',None,False,'Selected Vertical Speed is 1432 Ft/Min upwards'),
                ('log','Channel1','01010110110000000101011011000000',None,False,'Cabin Pressure is 1748 mB'),
                ('log','Channel1','11100101011000100110111111000000',False,False,'Horizontal Figure of Merit just under 2 nautical miles.'),
                ('log/alert','Channel1','10001001111001111010100000000011',None,False,'Indicated Angle of Attack is -70 WARNING!!'),
                ('log','Channel2','10000011000000111100000111100000',None,False,'Application Dependant Data 1 for FMC'),
                ('log','Channel2','11111111000000000000000000000100',None,False,'Identification required for FMC'),
                ('log','Channel2','01100011001100110110110011011000',None,False,'Application Dependant Data 2 for FMC'),
                ('alert','Channel1','00011101111001001010110110011000',None,False,'ADIRU Discrete Data '),
                ('alert/log','Channel1','00010111111001001010110110011000',None,False,'ADIRU MX Data '),
                ('alert','Channel2','00001111000000011110000000000000',None,False,'Aircraft Condition and Event Surveillance System (ACESS)'),
                ('alert','Channel2','10111110000101111110000000000000',None,False,'HGA/IGA HPA'),
                ('alert','Channel2','01011111000010111111000000000000',None,False,'CABIN TERMINAL 3'),
                ('alert', 'Channel2', '00010011000000100110000000000', None, False, 'GPWS'),
                ('alert','Channel2','10001111000100011110000000000000',None,False,'Electronic Flight Instrument System (EFIS)')])
```

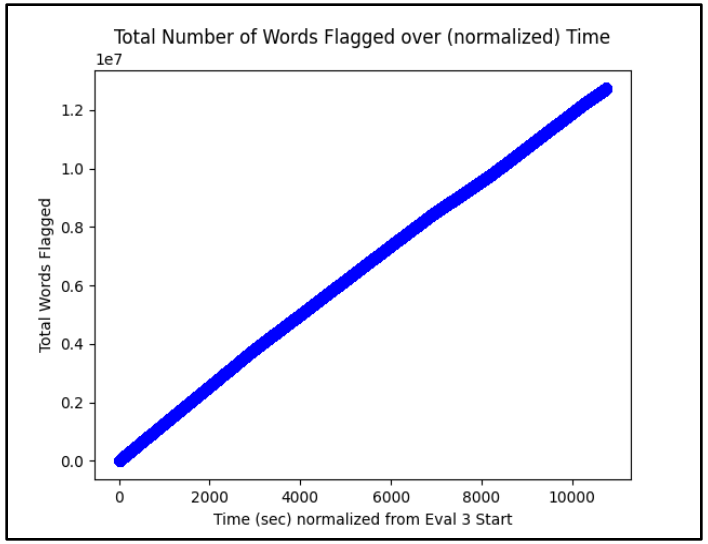
Evaluation 2:

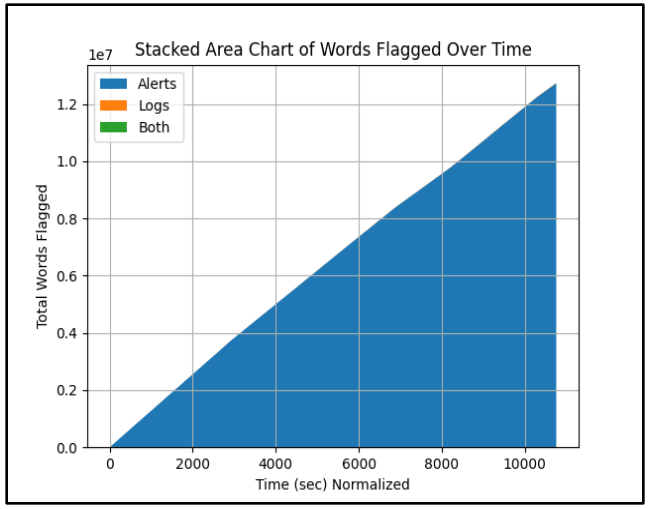
IDS_EVAL1.py → IDS Robustness

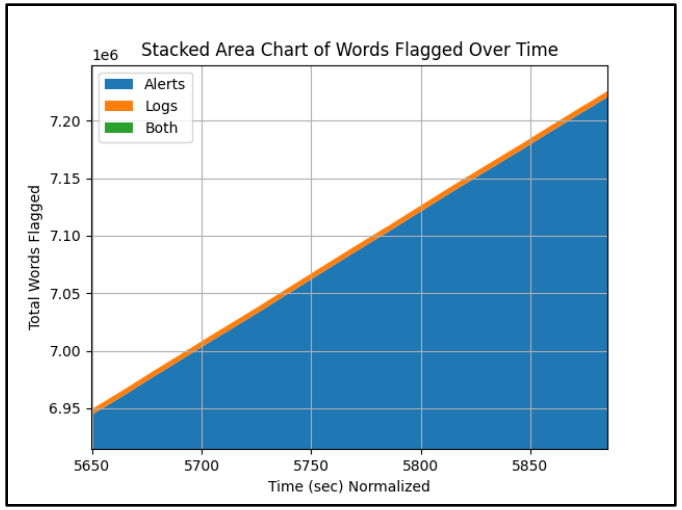
Bus Speed (% slowed down)	Voltage Sample Rate	Words Correct (of 5) averaged over 0-10 rules	Bits Correct per word averaged over 0-10 rules
-1,000,000%	0.5 sec (½ second)	5/5	32, 32, 32, 32, 32
-100,000%	0.05 sec (1/20th of a second)	5/5	32, 32, 32, 32, 32
-10,000%	0.005 sec (5 milliseconds)	5/5	32, 32, 32, 32, 32
-1,000%	0.0005 sec (½ millisecond)	5/5	32, 32, 32, 32, 32
-100%	0.00005 sec (1/20th of a millisecond)	5/5	32, 32, 32, 32, 32
-10%	0.000005 sec (5 microseconds)	4/5	32, 32, 32, 32, 31
-0%	0.0000005 sec (½ microsecond)	1/5	32, 31, 30, 29, 28

Evaluation 3:
IDS_EVAL2.py → IDS Correctness

```
Sending words:
Airspeed BCD:                   0b00011001000000000000000000000001,
Airspeed BNR:                   0b00010001000000000000000000000000,
Corrected Angle of Attack:      0b10000101000000000000000000000001,
Indicated Angle of Attack:      0b10001001000011100100000000000111,



This concludes Eval 2. It took 10164.366 seconds.
Number of alerts: 12726074
Number of logs: 9623

Process finished with exit code 0
```



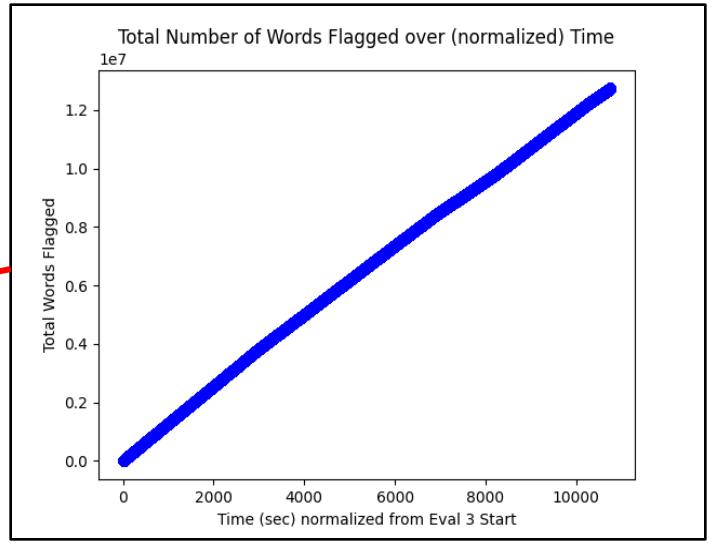


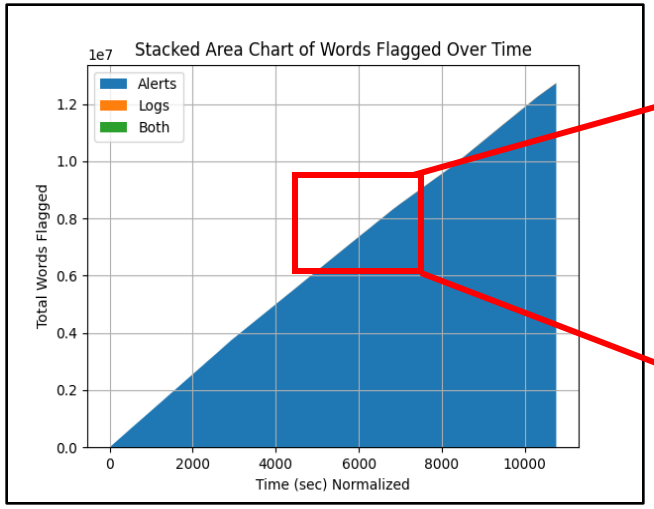


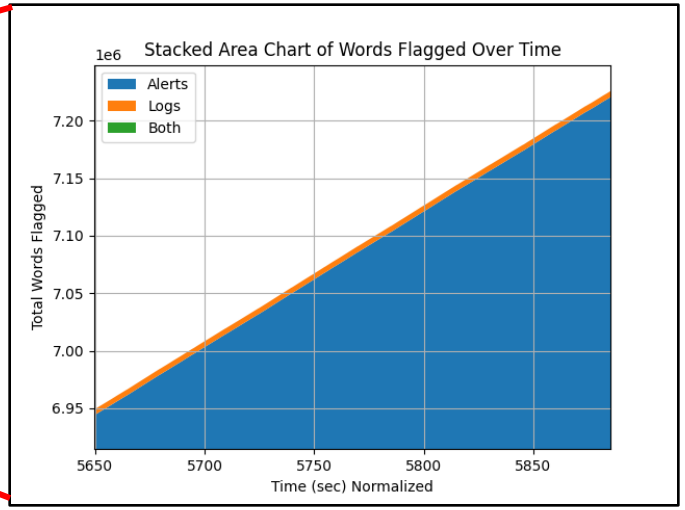
Evaluation 3:

IDS_EVAL2.py → IDS Correctness

```
Sending words:
Airspeed BCD:                   0b00011001000000000000000000000001,
Airspeed BNR:                   0b00010001000000000000000000000000,
Corrected Angle of Attack:      0b10000101000000000000000000000001,
Indicated Angle of Attack:      0b10001001000011100100000000000111,



This concludes Eval 2. It took 10164.366 seconds.
Number of alerts: 12726074
Number of logs: 9623

Process finished with exit code 0
```

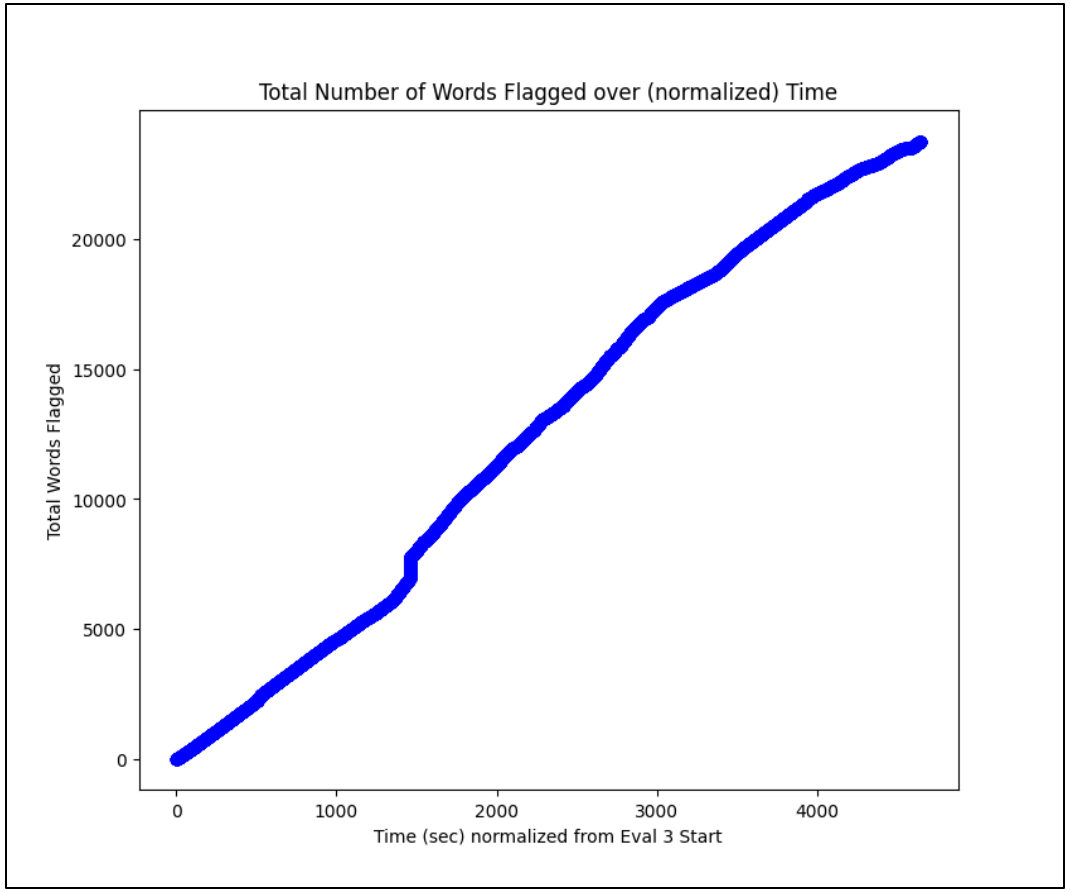


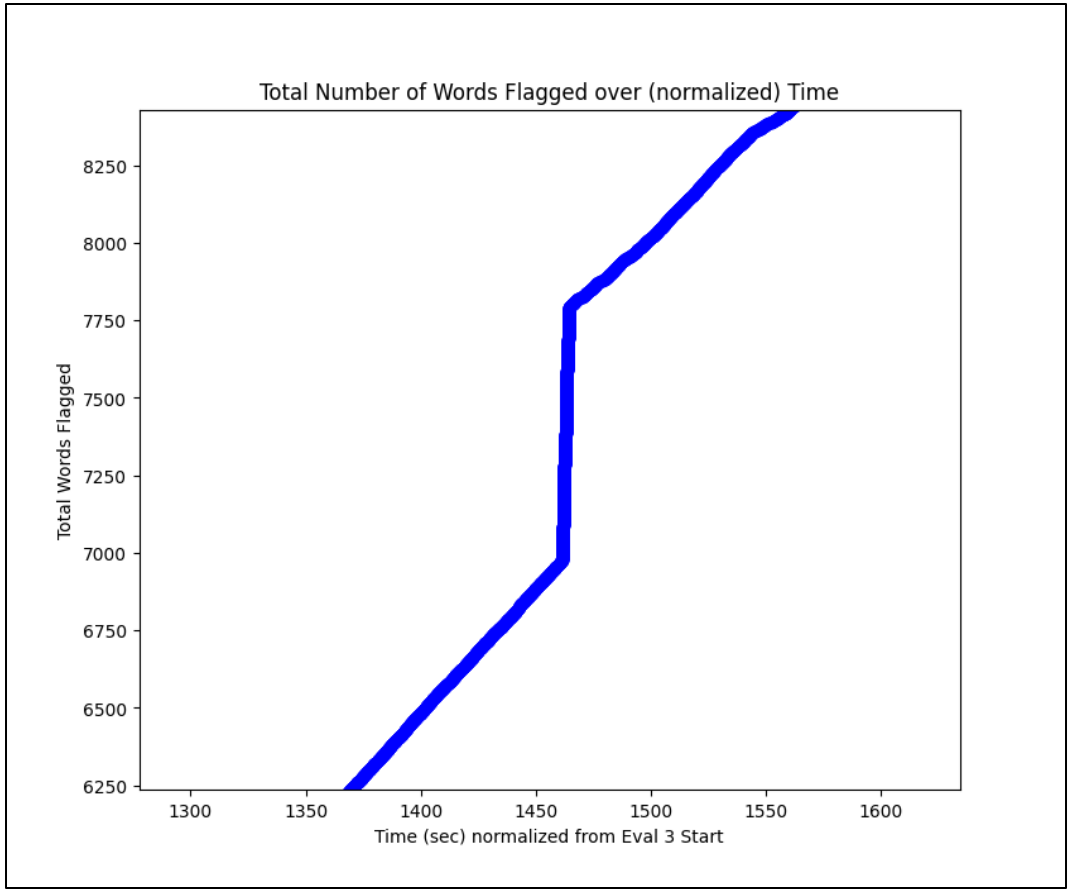




Evaluation 4:

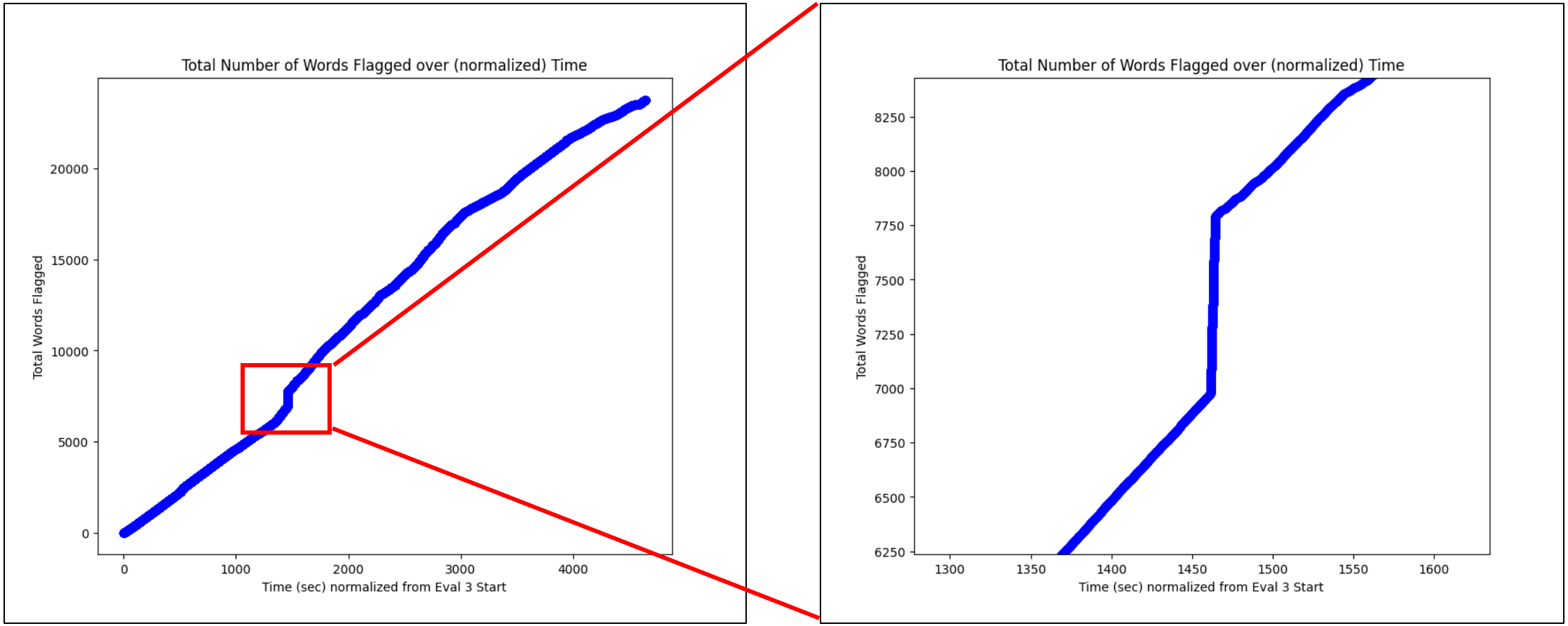
IDS_EVAL3.py → IDS Detection of Attackers





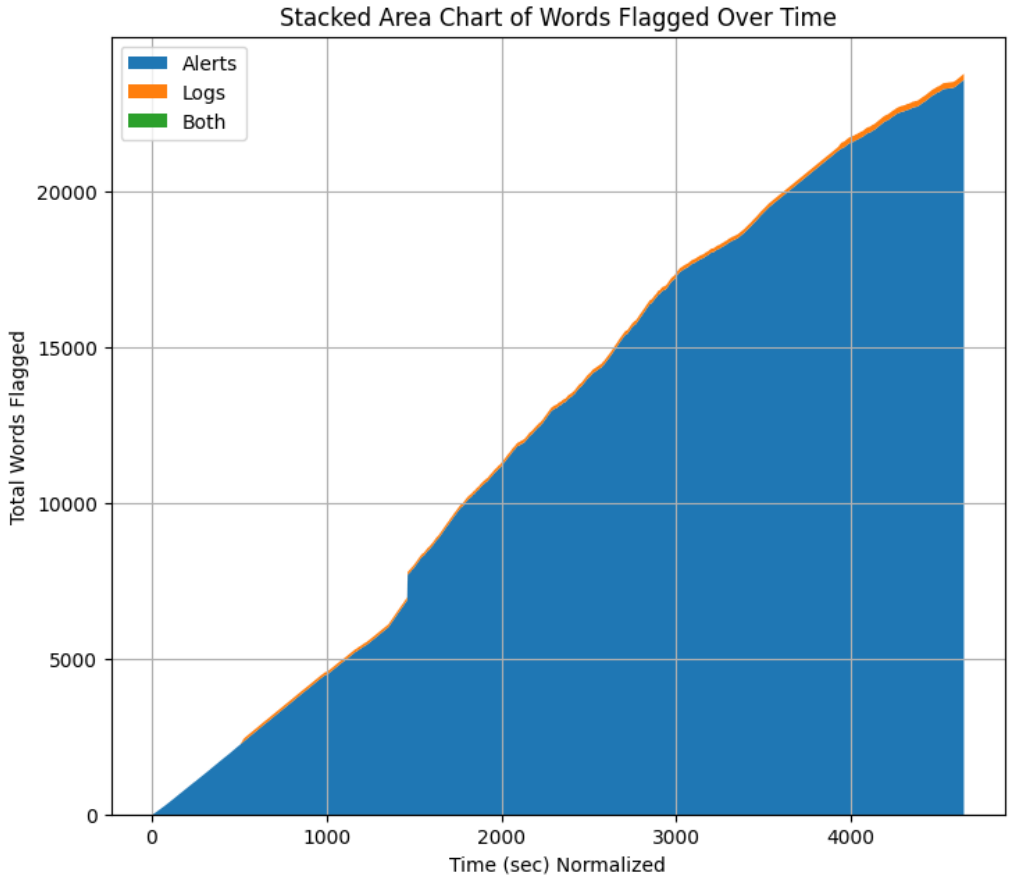
Evaluation 4:

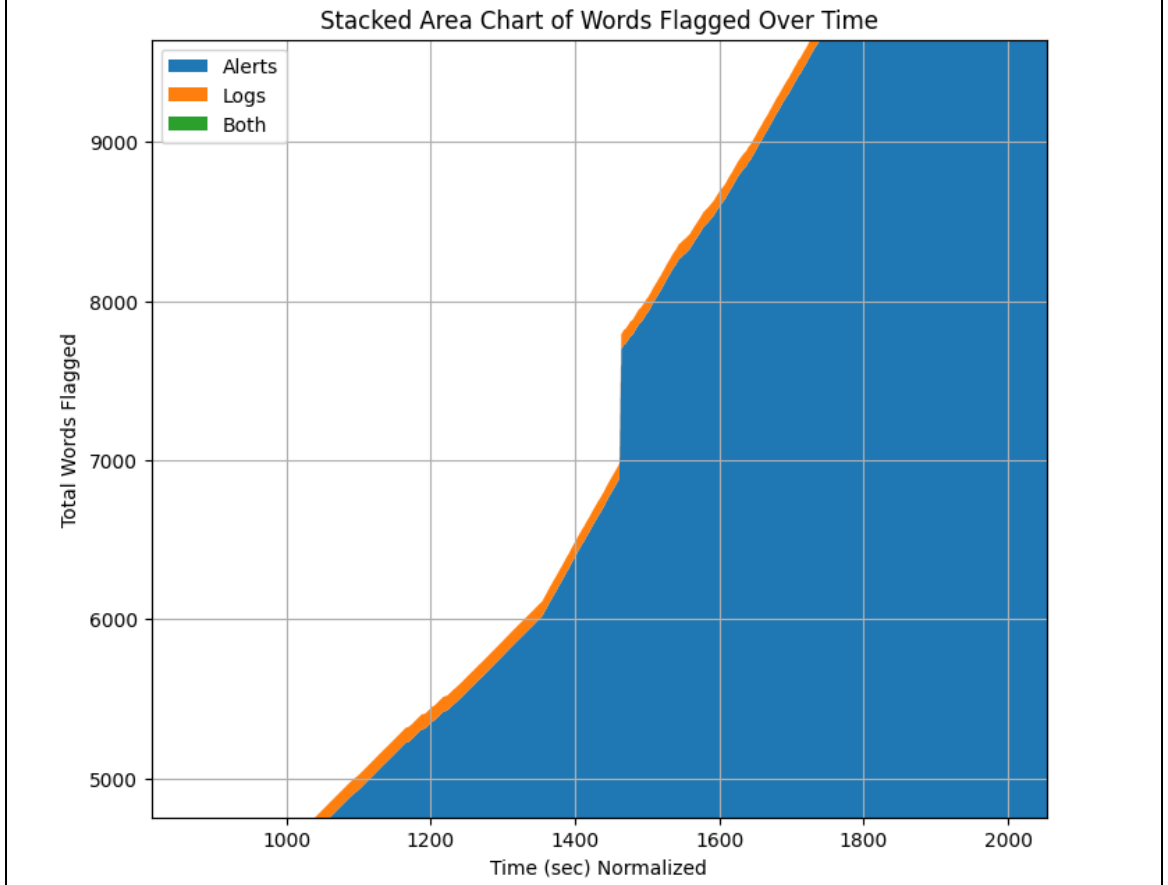
IDS_EVAL3.py → IDS Detection of Attackers



Evaluation 4:

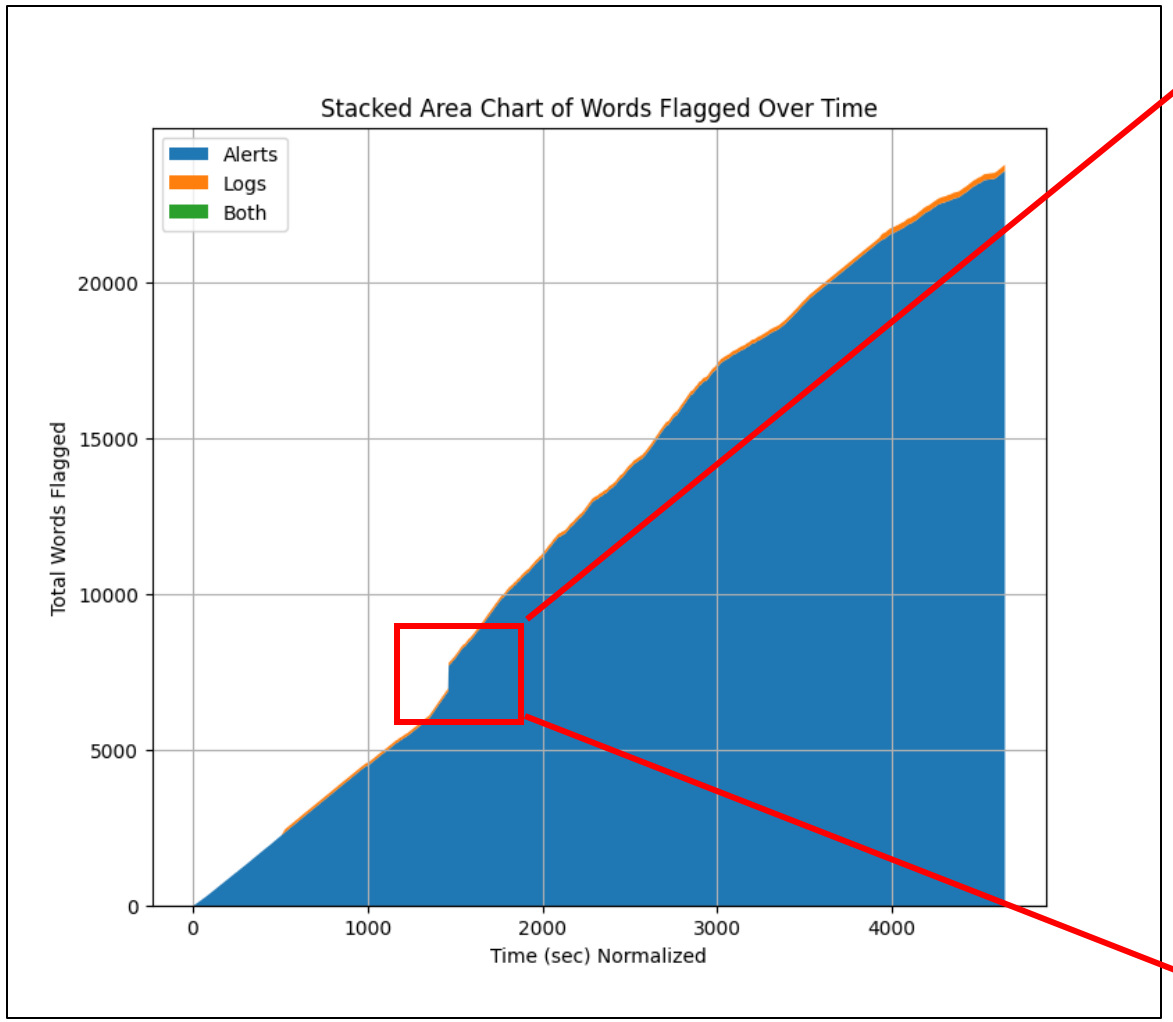
IDS_EVAL3.py → IDS Detection of Attackers

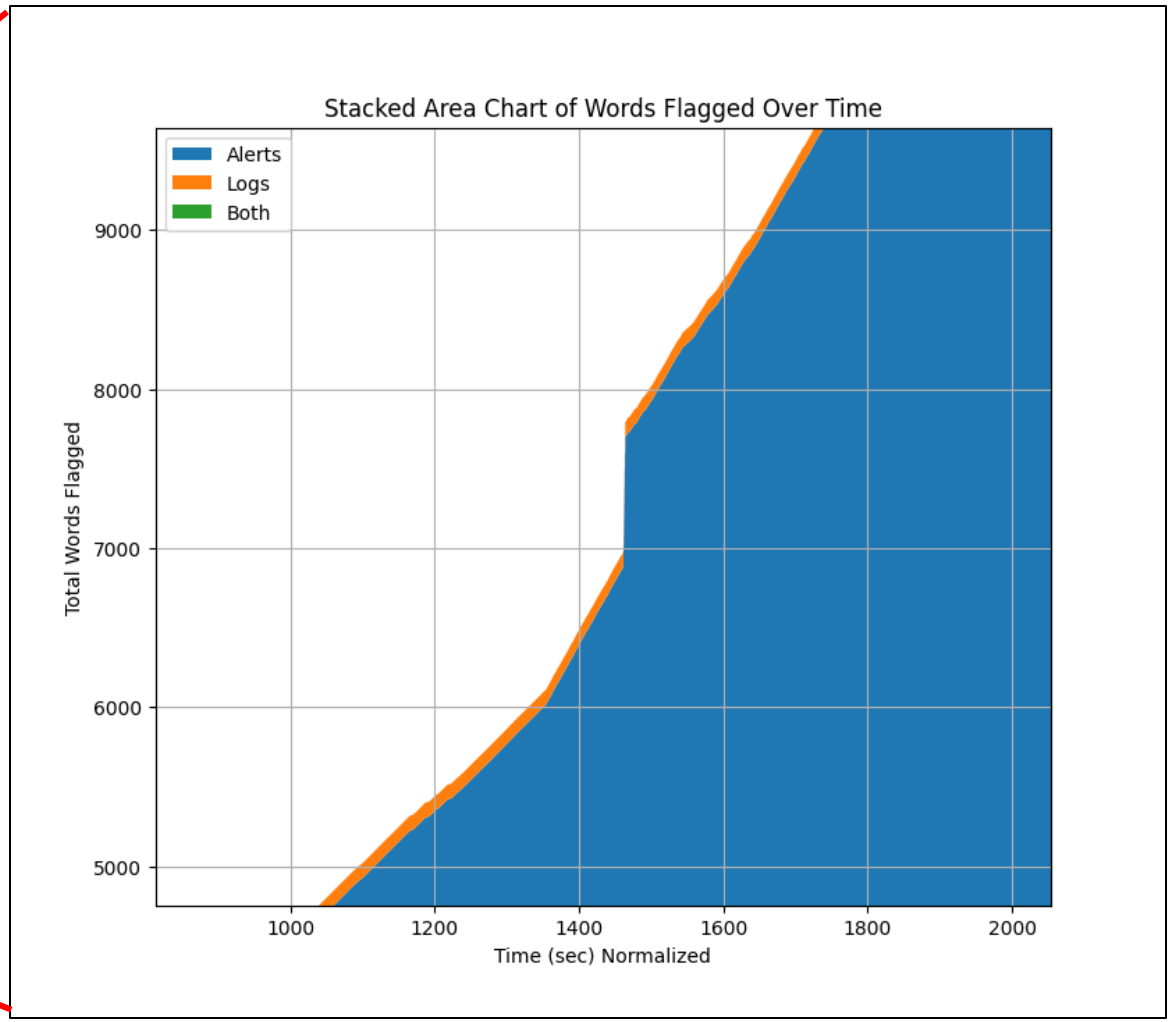




Evaluation 4:

IDS_EVAL3.py → IDS Detection of Attackers





Evaluation 5:
IDS_EVAL4.py → Time Added by IDS

Parameters:

- 36 Channels

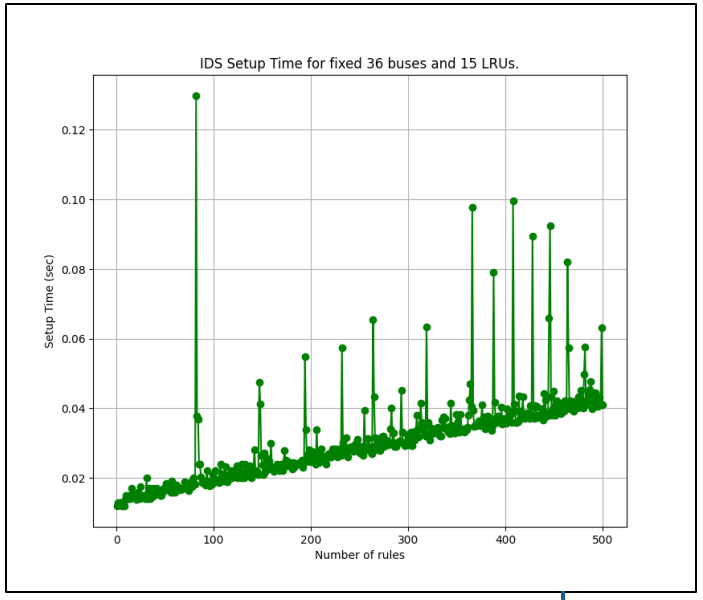
- 15 LRUs per Channel

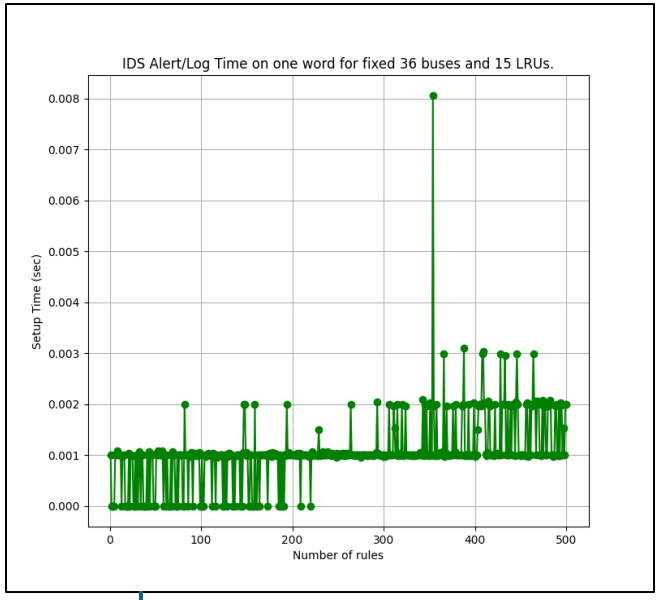
- 1 to 500 rules

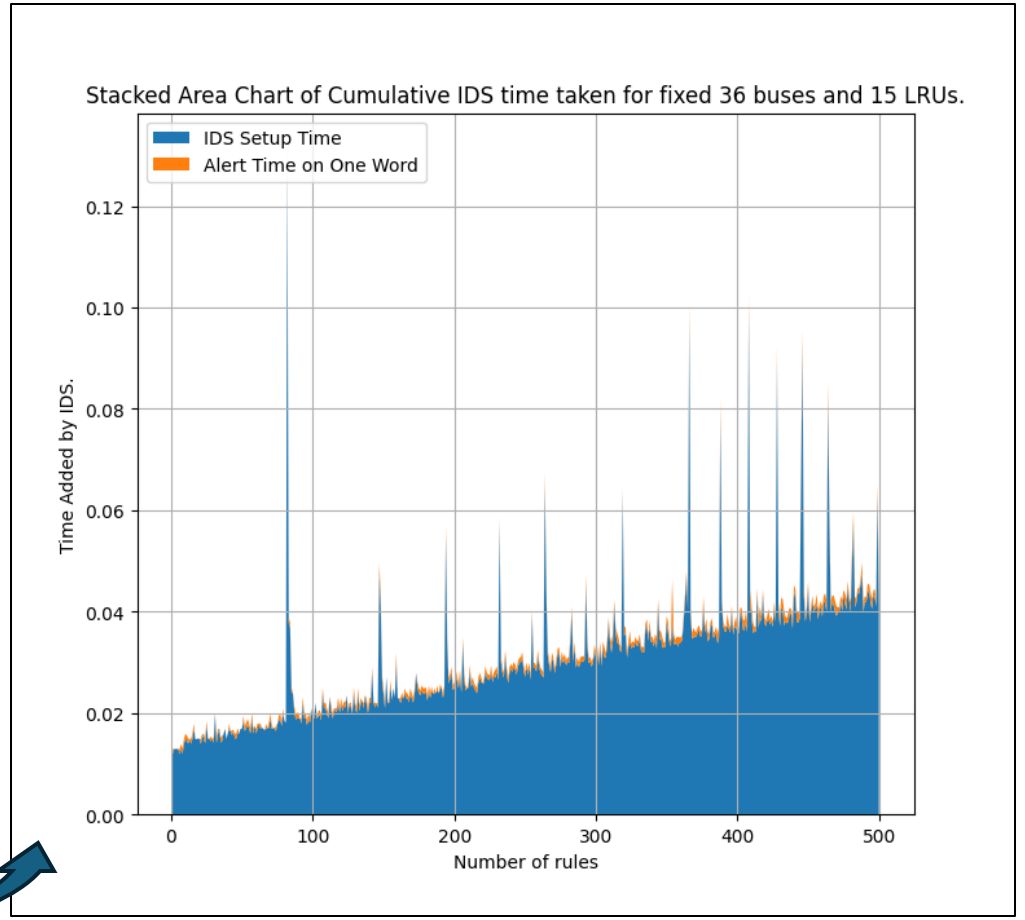
Outputs:

<u>- Time to set up IDS</u>

<u>- Time to Alert/Log on one word</u>







Evaluation 5:
IDS_EVAL4.py → Time Added by IDS

Parameters:

- 48 Channels

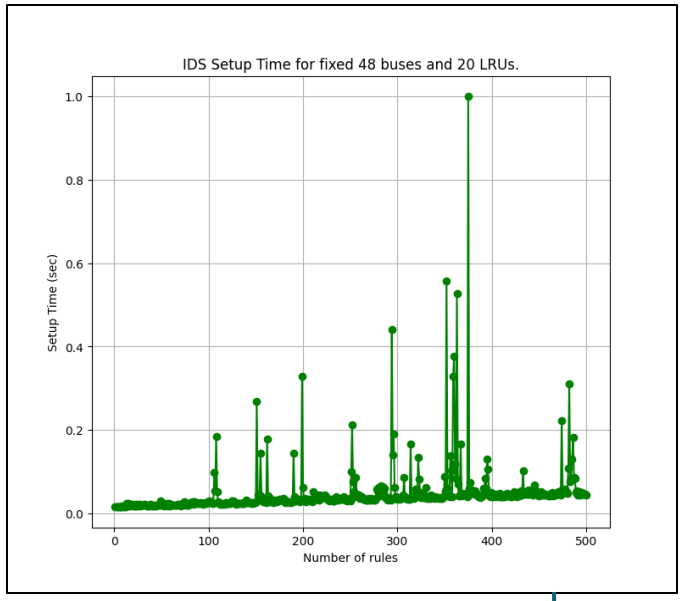
- 20 LRUs per Channel

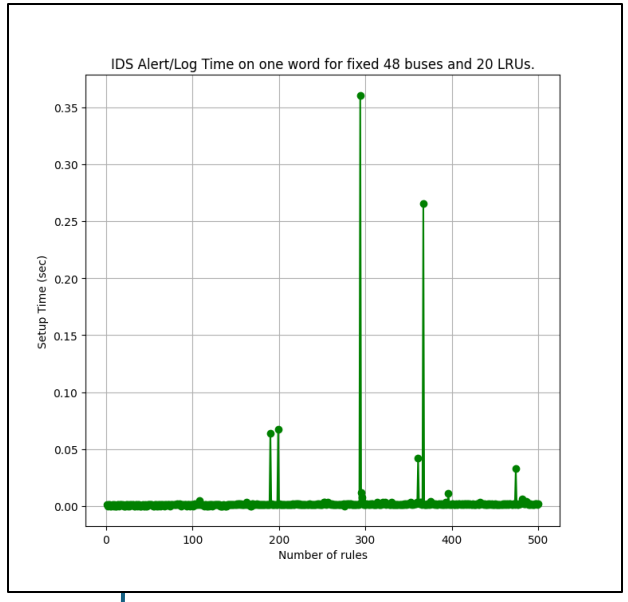
- 1 to 500 rules

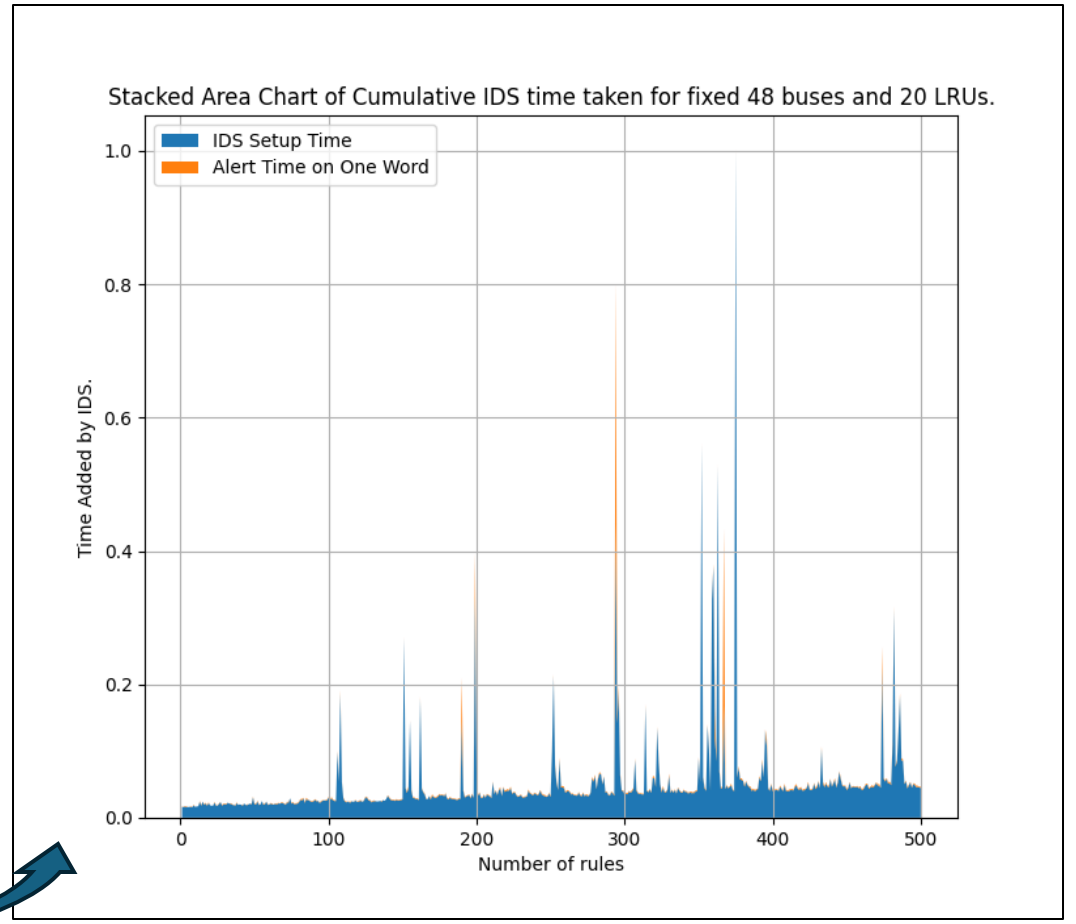
Outputs:

<u>- Time to set up IDS</u>

<u>- Time to Alert/Log on one word</u>



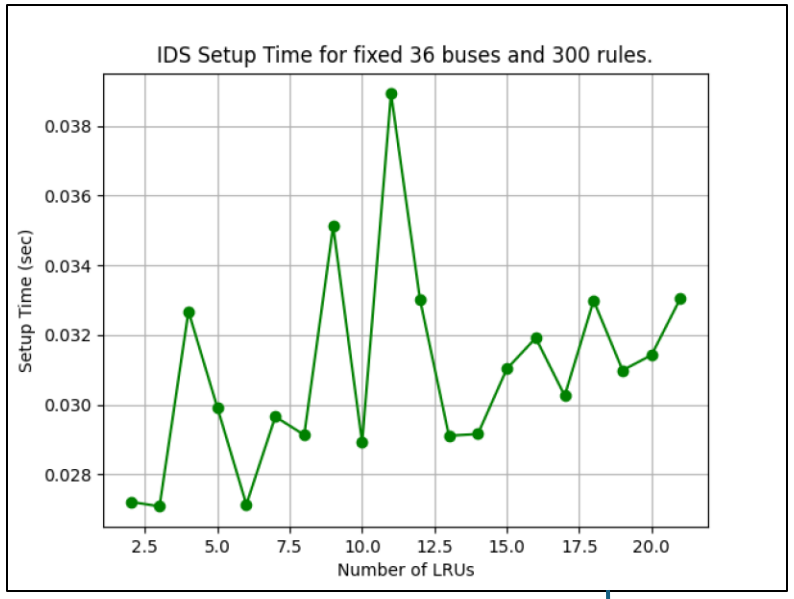


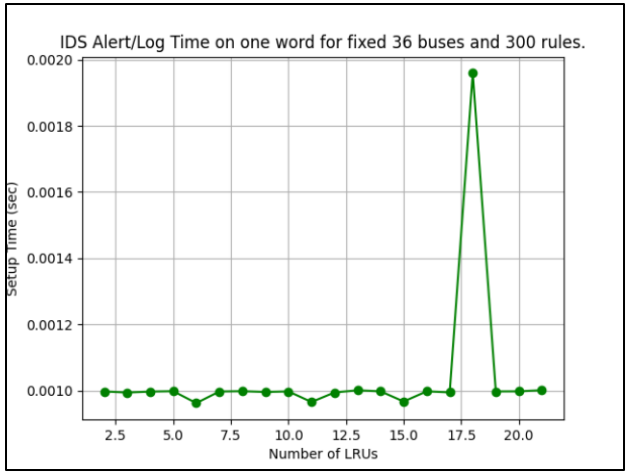


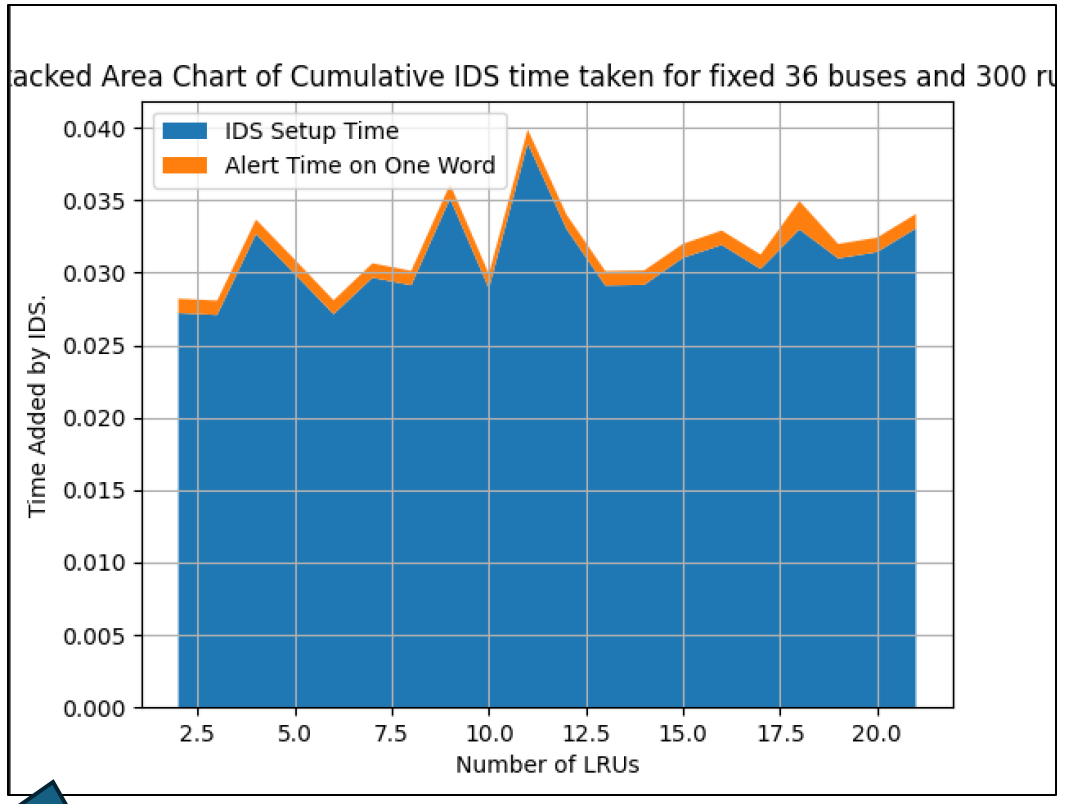
Evaluation 5:
IDS_EVAL4.py → Time Added by IDS

Parameters:
- **36 Channels**
- *1 to 21 LRUs/Channel*
- **300 rules**

Outputs:
- Time to set up IDS
- Time to Alert/Log on one word

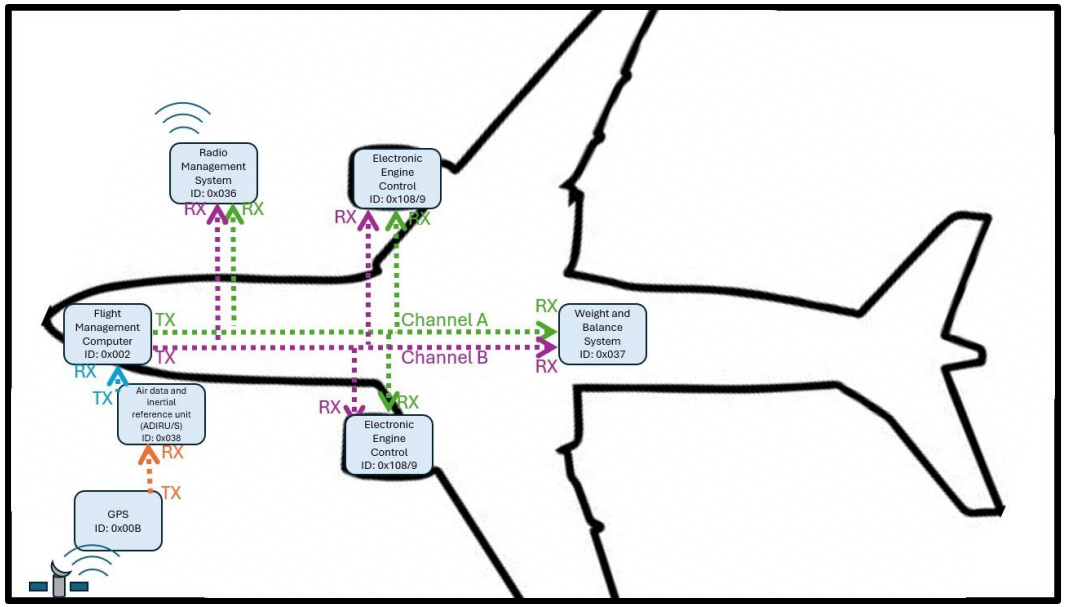






Limitations

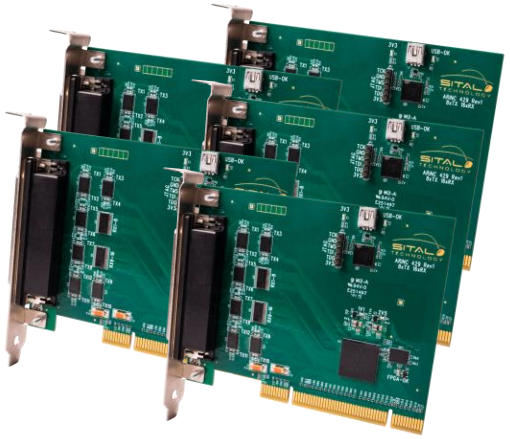
1. Simplicity of Model
2. Python Multithreading
3. Operator Training/Know-how
4. Stoppage
5. Fulfilling All Zero Trust Tenets
6. Untested on hardware



Future Work / Possible Next Steps

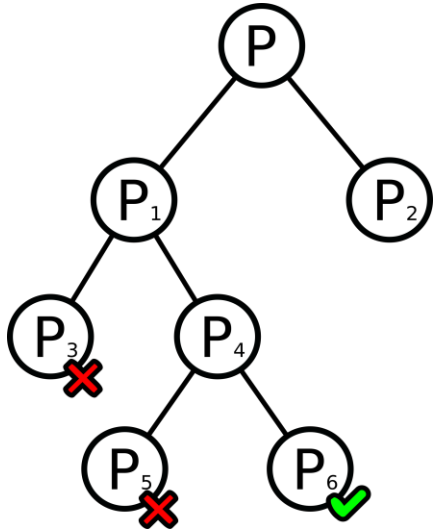
1. Expanding the simulation model (make more LRUs)
2. Recode Sim in more thread friendly language (e.g. C, rust, etc.)
3. Expand IDS functionality (condition chain)

1.



2.

3.



Future Work / Possible Next Steps

4. Create an IPS
5. Implement and test IDS on real hardware
6. Collect real world 'word' data (for further testing / maybe ML IDS creation?)
7. Diversify Threat Model and attack set(s)

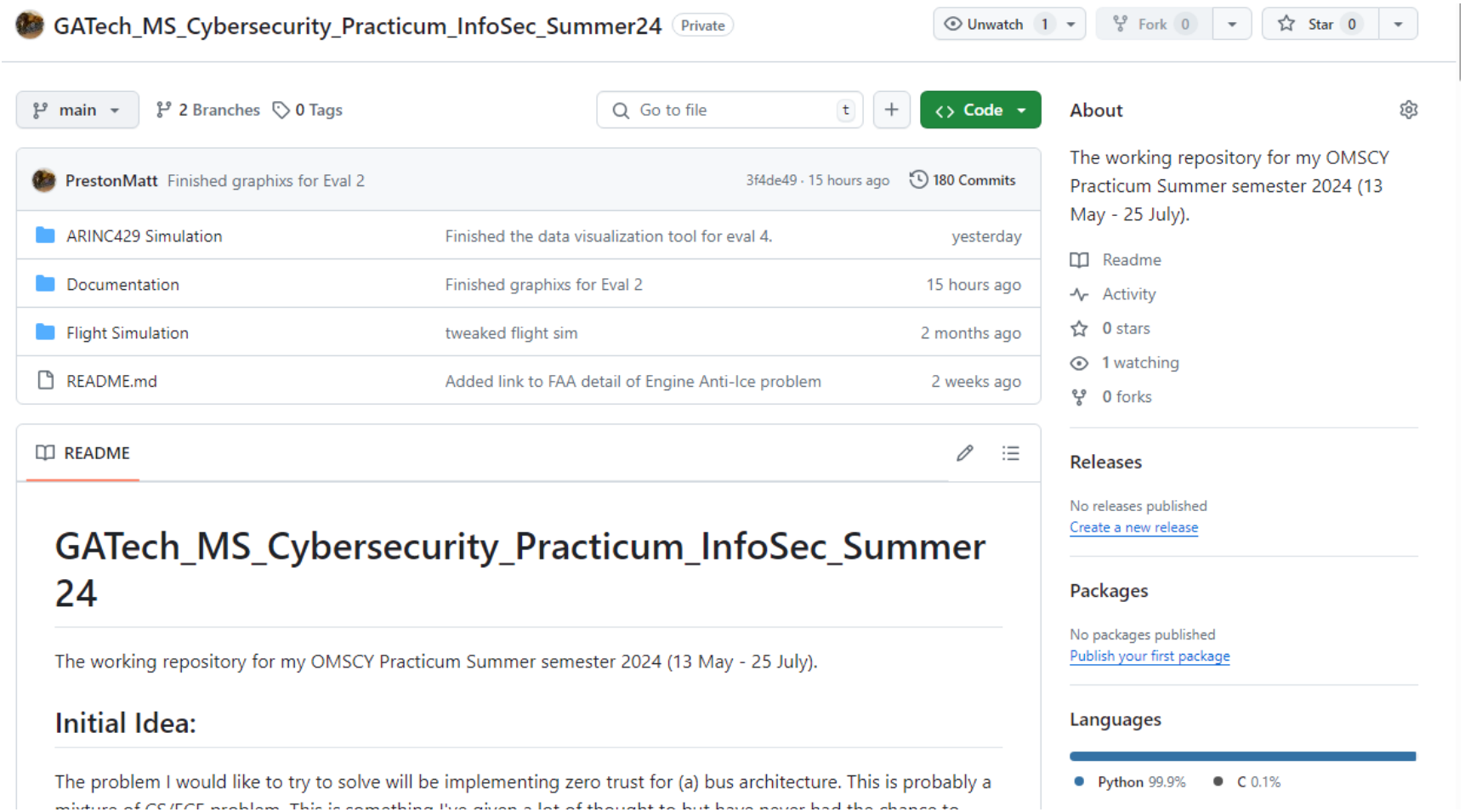
4.

5., 6.

7.

Summary

- Problem Description
- Solution
- Demo
- Evaluations
- Limitations
- Possible Next Steps / Future Work



https://github.com/PrestonMatt/GATech_MS_Cybersecurity_Practicum_InfoSec_Summer24

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/

Matthews, Bryan. "Flight Data for Tail 687." DASHlink, National Aeronautics and Space Administration, 4 Dec. 2012, https://c3.ndc.nasa.gov/dashlink/resources/664/

"DLH754, Leg 2, 2024-07-02." ADS-B Exchange, adsbexchange.com, Last Accessed: 15 Jul. 2024, Published: 2 Jul. 2024 https://globe.adsbexchange.com/?icao=3c4b35&lat=40.621&lon=52.094&zoom=4.7&showTrace=2024-07-02&leg=2