SOFTWARE REQUIREMENTS SPECIFICATION

for

Security for Robotics

Version 1.0 Final approved

Prepared by
Emily Longman, Zach Rogers, Dominic Giacoppe

June 11, 2017

Contents

1 Signatures

_____ _____

Sponsor Date

_____ _____

Group Member Date

_____ _____

Group Member Date

_____ _____

Group Member Date

2 Abstract

In drones and other networked robotics there is a broad array of security vulnerabilities that can be leveraged in an attack, leaving the potential for disaster. To attempt to prevent and mitigate these we evaluated ROS on a drone to find security holes and document them. The different vulnerabilities found were categorized into malware, sensor hacks, network and control channel attacks, and physical attacks. For some of these attacks were able to implement solutions, which were also documented. These findings and any solutions will be added to an ongoing academic effort to make robotics more secure.

3 Introduction

3.1 Purpose

To define the requirements and deliverables for Group 50's capstone project to our sponsor, Vedanth Narayanan.

3.2 Scope

We are to find security vulnerabilities in ROS/SROS, document these vulnerabilities, and if possible, produce patches for anything we find. Any patches produced will be submitted to the ROS project. Our testing will be focused around ROS/SROS running on a drone, and we will see if we can compromise that drone based on our findings.

3.3 Definitions

- Vulnerability: Any exploitable piece of code or system that would allow unauthorized users to interact with/damage/control the system, especially in a malicious manner.
- ROS: Robot Operating system, as found at link
- SROS: Secure ROS; a project based on ROS with the goal of implementing various security standards.
- Reliability: Consistently performs according to its specifications.
- Integrity: Maintaining and assuring the accuracy and consistency of data.
- Authentication: Any process where a system verifies the identity of a user who wishes to access it.

3.4 Overview

This document gives an overall agenda for our research. It is divided into 3 main sections: Section 1 is the introduction and purpose of this document, along with some acronym definitions. Section 2 contains the overall description of the goals we hope to achieve in our research. Section 3 lists our specific goals for our research.

4 Overall Description

4.1 Product Perspective

All software developed by Group 50 should have 2 objectives: 1. Fixing a specific, known vulnerability in ROS/SROS 2. Be lightweight enough that the implementation doesn't drastically affect the overall operation of the robot. Ideally, any code produced would be later incorporated into ROS itself, and not an external layer or program.

4.2 Product Functions

The main product we will be producing is research documentation on the potential security exploits in ROS. This will hopefully be used in future research and development of a more secure ROS. If we are able to exploit a vulnerability and create a functional patch for it, the changes will be submitted to the SROS project.

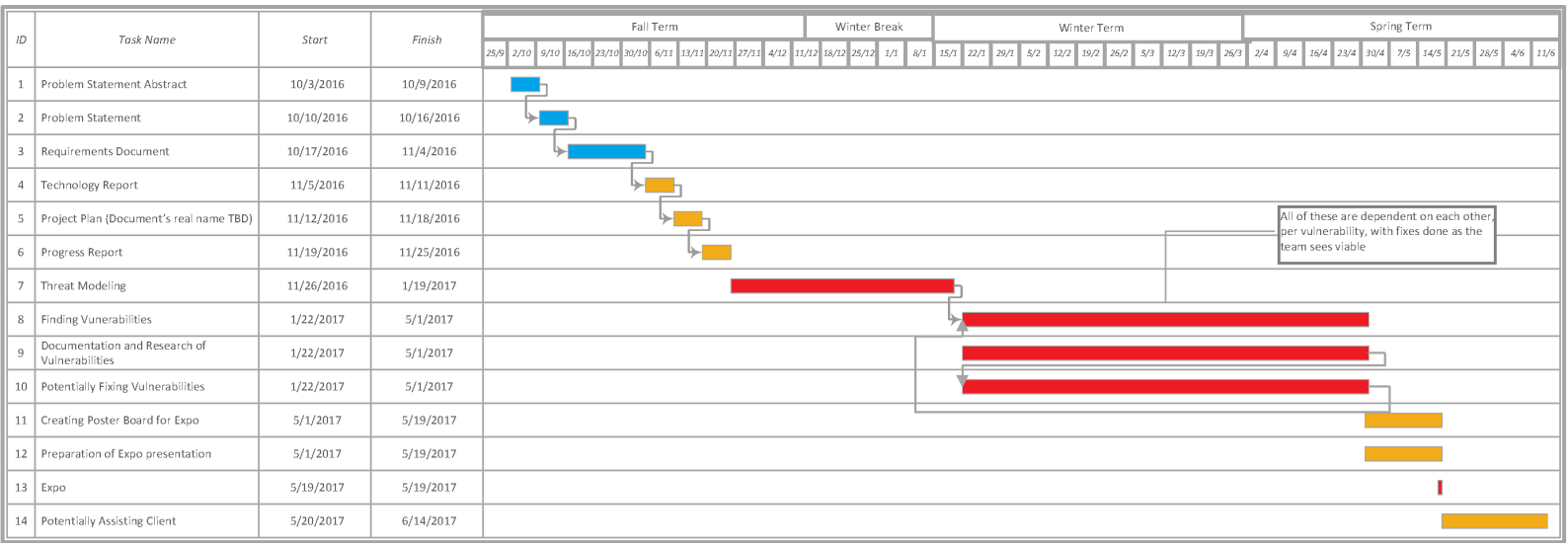
4.3 Constraints

Any work done on vulnerabilities needs to be appropriately documented and preferably published upstream to ROS/SROS as a whole.

4.4 Assumptions and Dependencies

There are only 3 real dependencies for this project at time of writing. First, we must complete our threat analysis before we start looking for threats, so that we have a general idea of where to investigate further. Second, we need to find a vulnerability before we can document it, or fix it, for obvious reasons. Lastly, we must have some sort of vulnerability at least documented before we prepare for Expo, or we won't have anything to present on.

4.5 Apportioning of Requirements



ID	Task Name	Start	Finish
1	Problem Statement Abstract	10/3/2016	10/9/2016
2	Problem Statement	10/10/2016	10/16/2016
3	Requirements Document	10/17/2016	11/4/2016
4	Technology Report	11/5/2016	11/11/2016
5	Project Plan (Document's real name TBD)	11/12/2016	11/18/2016
6	Progress Report	11/19/2016	11/25/2016
7	Threat Modeling	11/26/2016	1/19/2017
8	Finding Vunerabilities	1/22/2017	5/1/2017
9	Documentation and Research of Vulnerabilities	1/22/2017	5/1/2017
10	Potentially Fixing Vulnerabilities	1/22/2017	5/1/2017
11	Creating Poster Board for Expo	5/1/2017	5/19/2017
12	Preparation of Expo presentation	5/1/2017	5/19/2017
13	Expo	5/19/2017	5/19/2017
14	Potentially Assisting Client	5/20/2017	6/14/2017

Figure 4.1: Gantt Chart

5 Specific Requirements

5.1 Software Interfaces

As we are looking for vulnerabilities in ROS, all code produced must be compatible with it or integrated into it. The version of ROS we will be using is Kinetic Kame, the current long-term support version of ROS.

5.1.1 Communications Protocol

ROS uses 2 major forms of communication. Internally, ROS has the publisher subscriber system, which works basically like a socket system. Publishers export data, and anyone who subscribes to that publisher receives the data, with no limit to the number of subscribers or any authentication on who can subscribe. There is also normally some sort of wireless/wired connection to a base station, which controls the starting and stopping of the robot. These generally take the form of a standard LAN connection, although with extra effort more complicated setups are possible.

5.2 External Interfaces

The external interface we will be working with is a DJI FlameWheel 550 HexCopter. Included on it is a Beaglebone Black with a PixHawk v1.6 Fire cape mounted on it. The PixHawk has a number of sensors, and interfaces directly with our 3DR GPS mounted unit. There is also a controller which communicates with the drone over 2.5 Ghz RF, along with a 900 Mhz MAVLink telemetry radio, that communicates with our ground station. The ground station is a computer running Mission Planner, a misison control software that allows us to define flight paths, and read telemetry data from the drone.

5.3 Functions

Any vulnerabilities found should in some way compromise the functionality or integrity of the robot. In turn, any fixes created for said vulnerabilities should prevent those from being compromised.

5.4 Specific Requirements

As previously stated, at the moment Group 50 is in the process of finding specific vulnerabilities. When we do find one, we will produce documentation outlining at least but

not limited to: Our operating environment, the type of attack, the particular system/-piece of code attacked, the success rate of the attack, the result of the attack, and the potential fix to prevent the attack.

5.5 Software Attributes

5.5.1 Reliability

The reliability of ROS is what we will be trying to ensure, specifically that of it's security. We also want our documentation of any found exploits to be clear and readable for anyone using them in the future.

5.5.2 Availability

ROS and the potential exploits that we will be studying is open source, so it is available to anyone to confirm. If we are able to create a fix or patch for any of these we would contribute them to the open source SROS project.

5.5.3 Security

Security is the backbone of our entire project, we will be working specifically on finding potential security holes in ROS. If possible we hope to be able to create a patch for ones we are able to exploit and help to improve the security for everyone.

5.5.4 Maintainability

Because ROS is actively supported the software is maintained. If we produce a patch we would submit it to the SROS project and have it maintained within that.

5.5.5 Portability

ROS itself is very portable and we will be putting it onto our beaglebone. This also means that if we created a patch it would need to also be portable to the SROS project.