System Requirements Specification

for

Unmanned Aerial Systems Simulated Cyberwarfare

Version 1.0 approved

Prepared by Ryan Casa, Aliya Trussell, Fahad Alkaabi, Stefan Miller

Embry-Riddle Aeronautical University

10/07/2022

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
All	10/6/2022	Start of Document	0.0
Ryan Casa	10/7/2022	Initial Draft	1.0

1. Introduction

1.1 Purpose

The purpose of this product is to simulate the network events of unmanned drone aircraft, for the purposes of conducting experimental cyberattacks and creating proof of concepts for those attacks. Our simulation will use network based simulation software to create a connected drone swarm. Once created, we will test the results of various network attacks to determine their effect on the swarm.

1.2 Document Conventions

This document, along with the System Design Document follow conventional formatting practices to make comprehension easier for the reader. This is subject to change as the documents evolve.

1.3 Intended Audience and Reading Suggestions

The reader of this document is assumed to have basic undergraduate level knowledge in networking, simulation theory, unmanned aerial systems, and linux operating systems. Additional understanding may be found in the references section, the github page for the project, and the system Design Document.

1.4 Product Scope

Drone swarms are interconnected unmanned aerial vehicles that have applications ranging from defense, package delivery, emergency management, search and rescue, and entertainment. However, these systems are connected, which forces them to be susiciple to cyber based attacks. Our project attempts to document and demonstrate the effect of various cyber attacks on the productivity and usability of a drone swarm.

1.5 References

[1] “SOSPUAS,” *GitHub*, Mar. 28, 2022. https://github.com/AkbasLab/SOSPUAS (accessed Oct. 06, 2022).

[2] nsnam, “ns-3,” *ns-3*, 2019. https://www.nsnam.org/

2. Overall Description

2.1 Product Perspective

The product's backend is a fork of ns-3, an open-source network events simulation software. The product is self-contained & self-sufficient. The stimulation itself is based on the work of [1].

2.2 Product Functions

The product must be able to accurately simulate the network operations of unmanned drone aircraft. This includes both drone-to-drone and drone-to-host network traffic. Additionally, it must simulate the attacker's network.

2.3 User Classes and Characteristics

This product will be used by our team to demonstrate the effects of cyber attacks. Therefore, we are the only users. However, we will use documentation, engineering notebooks in the form of Github Wiki, and professional programming practices to make our project reusable for future research.

2.4 Operating Environment

Each member runs our environment and NS-3 builds from their personal computers. NS-3 requires a Linux operating system. Further, the system must have Python3 and CMake installed.

2.5 Design and Implementation Constraints

The only identified limitation in our project is the learning curve of NS-3. Most of our members are not familiar with Linux and NS-3 is a powerful, yet difficult to learn software. A lot of our effect will be mastering NS-3 before proper development and testing of Cyber attacks.

2.6 User Documentation

Our product will deliver this System Requirements Specification, a System Design Document, and a System Test Document. We will also deliver our product, including an in-depth wiki and documentation.

2.7 Assumptions and Dependencies

We assume that NS-3 will give us all the capabilities we need to simulate an accurate and realistic drone swarm. Additionally, we assume NS-3 will allow us to conduct an accurate cyber attack. Because of these assumptions, our only dependency is NS-3.

3. External Interface Requirements

3.1 User Interfaces

NS-3 has a very limited GUI interface, therefore, our project will use the Linux Command Line Interface.

3.2 Hardware Interfaces

There are no hardware interfaces for our project other than personal computers.

3.3 Software Interfaces

The software will be NS-3. This relies on Python, C++, and CMake. Additionally, it requires a linux environment which we run on a virtual machine. We will also use Wireshark and TCPdump to analyze packet captures to determine network communication.

3.4 Communications Interfaces

NS-3 is a local simulation and therefore our project requires no communication interfaces. However, should our project be adapted for hardware use, it will require those components.

4. System Features

4.1 Drone Swarm Simulation

4.1.1 Description and Priority

This feature is a high priority. However, most of the simulation was already built [2]. Therefore, we need to only adapt it to our needs and ensure working code.

4.1.2 Stimulus/Response Sequences

Most of the time, NS-3 must be rebuilt to allow for changes in the code. This means that alteration and such must be allowed and be shared by our team. The response time should be realistic between users.

4.1.3 Functional Requirements

REQ-1: The code simulations the communication and network environment of a drone swarm.

REQ-2: The code allows us to change the configuration, network type, and complexity of the swarm to allow for diverse testing.

4.2 Cyber Attack

4.2.1 Description and Priority

This feature is a high priority. The simulation must account for a malicious attack with intent to destroy or disrupt drone swarm operations.

4.2.2 Stimulus/Response Sequences

We must be able to design an attack and execute or do so live, as an attacker would.

4.2.3 Functional Requirements

REQ-1: The code allows us to execute an attack at the start.

REQ-2: The code allows us to implement attacks during execution.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

<If there are performance requirements for the product under various circumstances, state them here and explain their rationale, to help the developers understand the intent and make suitable design choices. Specify the timing relationships for real time systems. Make such requirements as specific as possible. You may need to state performance requirements for individual functional requirements or features.>

5.2 Safety Requirements

There are no safety concerns with our project.

5.3 Security Requirements

There are no security concerns with our open-source research project.

5.4 Software Quality Attributes

Our product must be built in such a way that researchers and students with experience in networking and computer science may build upon work further in this field of research.

5.5 Business Rules

There are no business rules for this project other than the timeliness and commitment of our members.

6. Other Requirements

At this time, there are no other requirements.