Software Requirements

Software requirements specification for the IT 2750 – Python Scripting for Cybersecurity course. This document outlines the application to be built for your final project

# Goals

This project establishes a Python-based tool for improving cybersecurity practices by implementing automated functions. The tool has been designed to help users tackle their problems with safe password creation alongside contact identification in extensive text documents. The tool will help users create secure passwords which conform to modern security criteria through customizable password generation as users face difficulties generating passwords with adequate strength. OSINT feature represents an efficient solution to eliminate manual labor for extracting phone numbers and email addresses from files. The project delivers an easy-to-use platform which enables security functions for both personal and professional requirements. This tool helps users improve their security while also saving their time through automation of the two listed functions.

# User Personas

The system needs to accommodate Sarah who works as a freelance graphic designer and manages multiple clients and their contact information throughout different files. The tool allows her to rapidly gather phone numbers and emails from project documents which then enables better client data organization. James represents the second user who aims to develop his cybersecurity behaviors throughout his college career as a computer science student. The password generator tool enables him to develop complex passwords which improve the security of his college logins and personal profile access. The office manager Maria needs an easy system to both verify and gather contact information directly from spreadsheet files. OSINT enhances her work process by making data review operations more efficient. The software matches David's needs because he operates his small business through tech means while seeking password security without outside password management services. The password generator function enabled David to renew his login passwords on various systems throughout each day. A basic yet powerful cyber security solution using Python technology meets different user requirements.

# User Stories

As a freelance graphic designer Sarah manages different projects while she handles client documents that exist without proper structure. The extraction of phone numbers and email addresses from text files will help me establish quick contact organization for clients. The creative focus of her work drives her to use tools that streamline administrative processes. The tool decreases both time consumption and the chance of missing vital contact information which exists deep within extensive files. The OSINT feature simplifies a job which normally requires her to spend long hours doing manual scanning. The professional demands efficiency through accurate and speedy operations which fulfills her needs for better productivity. The system enables her to process any document which results in arranged contact information suitable for adding to her client database. The tool maintains her professional appearance when addressing clients while improving both her communication methods and workflow efficiency.

The computer science student James joins nearly every platform dealing with coding as well as the academic portal and technical communities. I need to develop secure password strings for keeping my school accounts together with my personal online accounts protected against hackers. Although aware of weak password dangers he experiences difficulty creating new secure passwords each time he needs one. Password generator enables him to establish particular password specifications including length and symbol requirements along with upper-case character usage. Being a tech-savvy user allows him to choose password strength settings based on varying security needs between platforms. The tool functions as a security measure that lets him generate passwords and develop improved online behavior. The tool supports his cybersecurity education because it demonstrates to him how programming code provides solutions to everyday issues. The feature delivers value through education and practicality to offer support for his academic progress together with his online safety needs.

# Feature Descriptions

Users can generate robust passwords through the built-in password generator that produces codes based on their specified security requirements. Users can reach the password generator function by using a text-based menu that appears after script execution. After selection of the program the user must provide input regarding password length together with specifics about uppercase characters and number and special character usage. A random password emerges from the script that satisfies the security specifications before showing the selection on screen. The system introduces a solution to address both weak passwords and password re-use vulnerabilities that represent major security risks. This feature accepts two inputs which are password length and specified character types and produces one output as secure password string. The system presents two options which are another password generation or main menu selection to the user.

Through its OSINT function the program recognizes and retrieves telephone numbers and email addresses inside written text files. From the main menu users select this feature which leads them to enter the text document file path for the scanning process. The program takes input from the file while running regular expression detection to extract verified phone numbers and email addresses from it. The script shows output results on the screen after finishing the scan while offering an optional feature to generate new output files for later use. The program reduces the tedious process of manual contact search thereby making the feature beneficial mostly for business and administrative purposes. Users can provide the file document path of plain text content to produce a clean phone number and email address list. The user can either reselect files or go back to the menu interface to review the output analysis. The function significantly boosts the process of finding important data while simultaneously enhancing organizational capability.

# Wireframes/Walkthrough

The script initiates a terminal display of its text-based main menu after launch. The application presents three selectable options for users to pick from which include (1) Generate Secure Password and (2) Extract Phone Numbers and Emails from File and (3) Exit. The program asks users to specify their passcode length then their requirements for uppercase characters along with numbers and special elements when they opt for the first menu choice. The application implements an orderly sequence of commands which makes preference input straightforward for users. Users can access two options after providing screen entry requirements namely their newly generated password followed by an alternative generation or main menu return option. This system displays a simplified structure that enables James and other users to make quick work of their secure password development. This password generator displays no unnecessary elements since its primary goal is to provide clear functionality. Technical limitations become irrelevant because this feature allows all types of users to smoothly navigate its functionality with self-assurance.

From the main menu selection of option two the terminal requires the user to enter the path for their text document for scanning. The file check is finalized and the system starts phone number and email address detection through regular expressions. The scan process concludes by showing its findings in distinct sections that contain both phone numbers and email results. A new text file option or main menu return choice is presented to users after the results are displayed. The results display presents a design for better readability since phone numbers and email addresses are spaced equally across the interface to prevent clutter. The application facilitates simple access to contact organization for user types like Sarah and Maria through its user-friendly design. The program contains straightforward input instructions and easy-to-read output information which simplifies user understanding. This system design guides users to achieve functional tasks by maintaining their focus on efficient performance.

# Non-Functional Requirements

The application development utilizes Python programming language along with its simple and robust standard libraries. The entire application functions through command line instructions because it does not contain any graphical user interface parts. Users can execute this software on different operating systems because it maintains lightweight functionality across Windows, macOS and Linux platforms. The script will avoid incorporating system-specific libraries but may include them when new features need OS-specific functionality. All standard Python libraries including random, string, and re should make up the only dependencies for initial script development. One terminal command should execute the script easily while users do not require administrative permissions to run or install it. The code structure will be clean while providing effective comments and being built for ease of maintenance to enable future development work. The program will be equipped with fundamental error management and validation checks to avoid unexpected system crashes.

# Future Iterations

The project expansion beyond three weeks would allow us to incorporate various updates that would enhance usability and end-user satisfaction. The addition of a graphical user interface with Tkinter or PyQt libraries would be a future implementation to serve users who work better with visual interfaces. Thesoftware could benefit from file encryption capabilities through implementation of AES algorithm instead of utilizing basic substitution ciphers. The system should permit users to both create and access generated passwords through a secure editor. The OSINT component needs future version improvements because fundamental scan capabilities should enable multiple document types from libraries such as PyPDF2 and python-docx. Users would benefit from added recording features to monitor both scanned data information and password generation activity records. The implementation of multi-language support aims to extend accessibility features for the application. This project will evolve toward becoming a desktop program that includes modular characteristics.