

SDS Report

Bot-NET

Cody L, John D, Zak A and Will N | Software Development and Security | Report Date: 24-05-2020

# TABLE OF CONTENT

1. INTRODUCTION

* 1.1 Purpose
* 1.2 Scope
* 1.3 Overview

1. PROJECT DESCRIPTION

* 2.1 Project Perspective
* 2.2 Project Functions
* 2.3 Constraints

1. SPECIFIC REQUIREMENTS

* 3.1 External Interface Requirements
* 3.2 Software Features
* 3.3 Design Choices and Constraint

1. TASK ASSIGNMENTS
2. APPENDICES
3. REFERENCES

## INTRODUCTION

* 1. Purpose
* Our group project aims to create an interactive terminal-based command and control Bot-net coded in c, our programs are targeting the Linux operation system. The system infrastructure will include master and slave function where the Bot/Client machines will have functionality to create persistence and receive remote command execution from the master/Server.
  1. Scope (need help with filling in this part):
* “Identify the software product(s) to be produced by name. Explain what the products will and will not do. Describe how the software will be used, and identify relevant benefits, objectives, and goals.”
  1. Overview
* This report will contain the demonstration of each functionality included in our completed Bot-net project and their purposes justified. Each member’s progress and their assigned tasks will also be included for easier marking. Evaluations of each aspect such as the project’s features and constrains will be outlined. Alongside, Design choices that we have made throughout out the project are also presented to demonstrate that we have a clear understanding of what decisions are most suitable for our project.

## Project Description

2.1 Interfaces: (Is this relevant to our project? Justification needed.)

* System: List each system interface and identify the related functionality of the product.
* User: Specify the logical characteristics of each interface between the software product and its users (e.g., required screen formats, report layouts, menu structures, or function keys). Specify all the aspects of optimizing the interface with the person who must use the system (e.g., required functionality to provide long or short error messages). This could be a list of do’s and don’ts describing how the system will appear to the user.
* -Hardware: Specify the logical characteristics of each interface between the software product and the hardware components of the system. This includes configuration characteristics (e.g., number of ports, instruction sets), what devices are to be supported, and protocols.
* Software: Specify the use of other required software products (e.g., a DBMS or operating system), and interfaces with other application systems. For each required software product, provide the following:

Name

Mnemonic

Specification Number

Version Number

Source

For each interface, discuss the purpose of the interfacing software, and define the interface in terms of message format and content. For well-documented interfaces, simply provide a reference to the documentation.

* Communication:

Specify any interfaces to communications such as local area networks, etc.

2.2 Project Functions:

* Connectivity: Basic connectivity achieved between the clients and the server.
* Communication: Having the ability to execute shell/bash command on slave/client machines from the master/server (Remote Command execution).
* Interface: A basic user-friendly interface on the server machine.
* Persistence: Slave/Client machines are persistence through the reboot of the device.
* Able to observe output of the executed programs by the slave from the master. (is this feature included in our programs?)
* Network should have measures to evade firewall detections. (again, is this feature included?)

### 2.3 Constraints

* Need help filling in

Describe any other items that will constrain the design options, including

regulatory policies

hardware limitations

interfaces to other applications

parallel operation

audit functions

control functions

higher-order language requirements

signal handshake protocols

reliability requirements

criticality of the application

safety and security considerations

## specific requirements

### 3.1 External Interface Requirements (do we need this?)

### 3.2 Software Product Features

* 3.2.1 Connectivity Features

Feature 1:

Repeat subsections at this level and below for each associated functional requirement.

Each functional requirement should be described in natural language, in the four subsections below. Functional requirements include:

validity checks on inputs

exact sequencing of operations

responses to abnormal situations, including error handling and recovery

effects of parameters

relationships of inputs to outputs, including input/output sequences and formulas for input to output conversion

|  |  |
| --- | --- |
| Introduction |  |
| Inputs |  |
| Processing |  |
| Outputs |  |

* 3.2.2 Communication Features

Feature 1:

|  |  |
| --- | --- |
| Introduction |  |
| Inputs |  |
| Processing |  |
| Outputs |  |

* 3.2.3 Interface Features

Feature 1:

|  |  |
| --- | --- |
| Introduction |  |
| Inputs |  |
| Processing |  |
| Outputs |  |

* Persistence Features

Feature 1:

|  |  |
| --- | --- |
| Introduction |  |
| Inputs |  |
| Processing |  |
| Outputs |  |

### 3.3 Design Choices and Constraints

## 4.0 task assignments

## 5.0 Appendices

## 6.0 References