**SWE 4724**

OpenVPN SDN Management System
Team 6

Alan Castle

Keandre Graham

Brianna Howard

Hayes Roach

Chris Strahl

**Version: (1)** **Date: (03/03/2020)**

**Table of Contents**

[**1. Introduction**](#_heading=h.gjdgxs) **3**

[1.1 Purpose](#_heading=h.mmd24n155wam) 3

[1.2 Scope](#_heading=h.umqkqze0kiuj) 3

[1.3 Definitions, Acronyms, and Abbreviations.](#_heading=h.oaos16hisz8v) 3

[1.4 References](#_heading=h.o0y0sgg8vofx) 4

[1.5 Overview](#_heading=h.274q3vaep4oa) 4

[**2. The Overall Description**](#_heading=h.jl4pit6l410y) **5**

[2.1 Product Perspective](#_heading=h.702dm41lrvey) 5

[2.1.1 System Interfaces](#_heading=h.boo70xytp2m0) 5

[2.1.2 Interfaces](#_heading=h.9kqplig4oavq) 6

[2.1.3 Hardware Interfaces](#_heading=h.vykoizg1pmwt) 6

[2.1.4 Software Interfaces](#_heading=h.t0rq2g6exp3l) 6

[2.1.5 Communications Interfaces](#_heading=h.d36wkdgvcxqs) 6

[2.1.6 Memory Constraints](#_heading=h.b3gpecalblpj) 7

[2.1.7 Operations](#_heading=h.kg4bcc6uirzk) 7

[2.1.8 Site Adaptation Requirements](#_heading=h.bivz7x6hevf6) 7

[2.2 Product Functions](#_heading=h.kyc2s6gw3u3i) 7

[2.3 User Characteristics](#_heading=h.3zljpqeewmx) 7

[2.4 Constraints](#_heading=h.jpcmekmsthn7) 7

[2.5 Assumptions and Dependencies](#_heading=h.1nclh58u2u6n) 8

[2.6 Apportioning of Requirements.](#_heading=h.96md1mu1qnp6) 8

[**3. Specific Requirements**](#_heading=h.n6mke9h940rg) **9**

[3.1 Functions](#_heading=h.6qwow6erx5mp) 9

[3.2 Non-functional Requirements](#_heading=h.yhxowu8nctvw) 10

[3.3 Database Requirements](#_heading=h.eowvgzjdoqbh) 10

[3.4 Design Constraints](#_heading=h.qm4kgxyd5esi) 10

[3.4.1 Standards Compliance](#_heading=h.6wsrk5mnlpqc) 10

[3.5 Software System Attributes](#_heading=h.8dmf6axprfyl) 11

[3.5.1 Reliability:](#_heading=h.l4ahm3dxx5gt) 11

[3.5.2 Availability:](#_heading=h.9bns5fqio8l6) 11

[3.5.3 Security:](#_heading=h.9jn5foenvu5s) 11

[3.5.4 Maintainability:](#_heading=h.gvupig4sy0uo) 11

[3.5.5 Portability:](#_heading=h.ik5esih3dirj) 11

[**Change Management Process**](#_heading=h.mj5tb4spod10) **12**

[**Document Approvals**](#_heading=h.9m0hxxy1fixp) **12**

# 1. Introduction

## **1.1 Purpose**

The SRS document is meant for developers and project managers to see how the system will be built and function and what major components will be developed at launch. 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.

## **1.2 Scope**

The company Cybriant wants to create an OpenVPN Management System that goes beyond provisioning basic user VPNs, but can create Software Defined Networks (SDNs) between infrastructures. The VPN based SDNs would be able to control endpoint firewall rules and uni/bi-directional connectivity to the different infrastructures. Additionally, the system would utilize OSquery to report the health of the endpoint and allow or revoke access to SDN. The system will run on desktops either with a Linux or Windows operating system. The scope of this SRS will cover the basic features this system will have at its initial launch with room for other additions to be added down the line.

## **1.3 Definitions, Acronyms, and Abbreviations.**

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

Rest - (Representational State Transfer) - is a software architectural style that defines a set of constraints to be used for creating Web services

API - (Application Program Interface) - is an interface or communication protocol between different parts of a computer program intended to simplify the implementation and maintenance of software

Rest API - is an application program interface ([API](https://searchmicroservices.techtarget.com/definition/application-program-interface-API)) that uses HTTP requests to GET, PUT, POST and DELETE data

## **1.4 References**

1. “What is OpenVPN?,” OpenVPN. [Online]. Available: <https://openvpn.net/faq/what-is-openvpn/> [Accessed: 04-Feb-2020].
2. “Flask OAuth 2.0 Server”, [Online]. Available: <https://docs.authlib.org/en/latest/flask/2/>
3. “OAuth2 Server”, [Online]. Available: <https://flask-oauthlib.readthedocs.io/en/latest/oauth2.html>

## **1.5 Overview**

The rest of the SRS will be broken down into 2 important sections, the overall descriptions and the specific requirements. The overall descriptions will talk about the factors that affect the product and the requirements while the specific requirements will go into details about the features of the system and what it should and should not do. References to the frontend and backend of the system will also be referenced there as well.

# **2. The Overall Description**

This section will give an overview of the whole system. It will talk about the basic functionality this system has to offer to stakeholders and how it interacts with other systems. The constraints and assumptions for the system will also be discussed here as well.

## **2.1 Product Perspective**

This system is designed to be a web-based application that clients will use to login to get setup with the SDN for their network connections. For Cybriant, they will be able to view all the user’s in a server group with their configuration and be able to manage or update them accordingly. The general flow can be visualized as:



As mentioned before the clients will connect directly to the SDN from a frontend web application while the backend will pull from a data repository, rulesets and packages that will help define the connections that the clients will use. Cybriant will get a real time view of everyone connected and that will be separate from the client experience.

### **2.1.1 System Interfaces**

OpenVPN is used to dynamically create VPNs between endpoints across different cloud services and infrastructures. Our system will use python scripts to create the configuration files and certificates to setup these VPN connections.

###

### **2.1.2 Interfaces**

 The system will interact with its users primarily through a web-based application GUI that makes requests to our REST API. Since the REST API is separate from the frontend, this makes the system configurable in that it could be used with a mobile application, command line, or any type of software that can make requests to it. To be able to interact with the system, a user will need a valid access token to authenticate their account.

### **2.1.3 Hardware Interfaces**

The client plans to run the application in a cloud environment, therefore there are no hardware requirements.

###

### **2.1.4 Software Interfaces**

The client has specified the following software and interfaces be used.

* OpenVPN
	+ A well established and robust VPN protocol with which the client has familiarity leading to their choice.
* Python 3
	+ Similarly chosen for the clients familiarity with the language lending itself increased maintainability should the client need to change something after the project’s completion.
	+ Note that while the language was specified the framework for building the web functionality has been left to the development team.
* MariaDB
	+ The client’s SQL Server Syntax of choice.
	+ Note that versioning is of little concern as MariaDB has great backwards and forward compatibility.
* IPTables
	+ Also chosen due to the client’s familiarity and preference for Linux systems.

### **2.1.5 Communications Interfaces**

The system defines a network of servers connected using the OpenVPN site-to-site connection protocol. The system will act like a proxy router for connections between these “sites” or servers. These connections will be validated through a set of IPTable rules to verify that the requested connection is authorized. As such the system must abide by OpenVPN communication protocols and by IPTable syntactical standards.

### **2.1.6 Memory Constraints**

The client plans to run the application in a cloud environment, therefore there are no hardware requirements.

### **2.1.7 Operations**

The system is intended to be available nearly continuously. While changing certain security settings may require the system to restart, these restarts should be kept very brief and so the period of unavailability will be negligible.

### **2.1.8 Site Adaptation Requirements**

Adoption effort for the system will be minimal. Effort is not anticipated to be more than simple training on how to use the user interface.

## **2.2 Product Functions**

With the web-based application the user will be able to register online with a login name and password. The information will be verified with the information in the database and the user is appended. The user will be able to add, modify, or delete information pertaining to clients. The user’s client information will also be stored in a database, which will be updated after each change. The user will be able to control server connectivity uni-directionally or bi-directionally. The server connectivity will be displayed in a graph on the web application. The user will be able to control endpoint firewall rules. Rules will be stored in Iptables, which are connected to and will update the database system. The user’s clients will be notified via web portal if their system requires a new update. The web portal will provide functionality to manage the system and the servers.

## **2.3 User Characteristics**

The intended user is expected to be technically proficient in the basics of networking and network security. The user will be granted control to specify network nodes, connections between nodes, and the rules of connection, therefore some basic background knowledge is expected.

## **2.4 Constraints**

One major constraint for the Web Applications depends on how many people are connected to the SDN. It may take time to fetch the rulesets from the database to get them configured if multiple people are doing it at the same time which may cause latency. Another factor is also internet connection because this is an always online feature. The application is getting data from the database over the internet and without it, it won’t function. A minor constraint might be the operating system as well. Right now this is being developed in a Linux environment with the goal for it to be runned on Windows in the future, but there might be uncertainty with how the operating system will handle it, not to mention if this will be operating on MAC’s too.

## **2.5 Assumptions and Dependencies**

It is assumed that the alumni’s data will be made available for the project in some phase of its completion. Until then, test data will be used for providing the demo for the presentations. It is assumed that the client is familiar with an internet browser and familiar with handling the keyboard and mouse. The application will be running on a Linux operating system; therefore, it is assumed the users have access to the required operating system.

Since the application is a web-based application there is a need for the internet browser. It will be assumed that the users will possess decent internet connectivity. The application also requires the management of servers. As a result, it is assumed the client has their own server system or has access to an application, such as Amazon’s Azure.

## **2.6 Apportioning of Requirements.**

The project requested is of considerable scale and providing a robust, secure, and well designed system meeting all of the requested features would not be possible with the rigid time and resource constraints of the development team. The client recognizes this and has helped define release iterations for the project. The client has shared their plan to make the project open source to encourage further iterations once the development team has produced a solid base upon which to build.

The requirements discussed in this document primarily address the features prioritized for the first iteration which entails defining the OpenVPN network of Linux based servers and connection rules through a web GUI. The second iteration focuses on extending support to Windows based servers. The third iteration would primarily focus on “quality of life” and efficiency, such as including a mechanism to automatically publish updates to OpenVPN clients. The fourth and final iteration planned is to integrate a technology such as OSQuery to be able to query OpenVPN clients to determine health of server and to perform ad hoc queries and configurations.

# **3. Specific Requirements**

## **3.**1 **Functions**

1. The system shall allow the definition of a group of servers.
	1. Definition and viewing of server groups shall be available through a web based GUI.
	2. A group of server’s definition shall include all information required to connect to a server. This includes but may not be limited to:
		* IP range
		* Virtual IP range
		* General Connection Rules and Parameters
	3. The system shall allow server groups to be put into organizations for server groups that are controlled by the same company or real world entity.
2. The system shall optionally be able to create an install package for download when a new server group is defined.
	1. The system should also be able to optionally include in the install package the latest version of the OpenVPN Client.
3. The system shall deny connections between the server groups by default.
4. The system shall allow the definition of connections between server groups.
	1. Definition and viewing of connections shall be available through a web based GUI
	2. Connection shall include all information required to establish secure a secure connection between the server groups. This includes but may not be limited to:
		* Allowed directions of connection
		* Connection type allowed (TCP, UDP, SSH, etc.)
		* Connection / Request limit
		* Specific ports to block
	3. Connection rules shall be mutable and may be updated without adoption effort for either server group defined in the connection.
5. The system shall provide a graphical representation of server groups and the connections between them.
6. The system shall provide an authentication and log in system to prevent unauthorized access to view or edit items in the system.
	1. The system shall provide two authenticated roles one for creating, viewing, editing, and deleting items, and another role for viewing only.
	2. The system shall provide a list of users and their role in the web GUI.

## **3.**2 **Non-functional Requirements**

3.2.1 Performance

1. The system shall be usable by the appropriate members of the Cybriant team in an interactive format with little to no delays to input and network usage.
	1. As of yet, no quantifiable number of individual networks, users, or terminals have been suggested by the client.
2. The system shall be scalable in scope and compatible with Azure servers, with regard to server/client connections, bandwidth, storage, and storage.

3.2.2 Security

1. The System shall transmit all information securely with no changes to the original information.
	1. The connections must follow OpenVPN security protocols.

3.2.3 Reliability

1. The system shall be available to users 99% of the time.
	1. The system shall return to full operational status within 15 minutes should an interruption occur.

## **3.**3 **Database Requirements**

1. The system shall store the following data securely.
	1. Server and Client configuration and updates
	2. User data, including usernames and passwords, authorization levels, and log data
	3. Additional data dependent on the growth of scope of the system.
2. The system shall be the only entity to interact with the database.
3. The system shall utilize native MariaDB encryption tools.

## **3.**4 **Design Constraints**

### **3.**4**.1 Standards Compliance**

1. Hardware Limitation: Windows and Linux Operating System will be the primary environment the system will run on
2. Report Format: Logs for when connections are setup and made with a client will be documented
3. Application Memory Usage: Having more than 4GB of RAM
4. Internet Connection: System relies on having a stable internet connection

## **3.**5 **Software System Attributes**

These Software System Attributes are characteristics that the system must possess.

### **3.**5**.1 Reliability:**

The system, providing accurate and secure management of the server/client connections, shall use the VPN itself to be functional in the given environment, providing the intended services with constant, 24 hour/365 days of uptime, as the intended environment of the software is a 24-hour watch center. The reliability of the system must constantly be responsive to user changes of and accurately display the current connection status.

### **3.**5**.2 Availability:**

The system must be available to Cybriant users 80% of the time, with any downtime or interruptions being quickly manageable to return the system to full operational status. As the system directs the management of the VPN and not the operational status of the VPN itself, the downtime of the system is never to exceed 4.8 hours.

### **3.**5**.3 Security:**

Information transmission should be securely transmitted to the server with no changes to the original information. Additionally, clients should not have reach-back capabilities. The connections must all be secure and follow OpenVPN security protocols.

User accounts of the system will be based upon the appropriate level of privilege assessed and assigned by the system administrator. Each account shall be password protected, and logins and all connection changes and requests will be logged.

###

### **3.**5**.4 Maintainability:**

The system will come with documentation, as well as a basic tutorial or walkthrough. Additionally, the system itself will support network monitoring and event logging.

### **3.**5**.5 Portability**:

The system itself will be developed in the Python language, initially for the Linux Operating System and the appropriate libraries will be made available for a Windows version to be functional. The database of choice will be MariaDB, which makes accomodations for streamlined migration, should the need arise.

# **Change Management Process**

Due to the nature of the project, there are already more requirements than there is anticipated to be time to complete. Therefore new requirements added to the scheduled phases must be deemed by the client to be of the utmost criticality. If such an activity arises the development team will need to give a time estimate on the new requirement and submit an updated schedule for what other requirement will likely have to be cut from scope to fulfill the request. The client can then decide if this is an acceptable loss to gain the requested feature.

# **Document Approvals**

|  |  |
| --- | --- |
| *Client Contact*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_Sean Mitchell DateCybriant Project Manager | *Development Team Contact*Christian Strahl 2020 Mar 1\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_Christian Strahl DateKSU Project Team Lead |
|  |  |