System Requirements Specification

for

Aircraft Domain Security Enhancement

Version 4 approved

Prepared by
Anthony Johnson, Matthieu Privat, Charles Gilmore, Jacob Stephens, Allen Biagetti, Jorge Santos, Max Gorley, and Max Wilson

Embry-Riddle Aeronautical University

February 3, 2022

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
Max Wilson	9/17/21	Document Initiation/Set-up	1
Anthony Johnson	9/17/21	Purpose & Product Scope	1.01
Max Wilson	9/26/21	Addition of more details to product description	1.02
Anthony Johnson, Max Wilson, & Maxwell Gorley	9/27/21	Completion of SRS rough draft for sprint 1	1.03
Max Wilson	9/28/21	Final overview and completion of SRS draft 1	1.04
Max Wilson, Anthony Johnson	10/25/21	SRS draft 2 revision and completion	2
Max Wilson, Anthony Johnson	11/30/21	SRS draft 3 revision and completion	3
Anthony Johnson	2/2/22	SRS draft 4 revision	4
Max Wilson	2/3/22	SRS draft 4 review part 2	4.01

1. Introduction

The aircraft domain is a complex system containing many critical and non-critical systems used in flight. An aircraft's network can be split into three levels: the Aircraft Control Domain (ACD), the Airline Information Services Domain (AISD), and the Passenger Information and Entertainment Services Domain (PIESD). In this project, we hope to emulate the AISD network and potentially move towards connecting it with the ACD. Once these systems are set in place, we will investigate how these domains can be simulated, and tested for cybersecurity.

1.1 Purpose

This document specifies the requirements for an aircraft domain emulation and testing environment called the Aircraft Domain Security Enhancement version 3.0. This involves the development and configuration of an Aircraft Information Services Domain (AISD) emulation, and a compatible penetration testing kit. Upon completion, a Capture the Flag (CTF) event will be held in which people will use the penetration testing kit to search for keys hidden in the network. This is done via SSH and VCM depending on the set up of both teams (Prescott and Daytona Beach).

1.2 Document Conventions

Not applicable for current documentation.

1.3 Intended Audience and Reading Suggestions

This document is intended for cybersecurity experts, researchers, and users who interact with aircraft networks every day. It is recommended to begin with product scope, to get a vision of this project. For all readers, it is best to review the external interface requirement sections here, to get a better understanding of how the project was implemented and tested.

1.4 Product Scope

The aircraft domain security enhancement is intended to provide a testing ground for aircraft networks before they are implemented with real hardware. There are two stages of development for this project. First, the AISD will be emulated allowing for remote testing and configuration. The emulated network will encompass several mediums to low priority components to the network. Second, a penetration testing kit will be developed to test the security of the overall domain. The aircraft domain is a complex system containing many communicating critical and non-critical components. Testing these systems becomes increasingly difficult as more components are added which further leads to the importance of a flexible emulated environment. The completion of this project will provide a virtual environment capable of testing the security of the aviation domain being evaluated.

1.5 References

At this time there are no applicable references.

2. Overall Description

2.1 Product Perspective

This product is an emulation of current aircraft network nodes, such as the aircraft printer, control console, and entertainment network. Below is a breakdown of each network “node” and how it interacts with each system to create a fully interactive aircraft network. It has the ability to be expanded upon and is isolated. This allows users to be able to test malicious cyberattacks against the machine and learn how to improve the security of the network over time.

2.2 Product Functions

Below is a breakdown of both hardware and software capabilities for the product:

1. Emulated AISD network
2. Penetration testing kit modules
3. TFTP data loading server
4. Secure gate access point
5. Web/Log server
6. Software-firewalls
7. Encrypted file transfer
8. SSH Client Access

2.3 User Classes and Characteristics

The system designed for this project simulates a real aircraft information system but an important distinction to make is that it is first and foremost a test system. Because of this, user classes are limited to either security professionals or sources of data to the system. These user classes are defined as the following:

2.3.1 System Administrator:

The System Admin. has the most authorized control of the network. They are in charge of monitoring network activity and appropriately configuring components of the network to ensure the system is running correctly. In a CTF scenario, this user class would serve as the blue team.

2.3.2 Penetration Tester:

The Pentester has no authorized access but will interfere with the network nonetheless. The highest threat the Pentester holds against the system is to be able to execute malware on the system. This user class would serve as the red team.

2.3.3 Maintenance Service:

The Maintenance service is intended to be an automated service in this test and will be a source of system update files. This service sends .ZIP files into the network to be loaded onto computers for updates.

2.3.4 Airline Service:

The Airline service is also an automated service that will be a source of weather updates to the system. This service sends .JSON files containing weather updates.

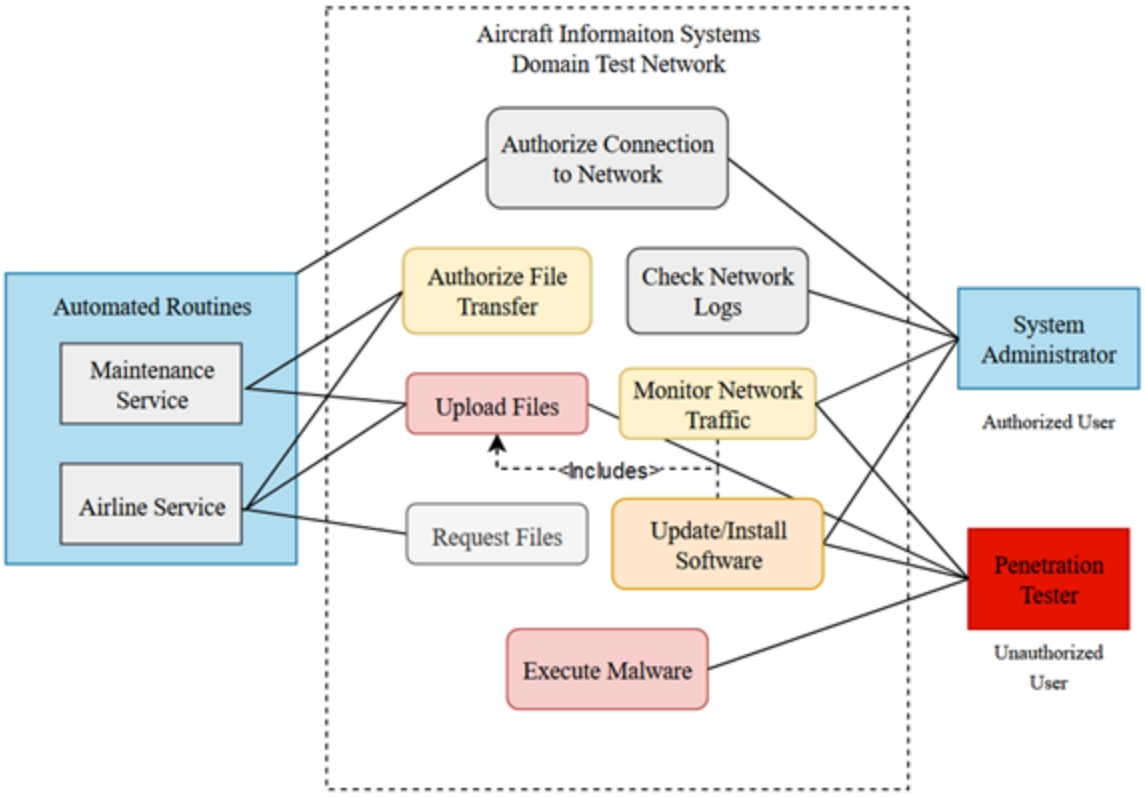

Figure 1: AISD Test Network Use Case Diagram

In the figure above, the criticality of use cases is depicted by how red the use case is. Gray indicates benign use cases whereas red indicates critical use cases. The two worst cases for an unauthorized user to have access to are uploading files, and executing malware. This is because these both can result in the complete breach of security for the information systems on the aircraft. Tests will be conducted to see if these use cases are possible to be accessed by an unauthorized user.

2.4 Operating Environment

The software used for the aircraft network system is Linux-based with QEMU as the virtualization software. Multiple versions of Ubuntu OS were used to set up each node and physical host machines. The hardware being used is multiple laptops with eight to sixteen gigabytes of RAM, and up to one terabyte hard drive. The penetration testing toolkit runs off of VirtualBox and/or QEMU depending on the user's main configuration for set up. The toolkit base is Linux ParrotSec Cybersecurity Edition which hosts a variety of penetration testing tools to help facilitate the exploitation of the AISD network.

2.5 Design and Implementation Constraints

Due to the scale of the project, as well as the timeframe, there are several constraints of the system. The first is hardware limitations due to shipping times, funding, and integration with the QEMU virtualization software. Another main limitation is the security considerations. Although the main part of the project is to design and harden an aircraft network system, there are always "zero-day" flaws that are unknown.

2.6 User Documentation

Not applicable to current systems.

2.7 Assumptions and Dependencies

It is assumed that a high-speed broadband internet connection will be available so that all virtual machines can be managed. An additional assumption is that an SSH client will be installed on all computers trying to connect to the virtual machine network. The virtual machines depend on a sufficiently powerful Linux-based host with at least 16 gigabytes of RAM to run them.

3. External Interface Requirements

3.1 User Interfaces

The users will interact with the system via a command-line interface (CLI) in order to perform maintenance tasks and manage the system via SSH.

3.2 Hardware Interfaces

The hardware encompasses a Linux operating system (OS) on a Laptop that has 16 GB of RAM and a 500 GB hard drive. This hardware hosts the software which runs QEMU (see below for details regarding software). Future iterations of the hardware will include servos to mimic wings, a joystick for flight control, along with additional interface instruments to give the feel of an actual aircraft and its network. Through these interfaces, users can run physical penetration tests against the machine instead of only virtual ones.

3.3 Software Interfaces

The software implemented for the network is Linux, which has more built-in functionality that allows ease of access for the user when working with the QEMU emulator. The emulator hosts the entirety of the aircraft network system consisting of multiple “nodes” or aircraft components.

As of Semester 2:
The AISD is in an isolated internal network due to the possibility of malware leaking onto the network. For penetration testing, an Ubuntu machine will serve as the machine used by pentesters to further navigate into the system. This machine will have all tools needed for the first CTF and a readme file to lay out what to look for in the network.

3.4 Communications Interfaces

The main communication interface of the project is the TFTP (Trivial File Transfer Protocol) server. The TFTP server acts as a bridge between the ADIS and other nodes of the aircraft network system. The server is hosted by a QEMU emulator that has a built-in TFTP network that can be incorporated with the rest of the systems for this project. The TFTP has built-in encryption and is interacted with via an SSH (Secure Shell) server.

The SSH server is the main hub of the aircraft network emulator. It allows the user remote access via logging in through a terminal or command prompt. The SSH server is encrypted and only available to users who have been cleared to access it. The server is on a Linux-based computer that hosts all of the QEMU network interfaces. By doing this, multiple people have access to the server at one time to simulate cyber-attacks, harden security, or do maintenance on the network.

4. System Features

4.1 Gate Access Point

4.1.1 Description and Priority

The Access Point serves as the outermost node in the network that is used to connect to. This point requires authentication to have access to the other machines on the network.

4.1.2 Stimulus/Response Sequences

1. The client will initiate a connection to the SSH server.
2. The SSH server will respond with a login prompt.
3. The username of the desired user will be entered.
4. The user is prompted to enter a passphrase for the private key.
5. Once the private key is unlocked, the public key for that user is calculated from the private key and sent to the server for verification.

6. If the keys match, the user will be prompted to enter a six-digit code presented by a Google Authenticator-compliant app or security key.
7. If the code is entered correctly, the user will be presented with a Bourne-Again Shell (Bash) prompt where commands may be entered.
8. If any of the above steps fail, the user will be disconnected from the SSH server.

4.1.3 Functional Requirements

System Requirements Specification, 4.1.3.1: The client shall connect to the Access Point via an SSH Server

System Requirements Specification, 4.1.3.2: The Access Point shall send uploaded JSON files to the Central Server

System Requirements Specification, 4.1.3.3: The Access Point shall send uploaded ZIP files to the TFTP Server

4.2 Printer

4.2.1 Description and Priority

The printer provides the network access to printed reports of the weather, aircraft information, and other important aspects required during a flight.

4.2.2 Stimulus/Response Sequences

1. The CUPS service shall receive a document from the Central Server.
2. The CUPS service shall forward this document to the printer.
3. Upon receiving the document, the printer shall print out that document.
4. In the event of an error, such as running out of paper, the printer shall display an error message.

4.2.3 Functional Requirements

System Requirements Specification, 4.2.3.1: The Printer shall print documents sent to it

4.3 TFTP Server

4.3.1 Description and Priority

The TFTP Server is used to load maintenance updates through zip files onto machines in the network. These files are uploaded from the access point and sent to this server.

4.3.2 Stimulus/Response Sequences

1. The client shall SSH into the network via login over the server.
2. The client shall load the virt-manager within the main Ubuntu system to access QEMU.
3. The client will select the TFTP Ubuntu Server from the QEMU virt-manager list.
4. The client then will load the command prompt, and use the CD command to enter the correct folder where the TFTP structure is set up.
5. Then they will enter an IP to confirm the TFTP's IP address to initiate a TFTP transfer.

6. The client will enter the command, TFTP, followed by the IP address given in the previous step.
7. The TFTP server is now active and can send and receive files. To add a file to the TFTP server, use the command put. To send a file use the command get.
8. The TFTP server will shut down when the client uses the command, quit within the TFTP prompt.

4.3.3 Functional Requirement

System Requirements Specification, 4.3.3.1: The TFTP Server shall upload files to other computers in the network

4.4 Central Server

4.4.1 Description and Priority

The purpose of this machine is to route and filter network traffic for the AISD network and serve as a bridge to the ACD network.

4.4.2 Stimulus/Response Sequences

1. All messages sent in the AISD network shall pass through the Central Server.
2. Messages sent across the network shall be filtered by a firewall.
3. A log of network traffic shall be continually updated.

4.4.3 Functional Requirements

System Requirements Specification, 4.4.3.1: The Central Server shall maintain an ongoing network log

System Requirements Specification, 4.4.3.2: The Central Server shall host a web server

System Requirements Specification, 4.4.2.3: The Central Server shall send instructions to the Printer

5. Other Nonfunctional Requirements

5.1 Performance Requirements

Not applicable to the current system.

5.2 Safety Requirements

SAF-1: The aircraft domain network system shall be shut down if malware leaks out of the enclosed system.

5.3 Security Requirements

SEC-1: A firewall shall be installed on the virtual machine network.
SEC-2: All ports other than the ones needed for the AISD network shall be closed on the firewall.
SEC-3: Only team members, professors, and Boeing shall have access to the aircraft domain network system.

5.4 Software Quality Attributes

The software used for the project is adaptable, portable, and readily available. The virtual network can be expanded upon at any point to include more aircraft network systems. The network is also available via SSH for ease of access across users.

5.5 Business Rules

Only authorized users will be permitted to access network resources.

6. Other Requirements

TBD

Appendix A: Glossary

Term	**Definition**
ACD	Aircraft Control Domain
AISD	Aircraft Information Services Domain
Bash	Bourne-Again Shell, the default Linux command shell
CLI	Command-line interface
CUPS	Common UNIX Printing System
OS	Operating System
PIESD	Passenger Information and Entertainment Services Domain
RAM	Random Access Memory

SSH	Secure Shell
TFTP	Trivial File Transfer Protocol
VM	Virtual Machine
VCM	Variable Coded Modulation
Zero-day	A type of cyberattack that uses a previously unknown vulnerability of a system, to carry out malicious activity.

Appendix B: Analysis Models

Not applicable.

Appendix C: To Be Determined List

Not applicable.