Project Plan

Document Number: [1.0]

9, February, 2020

OpenVPN SDN Management System

Members

KeAndre Graham

Hayes Roach

Christian Strahl

Brianna Howard

Alan Castle

Client

Cybriant

Table of Contents

1. Introduction

This Project Plan describes the scope, context, stakeholders, activities, resources, schedule, constraints, risks, quality goals, and processes. It is intended to provide a guide to how development of the product will be managed. At a high level the product is a Software Defined Network (SDN) for managing VPN connections.

2. Project Overview

2.1 Scope

2.1.1 System Overview

The software is designed to be an implementation of an SDN with a web GUI control panel. The software is initially being designed for internal use by the client and project sponsor Cybriant, however there are plans to make the project open source so that other entities may use the software for their own implementation.
For additional context refer to the Request for Proposal submitted by Cybriant.

2.1.2 Document Overview

This document is to identify items of importance to the project and potential ways to address these items. These items include, scheduling, risks, team skill, project features, scope, and deliverables, as well as resources for managing the project and resulting product.

2.2 Current System or Situation

2.2.1 Background, Objectives, and Scope

The client currently pays to use proprietary software to accomplish the same tasks that the feature list of this project would deliver. However, said proprietary software is expensive, difficult to set up, and is lacking in configurability. What the proprietary software does provide is the ability to define a group of individual servers and set up a connection between these servers with set rules. These are each features that this project must have as well.

2.2.2 Operational Policies and Constraints

The client, Cybriant, as a security company requires the software produced in this project to fulfill the highest standards in regards to reducing and minimizing vulnerabilities associated with connecting to external networks. A major issue could arise from the fact that the client connects to several external networks at a time, because of this there is the possibility that malware from one of these external networks could use the software as a hub to disseminate itself to all other networks the client is connected to. As a security company whose customers trust

2.2.3 Description of Current System or Situation

The system is implementing a web application using the RESTful architecture. As such the backend of the application is considered stateless, as all contextual information is kept on the client side. The client includes the context in requests, from which the backend can decide whether or not to process a request and how to do so. The system should also be database driven, in that the system should be able to publish updates to the OpenVPN server whenever a request alters the ruleset saved in the database. As previously mentioned many of the requirements for the system are similar to those found in the proprietary software that the client is currently licensing, however the product of this project should be easier to use and should be easier to configure to meet the clients need as a business.

3. Development Background/Approach

The OpenVPN Management System is currently being developed to run on both Linux and Windows machines as the targeted platform. However, the development environment for this phase of the project is being developed with Linux in mind as a first priority and the distribution of the OS is Ubuntu. The Graphical User Interface (GUI) of the service will utilize Flask as the web framework and Envoy for the web service. MariaDB will be used for the data repository and the latest version of Python (Python 3.8.1) will be used to develop some of the scripts for the system that will run in the background.

The team for this project consists of a variety of people with experience in the industry dealing with network security and data management. Also some have experience developing in a Linux environment and can code in Python. Each member has experience working in an Agile environment and that approach will be used to deliver on each deliverable made during the iterations of this project.

3.1 High Level Estimates

2 weeks will be the minimum time it will take to complete each phase of the project. If it happens to carry over, the time will be documented and adjustments will be made to the schedule. The lines of code will vary depending on which Sprint is being worked on, but it is estimated to be more than a 1000 lines of code. Steady communctication will be kept with the stakeholders to ensure we are still in the realm of meeting their requirements and new documentation collected will be stored in a shared folder that is viewable to members.

3.2 Key Contacts and Stakeholders

Company: Cybriant

Stakeholders

1. Name: Sean Mitchell

   Role: Project Manager, CAPM

   Phone Number: 470-443-9286

   Email: sean.mitchell@cybriant.com

2. Name: Andrew Hamilton

   Role: Chief Technology Officer (CTO)

   Phone Number: 423-827-7134

   Email: andrew.hamilton@cybriant.com

3. Name: Byron Deloach

   Role: Director of Adaptive Services

   Phone Number: 423-580-7070

   Email: byron.deloach@cybriant.com

4. Features, Primary Deliverables, and External Commitments

4.1 Feature List

- In-service software updates: significantly reduced planned downtime
- OpenVPN integrated into a management system
- Dynamic package deployment system for both Linux and Windows
- Firewall polling mechanism
- Rest API based GUI
- OSQuery parsing and logging

4.2 Customer Deliverables

[Provide a list of the final delivery dates along with a description of the contents of each delivery (e.g. user guide)]

Overview of Tasks

The OpenVPN SDN Management System team has organized its detailed statement of work into seven (7)

major implementation tasks. A summary of each task is provided below.

1. Deliverable 1: February 9, 2020 - Project Initiation and Project Plan: The team's expectations regarding the project kick-off and management.

2. Deliverable 2: February 14, 2020 – Web interface design: the team's expectations regarding the developing and detailing of the web GUI plans for designing the System to meet the needs of Cybraint.

3. Deliverable 3: February 28, 2020 - Database Development / Linux Package Configuration: expectations regarding the development and/or configuration of the system to meet Cybraint's needs through execution of the deliverable.

4. Deliverable 4: March 13,2020 – OpenVPN creation and configuration of the system to meet Cybraint's needs through execution of the deliverable.

5. Deliverable 5: March 27, 2020 – Windows Support: Windows Package Configuration: expectations regarding the development and/or configuration of the system to meet Cybraint's needs through execution of the deliverable.

6. Deliverable 6: April 20, 2020 – System Testing: testing of the System developed/configured in Task 2,3,4,5 to ensure that it meets the needs of Cybraint.

5. Project Schedule

5.1 Major Project Milestones

Date	Milestone/ task	Deliverable	Remarks
01-17-2020 to 01-31-2020	Sprint 1	Research Into OpenVPN, Python, IPTables, Ubuntu, Web Frameworks	
01-31-2020 to 02-14-2020	Sprint 2	Setup Web Framework with Flask	
02-14-2020 to 02-28-2020	Sprint 3	Database Design, Linux Package Creation	
02-28-2020 to 03-13-2020	Sprint 4	VPN creation and configuration	
03-13-2020 to 03-27-2020	Sprint 5	Certificate management and WinDows support	
04-06-2020 to 04/20/2020	Sprint 6	Testing and Polishing	

5.2 Project Status Tracking & Working Meeting Minutes

For the duration of the project, 2 meetings will be held every week with every member; one on Tuesday and the other on Thursday. Tuesday meetings will be conducted online where each member will discuss what they did since the past meeting and problems they encountered. Tuesday meetings are meant to be just a quick chat where members will casually discuss what they've done. Thursday meetings are longer and will be held in person in an agreed upon location where members touch basis on any progress made, status changes or potentially show demos of working concepts. Also plans for the next weeks meetings will be made during that time.

Thursday meetings will be tracked by a member of the group through a google docs where discussion items and anything important said will be logged. Also the software tracking collaboration tool Zoho will be used to log actions items and keep track of the project completion

6. Project Work and Product Estimates

Inputs to estimation include:

1) Deliverables listed in Section 4
2) Major Milestones listed in Section 5
3) Tasks and/or sub-tasks identified in the detailed project Schedule.

6.1 Estimate Summary

The tables below summarize the product size and effort estimates:

*Effort measured in total person hours of the team

Project	Estimate Attributes		
	Size		
WBS areas	**Unit of Size**	**Size**	**Effort**
Total Requirements Effort (includes feature-related **and** "other" (non-feature) Requirements work)			150
Feature Related Requirements Size and Effort Totals	PAGES	15	100
Total Development Effort (includes feature-related **and** "other" (non-feature) Development work)			500
Feature Related Development Coding Size and Effort Totals	LOC	2,000	300
Feature Related Development Documentation Size and Effort	PAGES	25	200
Total Testing Effort (includes feature-related **and** "other" (non-feature) Testing work)			150
Feature Related Testing Size and Effort Totals	TEST CASES	100	50

7. Project Resource Requirements

7.1 Staffing/ Skill Requirements

Chris Strahl

Role: Team Leader

Responsibilities:

Serves as primary point of contact with Client
Oversee planning and work distribution
Provide overall vision and direction of development

Critical Skills:

Knowledge of Web Architecture
Knowledge of Networking Fundamentals
Knowledge of Databasing Principles
Experience Leading Group Projects

Skill Gaps:

Depth of knowledge in Networking and Databasing
Experience in Python

Hayes Roach

Role: Development Manager

Responsibilities:

Track development efforts
Serve as DRI for Front-End

Critical Skills:

Knowledge of Web Architecture
Knowledge of Front-End Design

Skill Gaps:

Networking knowledge
Experience in Python

Keandre Graham

Role: Quality & Process Manager

Responsibilities:

Tracks our processes to look for improvements
Serve as DRI for Back-End and Python Scripts

Critical Skills:

Knowledge of Python
Knowledge of Front-end

Skill Gaps:

Networking knowledge

Alan Castle

Role: Planning Manager

Responsibilities:

Help track schedule and task completion
Serve as DRI for Networking requirements

Critical Skills:

Knowledge of Networking
Knowledge of Databasing
Knowledge of Scripting

Skill Gaps:

Web Architecture knowledge
Experience in Python

Brianna Howard

Role: Support Manager

Responsibilities:

Provide support in completing tasks of other domains
Serve as DRI for Databasing

Critical Skills:

Knowledge of Databasing
Knowledge of Python and JavaScript Fundamentals

Skill Gaps:

Depth of knowledge in Python and JavaScript

7.2 Plan to Fill Skill Gaps

The primary mitigation strategy is to keep a constant line of communication between group members. This will facilitate transfer of knowledge so that all group members can develop expertise on the system as a whole. The team has also allocated the first sprint to research to rigorously study the domain knowledge required for the project.

8. Dependencies and Constraints

8.1 Constraints

- Project Deadline - There is a limited amount of time to build the project as it needs to be completed and in a presentable state by April 23rd, 2020.
- Domain Knowledge - Team members are not fully knowledgeable in networking or developing applications in python. Plans to understand these concepts can be found in section 7.2.
- Team Schedules - Team members have busy schedules due to other classes and work that limit the amount of time to work on the project.
- Travelling Onsite - Cybriant is about an hour away from Marietta which makes in-person meetings difficult to schedule.

8.2 Dependencies

- Stakeholders - Team members will contact stakeholders often to ensure that the project meets their wants and needs.
- OpenVPN - The project relies heavily on OpenVPN's software and documentation to create secure virtual private network connections. The team needs a firm understanding of how OpenVPN works before starting development.
- The team must finish building a dynamic package deployment system using Linux endpoints before the team starts on the windows version.

9. Risk Management

9.1 Initial Risk List

This is the initial risk list; the risks are listed in priority order from top to bottom. Descriptions are provided below the table. Risks with a pre-mitigation magnitude of 2.0 or below are not listed.

Risk number	Risk Priority (H, M, L)	Likelihood of Occurrence	Risk name: brief description	Mitigation Strategy "ACCEPTED" or "MITIGATED" with pointer to plan.
1	H		Misunderstanding requirements	Develop clear, complete, detailed, cohesive, attainable, and testable requirements that are agreed to by all players. In "agile"-type environments (fluid, changing continually), frequent coordination with customers/end users is necessary.
2	H		Schedule Risk: project schedule not thoroughly planned. Wrong time estimation.	Incorporate adequate time for planning, design, testing, bug fixing, retesting, changes, and documentation Reduce expansions in project scope Tracking of resources properly
3	H		Technical Risk: appearance of undesirable events, which may affect the development of the project	Maintaining the requirement of constantly Reducing the difficulty in project modules
4	L		Programmatic Risk: external risk that occurs outside the operational limits. Beyond the control of the program	Maintain the constant priority of the customer

				Constant project strategy
5	M		Operational: lack of knowledge of the developers, improper implementation of the process. Problems with the coordination of team	Addressing priority conflicts properly Planning resources properly Maintain communication throughout the team
6	H		Product Risk: system or software fails	Testing the developed product regularly Finishing the requirement for every phase
7	L		Budget Risk: miscalculation of budget estimation	Proper project preparation and resource allocation
8	M		Programming Risk: lack of knowledge of programming language	Proper research among the team Proper communication
9	M		Poor team dynamics:	Involve interactive team management to identify issues and act as facilitator to resolve team issues. Implement processes to escalate conflict resolution to proper personnel as needed
10	L		Friction between the project team and the customer	Establish clear lines of communication

				between the project team and the customer. Proactively manage communication. Develop issue logs and plans to track and resolve issues. Follow up on action items. Ensure all project status information is accurate and up-todate

10. Project Configuration and Data Management

10.1 Configuration Management

There are several management tools in place to facilitate the project. The primary of which being Zoho, an online project management tool provided by the client where tasks can be created, assigned, and tracked. Zoho also serves as a document repository. Additionally GitHub is a repository for code, allowing for code to be shared between team members and has tools for version control and code review. The development team will stay in contact with the client through regular online meetings using RingCentral, and is communicating internally both through weekly in person meetings and through messaging applications.

11. Project Process

11.1 Software Life Cycle Model

The Life Cycle Approach being used can best be described as the Spiral pattern. The Client has a set vision of features for the software to implement, however these features have been broken into several iteration phases. The goal of this project is to complete at least one

iteration, and ideally a second or third. At which point the Client plans to open up the project to the open source community so that other companies with similar problems to solve may implement and further iterate on the software. GitHub will be used as a code repository as GitHub serves as a home for many open source projects and has tools to allow the client to manage the roadmap of future iterations.

12. Referenced Documents

"What is OpenVPN?," OpenVPN. [Online]. Available: https://openvpn.net/faq/what-is-openvpn/. [Accessed: 04-Feb-2020].

"Request for Proposal, OpenVPN SDN Management System, Spring 2020. Available by contacting either Sean Mitchell (sean.mitchell@cybriant.com) or Chris Strahl (cstrahl@students.kennesaw.edu)

13. Glossary

Back-End - The server to which a website makes requests to. The server is in Python for this project

DRI - (Directly Responsible Individual) - The person held ultimately responsible for the quality of a particular domain.

Front-End - The code which is run in the browser to render a web site. HTML/CSS/JavaScript is used for this project.

GUI - (Graphical User Interface) - The user is able to interact with the system in a visual form usually with a mouse or by touch; the system is not exclusively through terminal commands.

OpenVPN - An open-source commercial software that implements virtual private network techniques to create secure point-to-point or site-to-site connections in routed or bridged configurations and remote access facilities.

SDN - (Software Defined Network) - A hub for configuring and routing VPN connections. The project as a whole can be considered an SDN.

VPN - (Virtual Private Network) - Extends a private network across a public network, and enables users to send and receive data across shared or public networks as if their computing devices were directly connected to the private network

14. Change Record

Version#	Date of Change	Reason for Change	Description of Change	Approved by:
0.1	2020/02/06	First Draft	Draft created for discussion with client	Chris Strahl
1.0	2020/02/09	Address Client's comments	Addressed clients comments and finalize draft	Chris Strahl Sean Mitchell