Lab 6: Automated App Installation

VideoLAN, a project and a non-profit organisation.

VLC media player

VLC is a free and open source cross-platform multimedia player and framework that plays most multimedia files as well as DVDs, Audio CDs, VCDs, and various streaming protocols.

Download VLC
Version 3.0.17.4 • Windows 64bit • 40 MB
28,437,556 downloads so far
NDR
iOS

COMP 593 - Scripting Applications

TABLE OF CONTENTS

LEARNING OBJECTIVES

Upon completion of this lab assignment, students should be able to:

- Use the Requests package to download a file from a URL
- Download a text file and extract portions of the text without saving the file to disk
- Use the hashlib module to compute the hash value of the body of an HTTP response message
- Verify that a file has been downloaded correctly by comparing its actual hash value against its expected hash value
- Save a downloaded file to disk
- Initiate silent execution of an installer application from a Python script
- Delete a file from disk

INTRODUCTION

For this lab assignment, students will write a Python script that automates downloading, verifying, and installing the VLC Media Player.

Specifically, the script will do the following:

- Get the expected SHA-256 hash value of the VLC installer from the VLC website
- Download the VLC installer from the VLC website
- Verify the integrity of the downloaded VLC installer by comparing the expected and computed SHA-256 hash values
- Save the downloaded VLC installer to disk
- Silently run the VLC installer
- Delete the VLC installer from disk

USING REQUESTS TO DOWNLOAD FILES

The steps for downloading a small file can generally be described as:

1. Send a GET request to the file URL.
2. Receive the response message.
3. Confirm that the response status code is 200.
4. Extract the content from the response message body.
5. Save the content as a file.

When downloading a large file, a more complicated process is required to avoid using an excessive amount of memory, but all the files you will be downloading for this lab are relatively small. If you are interested in learning about different ways to download a large file, check out this Stack Overflow thread.

DOWNLOADING A TEXT FILE

The following code will download and save a small text file to disk.

```
import requests

# Send GET message to download the file
file_url = 'https://raw.githubusercontent.com/JeremyDalby/SampleFiles/main/jokes.txt'
resp_msg = requests.get(file_url)

# Check whether the download was successful
if resp_msg.status_code == requests.codes.ok:

    # Extract text file content from response message
    file_content = resp_msg.text

    # Save the text file to disk
    with open(r'C:\temp\jokes.txt', 'w') as file:
        file.write(file_content)
```

Downloading a Binary File

The following code will download and save a small binary file (e.g., image, executable) to disk.

```python
import requests

# Send GET message to download the file
file_url = 'https://raw.githubusercontent.com/JeremyDalby/SampleFiles/main/dog.jpg'
resp_msg = requests.get(file_url)

# Check whether the download was successful
if resp_msg.status_code == requests.codes.ok:

    # Extract binary file content from response message
    file_content = resp_msg.content

    # Save the binary file to disk
    with open(r'C:\temp\dog.jpg', 'wb') as file:
        file.write(file_content)
```

Working with Response Message Body

It is not always necessary to save a downloaded file to disk. For example, if your script only needs specific information from a downloaded text file, there is no need to save the entire file to disk – the information can be extracted directly from the response message body.

Extracting Text from a Response Message Body

The following code downloads a text file that contains several jokes (each of which are separated by a line of 25 consecutive tildes), splits the text file content into a list of jokes, and prints the 5th joke in the list, without saving the text file to disk.

```python
import requests

# Send GET message to download the file
file_url = 'https://raw.githubusercontent.com/JeremyDalby/SampleFiles/main/jokes.txt'
resp_msg = requests.get(file_url)

# Check whether the download was successful
if resp_msg.status_code == requests.codes.ok:

    # Extract text file content from response message body
    file_content = resp_msg.text

    # Split the text file content into a list of jokes
    jokes = file_content.split('~~~~~~~~~~~~~~~~~~~~~~~~~\n')

    # Print the 5th joke in the list
    print(jokes[4])
```

Computing the Hash Value of a Response Message Body

The following code downloads an image file and computes its SHA-256 hash value without saving the image file to disk.

```python
import requests
import hashlib

# Send GET message to download the file
file_url = 'https://raw.githubusercontent.com/JeremyDalby/SampleFiles/main/dog.jpg'
resp_msg = requests.get(file_url)

# Check whether the download was successful
if resp_msg.status_code == requests.codes.ok:

    # Extract binary file content from response message body
    file_content = resp_msg.content

    # Calculate SHA-256 hash value
    image_hash = hashlib.sha256(file_content).hexdigest()

    # Print the hash value
    print(image_hash)
```

Running an Executable File

A Python script can initiate execution of a program and wait for the program to complete using the **run()** function defined in the subprocess module. The subprocess module is included in the Python Standard Library, so it should already be installed on your computer.

Running the VLC Installer

The following code will run the VLC Media Player installer, wait for the installation to complete, and then delete the installer file. The additional parameters passed into the **subprocess.run()** function indicate that the English language version of VLC installation should be done silently, as described in the VLC installation documentation. Silent installation is desirable because it does not require clicking any buttons on the installation dialogs (e.g., Next, Next, Finish, etc.), so the installation can run unattended.

```python
import subprocess
import os

installer_path = r'C:\temp\vlc-3.0.17.4-win64.exe'
subprocess.run([installer_path, '/L=1033', '/S'])
os.remove(installer_path)
```

STUDENT INSTRUCTIONS

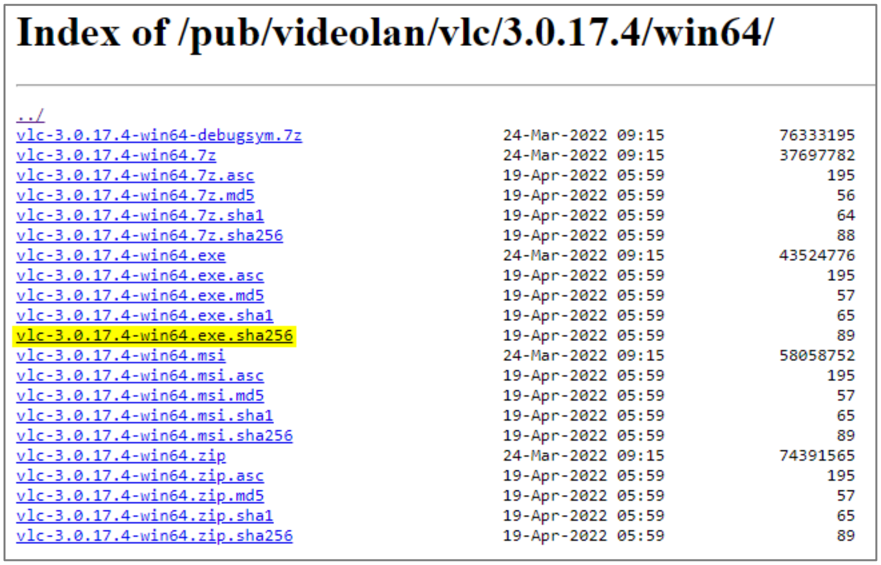
For this lab, students will implement a Python script that automates downloading, verifying, and installing the VLC Media Player on a Windows PC. The steps that the script must perform are described in the following subsections.

Students should reference the example code presented in the preceding sections when implementing the script, i.e., almost all of what the script needs to do can be found in the example code above.

Various types of VLC installers are available for Windows, MacOS, and Linux. You script should download, verify, and install the **.exe** installer that is appropriate for your computer for the latest version of VLC Media Player. At the time of writing these lab instructions, the latest version is 3.0.17.4.

STEP 1: GET THE EXPECTED HASH VALUE OF THE VLC INSTALLER

The VLC installer archive website provides text files that contain the expected hash values of each VLC installer version. For example, the expected SHA-256 hash value of the 64-bit Windows **.exe** installer for VLC version 3.0.17.4 is located in the text file highlighted below, which can be found here: http://download.videolan.org/pub/videolan/vlc/3.0.17.4/win64/



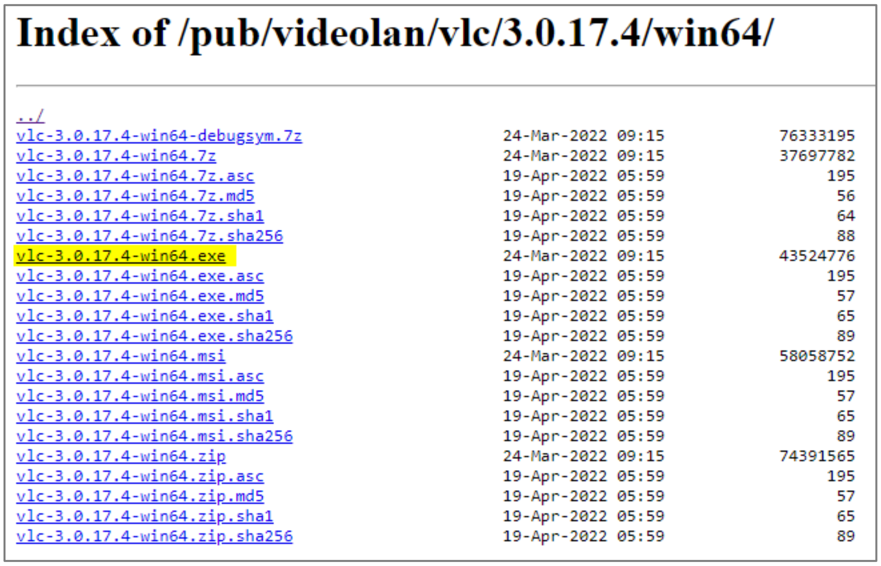
Index of /pub/videolan/vlc/3.0.17.4/win64/

```
../
vlc-3.0.17.4-win64-debugsym.7z                  24-Mar-2022 09:15          76333195
vlc-3.0.17.4-win64.7z                           24-Mar-2022 09:15          37697782
vlc-3.0.17.4-win64.7z.asc                       19-Apr-2022 05:59               195
vlc-3.0.17.4-win64.7z.md5                       19-Apr-2022 05:59                56
vlc-3.0.17.4-win64.7z.sha1                      19-Apr-2022 05:59                64
vlc-3.0.17.4-win64.7z.sha256                    19-Apr-2022 05:59                88
vlc-3.0.17.4-win64.exe                          24-Mar-2022 09:15          43524776
vlc-3.0.17.4-win64.exe.asc                      19-Apr-2022 05:59               195
vlc-3.0.17.4-win64.exe.md5                      19-Apr-2022 05:59                57
vlc-3.0.17.4-win64.exe.sha1                     19-Apr-2022 05:59                65
vlc-3.0.17.4-win64.exe.sha256                   19-Apr-2022 05:59                89
vlc-3.0.17.4-win64.msi                          24-Mar-2022 09:15          58058752
vlc-3.0.17.4-win64.msi.asc                      19-Apr-2022 05:59               195
vlc-3.0.17.4-win64.msi.md5                      19-Apr-2022 05:59                57
vlc-3.0.17.4-win64.msi.sha1                     19-Apr-2022 05:59                65
vlc-3.0.17.4-win64.msi.sha256                   19-Apr-2022 05:59                89
vlc-3.0.17.4-win64.zip                          24-Mar-2022 09:15          74391565
vlc-3.0.17.4-win64.zip.asc                      19-Apr-2022 05:59               195
vlc-3.0.17.4-win64.zip.md5                      19-Apr-2022 05:59                57
vlc-3.0.17.4-win64.zip.sha1                     19-Apr-2022 05:59                65
vlc-3.0.17.4-win64.zip.sha256                   19-Apr-2022 05:59                89
```

Your script must download that text file and extract the SHA-256 hash value from it. There is no reason for the script to save the text file to disk since the hash value can be extracted directly from the response message body.

Step 2: Download the VLC Installer

Various versions of the VLC installer are available for download from the VLC installer archive website. For example, the 64-bit Windows **.exe** installer for VLC version 3.0.17.4 can be downloaded from here: http://download.videolan.org/pub/videolan/vlc/3.0.17.4/win64/



Index of /pub/videolan/vlc/3.0.17.4/win64/

```
../
vlc-3.0.17.4-win64-debugsym.7z                     24-Mar-2022 09:15            76333195
vlc-3.0.17.4-win64.7z                              24-Mar-2022 09:15            37697782
vlc-3.0.17.4-win64.7z.asc                          19-Apr-2022 05:59                 195
vlc-3.0.17.4-win64.7z.md5                          19-Apr-2022 05:59                  56
vlc-3.0.17.4-win64.7z.sha1                         19-Apr-2022 05:59                  64
vlc-3.0.17.4-win64.7z.sha256                       19-Apr-2022 05:59                  88
vlc-3.0.17.4-win64.exe                             24-Mar-2022 09:15            43524776
vlc-3.0.17.4-win64.exe.asc                         19-Apr-2022 05:59                 195
vlc-3.0.17.4-win64.exe.md5                         19-Apr-2022 05:59                  57
vlc-3.0.17.4-win64.exe.sha1                        19-Apr-2022 05:59                  65
vlc-3.0.17.4-win64.exe.sha256                      19-Apr-2022 05:59                  89
vlc-3.0.17.4-win64.msi                             24-Mar-2022 09:15            58058752
vlc-3.0.17.4-win64.msi.asc                         19-Apr-2022 05:59                 195
vlc-3.0.17.4-win64.msi.md5                         19-Apr-2022 05:59                  57
vlc-3.0.17.4-win64.msi.sha1                        19-Apr-2022 05:59                  65
vlc-3.0.17.4-win64.msi.sha256                      19-Apr-2022 05:59                  89
vlc-3.0.17.4-win64.zip                             24-Mar-2022 09:15            74391565
vlc-3.0.17.4-win64.zip.asc                         19-Apr-2022 05:59                 195
vlc-3.0.17.4-win64.zip.md5                         19-Apr-2022 05:59                  57
vlc-3.0.17.4-win64.zip.sha1                        19-Apr-2022 05:59                  65
vlc-3.0.17.4-win64.zip.sha256                      19-Apr-2022 05:59                  89
```

Your script must download the**.exe** installer for the latest version of VLC that is appropriate for your computer. The installer file should not be saved to disk until after its integrity has been verified by comparing its SHA-256 hash value to the expected SHA-256 hash value.

Step 3: Verify the Integrity of the Downloaded VLC Installer

Your script must calculate the SHA-256 hash value of the downloaded installer file and compare it against the expected SHA-256 hash values. If both hash values match (as they probably will), then the integrity of the installer file has been verified. If not, the script should not run the installer as it may be infected with some malware.

Hint: Check out the example code in the Lecture 6 slides -- the slide where Guido is looking really cool in his Python shirt. 😜

Step 4: Save the Downloaded VLC Installer to Disk

If the integrity of the downloaded VLC installer file is verified, your script must save the installer file to disk. Where the script saves it is up to you. One decent option is to save it in the operating system's temp folder, the path of which can be retrieved using **os.getenv('TEMP')**, since this folder should exist on every computer.

Step 5: Silently Run the VLC Installer

Once the VLC installer file is downloaded, verified, and saved to disk, the script must run it, since installing VLC is the entire purpose of the script. There *might* be some code in the above examples that shows exactly how to do this.

Step 6: Delete the VLC Installer from Disk

After VLC has been installed, there is no need to keep the installer file – it's just using up valuable disk space – so your script must clean up its tracks by deleting the file. It's *possible* the code needed to do this is in the above examples.

Script Structure

Script functionality must be divided up into functions that are called from the **main()** function. The exact functional breakdown is up to you to decide. The code below shows one potential design. You may use it or come up with your own design.

```
def main():

    # Get the expected SHA-256 hash value of the VLC installer
    expected_sha256 = get_expected_sha256()

    # Download (but don't save) the VLC installer from the VLC website
    installer_data = download_installer()

    # Verify the integrity of the downloaded VLC installer by comparing the
    # expected and computed SHA-256 hash values
    if installer_ok(installer_data, expected_sha256):

        # Save the downloaded VLC installer to disk
        installer_path = save_installer(installer_data)

        # Silently run the VLC installer
        run_installer(installer_path)

        # Delete the VLC installer from disk
        delete_installer(installer_path)
```

Dropbox Submission

Submit the URL of the GitHub repository that contains your script, e.g.,
https://github.com/BobLoblaw/COMP593-Lab6

ASSESSMENT

Item	Out Of	Assessment Criteria
GitHub	1	• GitHub repository is public • Repository contains script file • Repository does not contain any unnecessary files
Step 1	2	• Text file downloaded from VLC website • Expected SHA-256 hash value extracted from text file
Step 2	2	• VLC installer downloaded from website, but not saved to disk
Step 3	2	• SHA-256 hash value of downloaded installer computed • Hash values compared for equality • Installer not run if hash values are different
Step 4	1	• VLC installer file saved to disk
Step 5	1	• VLC installer run silently
Step 6	1	• VLC installer file deleted
Total:	**10**	