Final Project Methodology Report

Scripting for Cybersecurity

Professor. Hansen

April 6, 2018

By Gavin Lau

Table of Contents

1. Introduction

1.1 Purpose of the Document

The document will describe the process architecture of the scripting final project for this class. These process details include how the script will handle its processes, the result that they script will generate, and a detailed explanation of use-cases in all scenarios.

1.2 Scope of Development Project

This methodology report document provides an architectural overview of the script for identifying if the files are being successfully uploaded to the cloud storage solutions. The script will be developed in Python and will be able to run on both OSX and Windows systems.

The script will be able to run on both terminal or command prompt by the user. The user will be able to see a plain-text UI and check if the user's files are being successfully uploaded to the cloud.

1.3 Definitions, Acronyms, and Abbreviations

Table 1: Definitions

Keyword/Keyphrase	**Description**
Python	An interpreted high-level programming language for general-purpose programming.
Script	A computer program written in a **scripting** language.
Cloud Storage Solution	A service which allows users to store files on cloud storage and can access anywhere else with the internet.
Dropbox	Cloud storage solution company
OneDrive	Cloud storage solution owned by Microsoft
Google Drive	Cloud storage solution owned by Google
Box	Cloud storage solution company

Table 2: Acronyms

Acronym	**Description**
UI	User Interface: in the industrial design field of human–computer interaction, is the space where interactions between humans and machines occur.

1.4 References

Stackoverflow.com. (2018). Stack Overflow. [online] Available at: https://stackoverflow.com/ [Accessed 1 Apr. 2018].

2. Architectural Representation

2.1 Overview of Modules and Components

2.1.1 User Interface

The user interface provides an easy-to-use interface for the user to use the script. The user interface will provide user guides , which gives selections for the user to pick. Then the script will run based on the user's input.

2.1.2 Script Functionalities

The script will identify whether the targeted file is existed in the cloud drive. However, the user must install the cloud drive application into the computer because the script will search through the cloud drive based on the pathway. The script will check if the selected cloud drive is existed in the system, then allows the user to enter the targeted file name. Finally the script will run and search for the targeted file, and return the search result.

2.1.3 Cloud Drive Search

The cloud drive search is simple and effective because it is treated as one of the drives in the computer, which allows the script to run faster and also more effective.

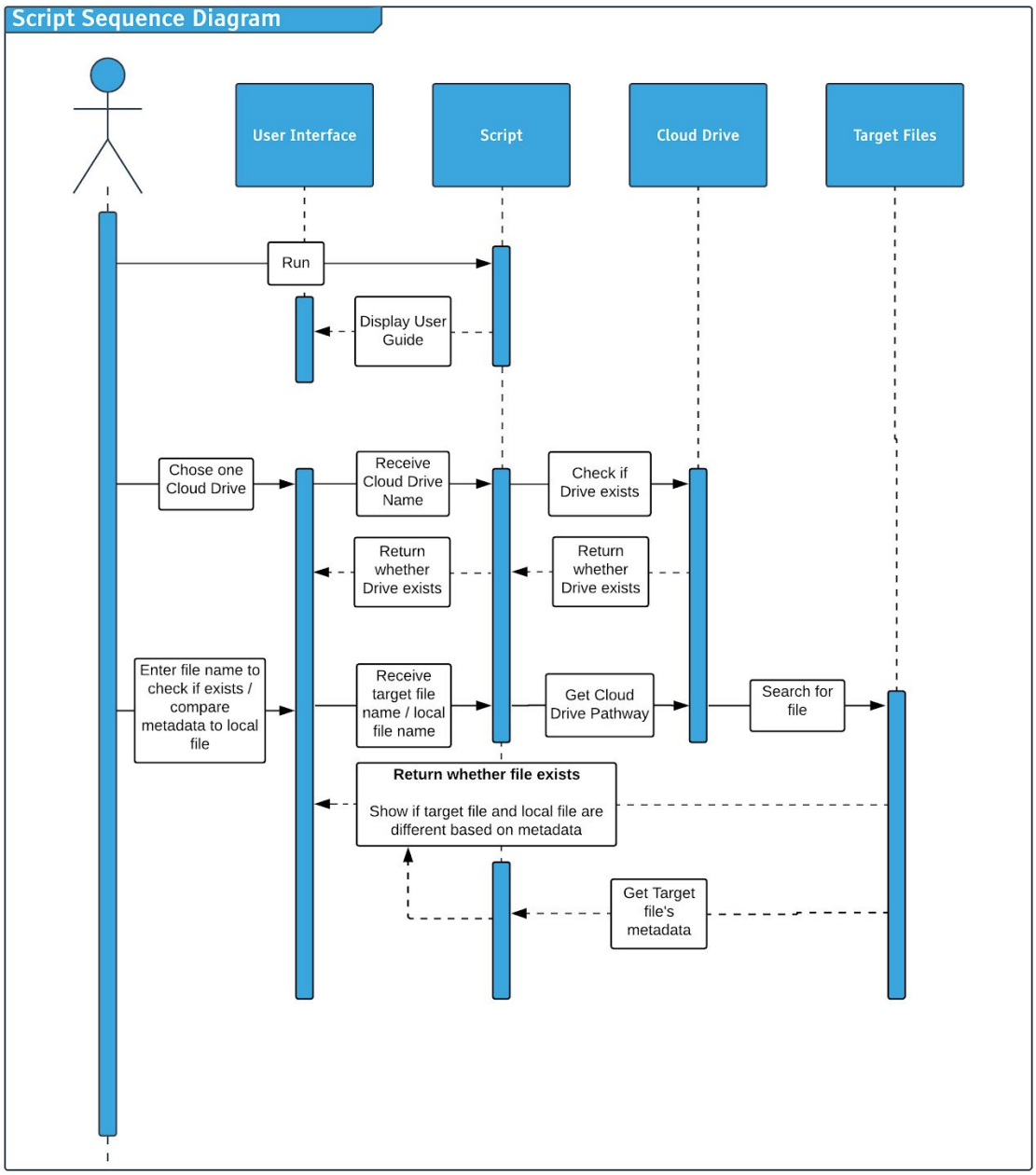
2.1.4 File Metadata Comparison

If the file is already existed in the drive and the user feels the file is either our dated or the file needs to be replaced by a local file, the script will allows the user to enter a pathway for the local file, then the script will search through the cloud drive and check if the two files are the same based on their metadata.

The comparison will based on the file's last modified time of the files and show the results to the user for the user's convince.

2.2 Script Sequence Diagram

Figure 1: Script Sequence Diagram



3. Use-Case View

3.1 Architecturally-Significant Use Cases

A description of the use-case view of the script architecture. The Use Case View is important input to the selection of the set of scenarios and/or use cases that are the focus of an iteration. It describes the set of scenarios and/or use cases that represent some significant, central functionality. It also describes the set of scenarios and/or use cases that have a substantial architectural coverage (that exercise many architectural elements) or that stress or illustrate a specific, delicate point of the architecture.

The script for identify if file exists in the cloud use cases are:
- Run Script: the user runs the script.

- Display UI: the user runs the script then the UI shows.

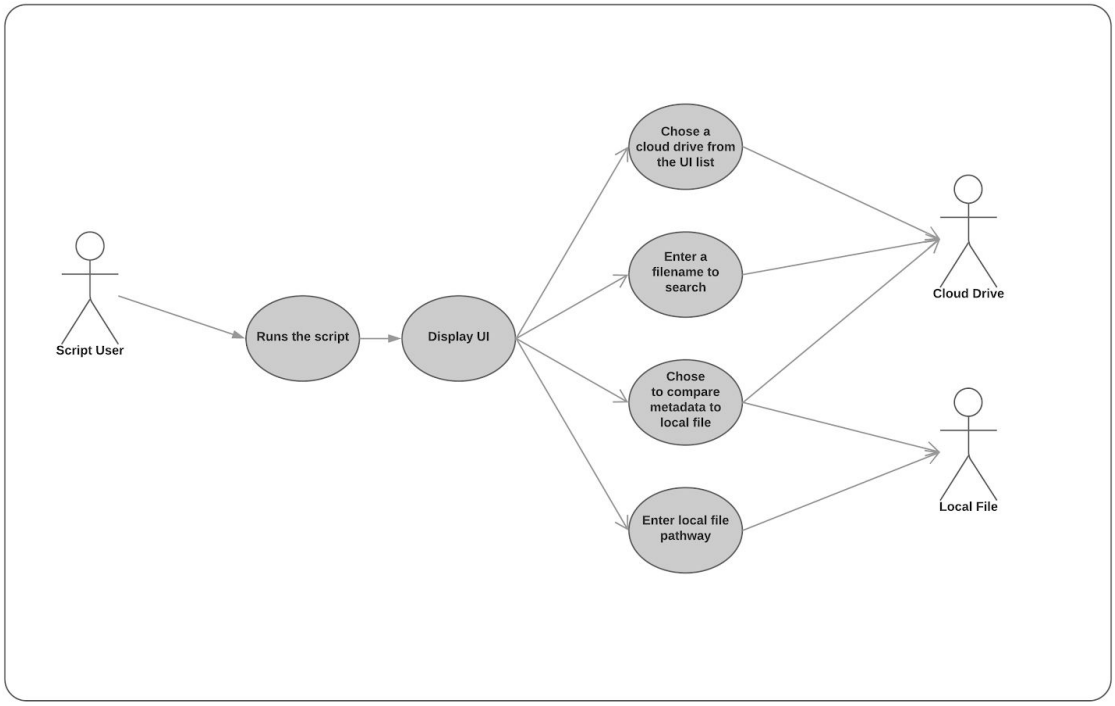
- Chose a cloud drive to search: the user chose the cloud storage solution from a list of cloud drive on the UI.

- Enter a filename to search: the user enter a filename to search if exists in the selected cloud drive.

- Enter a local filename to compare: if the user enter wishes to know if the same-name local file is the same as the one being uploaded to the cloud, the user can chose to compare the metadata to see which one is newer based on the “last modified date”.

These use cases are initiated by the user.

Figure 2: Use Case Diagram

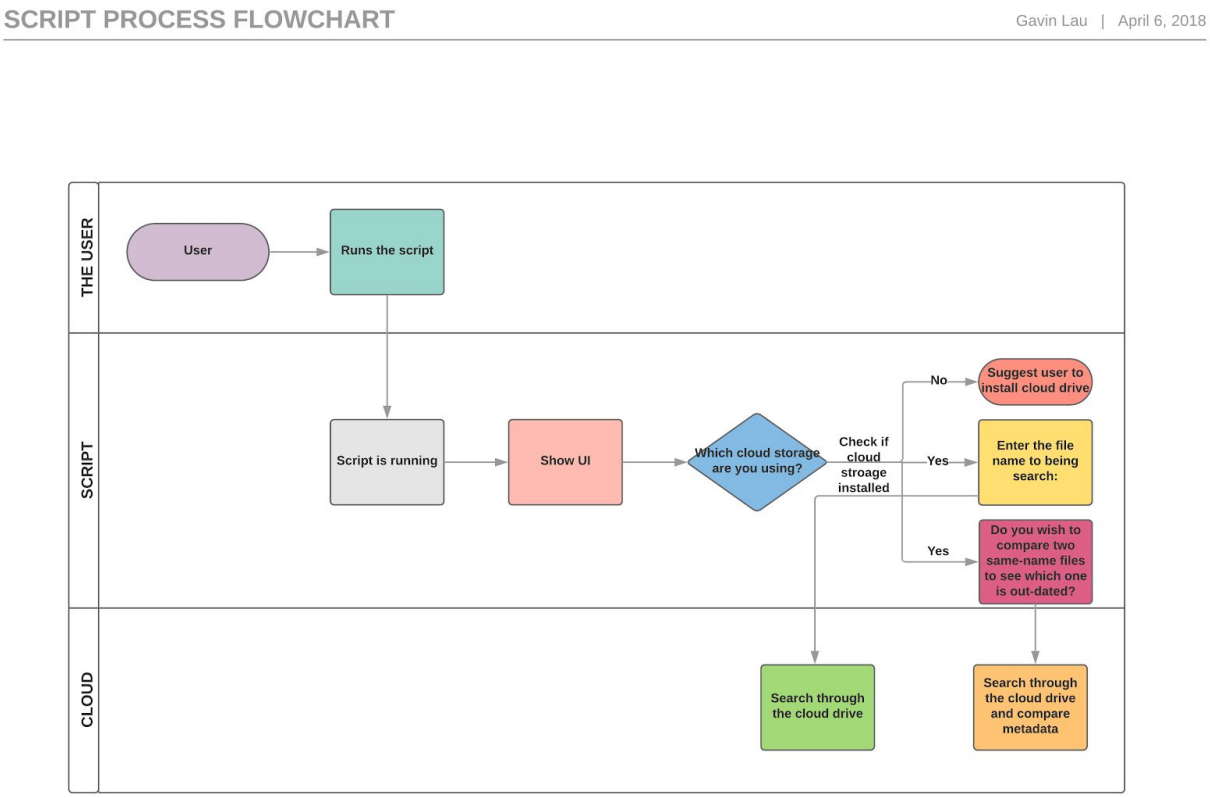


4. Process View

4.1 Process Flow Chart Diagram

As described above, the script will react based on the user's selection. Below is the process flowchart that shows what the script does based on the user's action.

Figure 3: Script process flowchart



5. Conclusion

In conclusion, the user can use the script to determine not only if the file is successfully uploaded to the specific cloud drive, but also determine which one is the newer one based on the “the last modify date”.