SOFTWARE REQUIREMENTS SPECIFICATION

for

Security for Robotics

Version 1.0 Final approved

Prepared by
Emily Longman, Zach Rogers, Dominic Giacoppe

November 4, 2016

Contents

1 Signatures

_____ _____

Sponsor Date

_____ _____

Group Member Date

_____ _____

Group Member Date

_____ _____

Group Member Date

2 Introduction

2.1 Purpose

<To define the requirements and deliverables for (team name)'s capstone project to our sponsor, Vedanth Narayanan.>

2.2 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.>

2.3 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.

2.4 References

<blah blah blah>

2.5 Overview

<blah blah blah>

3 Overall Description

3.1 Product Perspective

<All software developed by (Team name) 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.>

3.2 Product Functions

<blah blah blah>

3.3 User Characteristics

<blah blah blah>

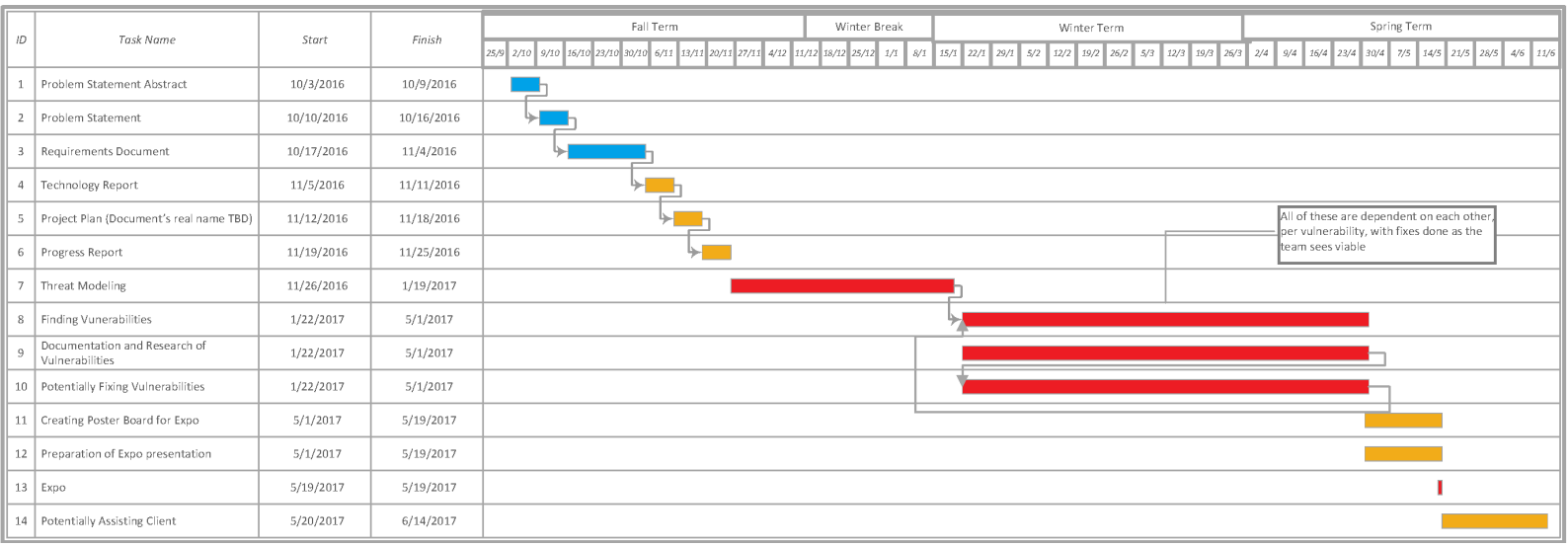
3.4 Constraints

<blah blah blah>

3.5 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 vunerability before we can document it, or fix it, for obvious reasons. Lastly, we must have some sort of vunerability at least documented before we prepare for Expo, or we won't have anything to present on.>

3.6 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 3.1: Gantt Chart

4 Specific Requirements

4.1 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.>

4.1.1 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. (SOURCE)>

4.2 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 firecap mounted on it. The sensors it has are the same as that of a regular pixhawk, which includes a gps and 9DOF.>

4.3 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.>

4.4 Specific Requirements

<As previously stated, at the moment (Team Name) 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.>

4.5 Software Attributes

4.5.1 Reliability

<blah blah blah>

4.5.2 Availability

<blah blah blah>

4.5.3 Security

<blah blah blah>

4.5.4 Maintainability

<blah blah blah>

4.5.5 Portability

<blah blah blah>