System Design Document

For

Unmanned Aerial Systems Simulated Cyberwarfare

Team members: Ryan Casa, Aliya Trussell, Fahad Alkaabi, Stefan Miller

Version/Author	Date
1.0 Ryan Casa	10/7/2022

TABLE OF CONTENT

SYSTEM DESIGN DOCUMENT

1 INTRODUCTION

1.1 Purpose and 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.

1.2 Project Executive Summary

The project will incorporate several experimental techniques being researched and to produce an accurate drone swarm and conduct cyber attacks. Then it will produce data which we will analyze to determine the effects of those attacks.

1.2.1 System Overview

Our system will use Network Simulator 3 (NS-3) to produce a simulated network environment. This network environment will act like a drone swarm and communicate between the host node and each drone. Then, we will conduct various Cyber attacks to determine the effect they have on the drones.

1.2.2 Design Constraints

The only real constraint our project has is how difficult NS-3 is to use at first. Each members must spend a considerable amount of time learning and testing the software before we may produce real results.

1.2.3 Future Contingencies

Should our project fall behind, we will only limit the amount of diversity in our project by reducing the different cyber attacks and drone swarm types and configurations.

1.3 Document Organization

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.

1.4 Project 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/

1.5 Glossary

NS-3: Network Simulator 3

2 SYSTEM ARCHITECTURE

In this section, describe the system and/or subsystem(s) architecture for the project. References to external entities should be minimal, as they will be described in detail in Section 6, External Interfaces.

2.1 System Hardware Architecture

Personal computers will be used for this project, therefore, there is no hardware architecture.

2.2 System Software Architecture

We run NS-3 on Ubuntu virtual machines that rely on Python 3 and CMake to build the simulations.

2.3 Internal Communications Architecture

NS-3 uses the CPU on our computers to interface directly with itself.

3 HUMAN-MACHINE INTERFACE

Our product has no human to machine interface as we design the tests and simulations ourselves to run and analyze.

3.1 Outputs

The main output produced by NS-3 is a packet capture. These packet captures details of all the packets sent across the network, including drops, acceptance, content, frequency, protocol, source, destination, and time. We will use these captures to analyze the simulation.

4 DETAILED DESIGN

This section provides the information needed for a system development team to actually build and integrate the hardware components, code and integrate the software modules, and interconnect the hardware and software segments into a functional product. Additionally, this section addresses the detailed procedures for combining separate COTS packages into a single system. Every detailed requirement should map back to the FRD, and the mapping should be presented in an update to the RTM and include the RTM as an appendix to this design document.

4.1 Hardware Detailed Design

There are no hardware designs for this project.

4.2 Software Detailed Design

Due to the nature of NS-3, each simulation must be designed, compiled and built before being run and then will be analyzed. This requires that we use C++ as Python only has limited support in the most recent version of NS-3.

4.3 Internal Communications Detailed Design

If the system includes more than one component there may be a requirement for internal communications to exchange information, provide commands, or support input/output functions. This section should provide enough detailed information about the communication requirements to correctly build and/or procure the communications components for the system. Include the following information in the detailed designs (as appropriate):

- The number of servers and clients to be included on each area network
- Specifications for bus timing requirements and bus control
- Format(s) for data being exchanged between components
- Graphical representation of the connectivity between components, showing the direction of data flow (if applicable), and approximate distances between components; information should provide enough detail to support the procurement of hardware to complete the installation at a given location
- LAN topology

5 EXTERNAL INTERFACES

External systems are any systems that are not within the scope of the system under development, regardless whether the other systems are managed by the State or another agency. In this section, describe the electronic interface(s) between this system and each of the other systems and/or subsystem(s), emphasizing the point of view of the system being developed.

5.1 Interface Architecture

In this section, describe the interface(s) between the system being developed and other systems; for example, batch transfers, queries, etc. Include the interface architecture(s) being implemented, such as wide area networks, gateways, etc. Provide a diagram depicting the communications path(s) between this system and each of the other systems, which should map to the context diagrams in Section 1.2.1. If appropriate, use subsections to address each interface being implemented.

5.2 Interface Detailed Design

For each system that provides information exchange with the system under development, there is a requirement for rules governing the interface. This section should provide enough detailed information about the interface requirements to correctly format, transmit, and/or receive data across the interface. Include the following information in the detailed design for each interface (as appropriate):

- The data format requirements; if there is a need to reformat data before they are transmitted or after incoming data is received, tools and/or methods for the reformat process should be defined
- Specifications for hand-shaking protocols between the two systems; include the content and format of the information to be included in the hand-shake messages, the timing for exchanging these messages, and the steps to be taken when errors are identified
- Format(s) for error reports exchanged between the systems; should address the disposition of error reports; for example, retained in a file, sent to a printer, flag/alarm sent to the operator, etc.
- Graphical representation of the connectivity between systems, showing the direction of data flow
- Query and response descriptions

If a formal Interface Control Document (ICD) exists for a given interface, the information can be copied, or the ICD can be referenced in this section.

6 SYSTEM INTEGRITY CONTROLS

Sensitive systems use information for which the loss, misuse, modification of, or unauthorized access to that information could affect the conduct of State programs, or the privacy to which individuals are entitled.

Developers of sensitive State systems are required to develop specifications for the following minimum levels of control:

- Internal security to restrict access of critical data items to only those access types required by users
- Audit procedures to meet control, reporting, and retention period requirements for operational and management reports
- Application audit trails to dynamically audit retrieval access to designated critical data
- Standard Tables to be used or requested for validating data fields
- Verification processes for additions, deletions, or updates of critical data

Ability to identify all audit information by user identification, network terminal identification, date, time, and data accessed or changed.