**Python C2 Student 601**

**Version**

**Version:** 0.3.1

**Description**

The C2 (Command and Control) application is a robust client-server system designed to facilitate secure data extraction from potentially compromised devices. Leveraging TLS (Transport Layer Security) for encrypted communications, this application ensures that all data transmitted between the client and server remains confidential and protected from eavesdropping.

The client component of the application operates on the target device, enabling the extraction of data while maintaining minimal footprint to avoid detection. The server component is responsible for receiving and processing the data collected from the client.

**Important Note:** This application is intended solely for use with the express permission of the device owners. Unauthorized use of this application may violate laws and ethical standards. Users are advised to ensure compliance with all relevant legal and ethical guidelines before deployment

**Table of Contents**

* [Installation](#installation)
* [Usage](#usage)
* [Modules Used](#modules-used)
* [Functions Implemented](#functions-implemented)
* [Functions to be Implemented](#functions-to-be-implemented)
* [Contributing](#contributing)
* [License](#license)

**Installation**

1. Clone the repository:

bash

Copy code

git clone ###GITEA LINK HERE###

1. Navigate to the project directory:

bash

Copy code

cd project-name

1. Install the required packages:

Copy code

pip install -r requirements.txt

**Usage**

Users will be operating from a list of pre-specified commands that will be securely transmitted across a network or the internet to execute on a remote machine. The majority of computational requirements are at the client side of the C2 application. Further explanation can be found later in this readme.

All features, both those currently implemented and those intended for future development are functional on the following operating systems and python versions:

* Windows 10 Enterprise with Python 3.11.9
* Kali Linux with Python 3.11.4

**Modules Used**

* **Server**: The control node of the C2 application. Commands are input here and responses will be received by the server in a JSON format which is then decoded into python objects and displayed for use. All actions on the server are logged and responses can be found in the server logs if required.
* Client: Receives the commands from the server and executes the pre-defined functions on request. It will return the responses in a JSON format to the server.
* Protocol: Controls the parameters of the send and receive functions between server and client. Operates with a 4bit header being prepended to the data to specify the size of the data and ensure only that amount is being sent in order to prevent arbitrary or unintended data transmission.
* Lookups: Contains a dictionary that acts as a middle man between the client and the functions/commands it is required to run.
* Services: Contains the functions that will be called by the client.

**Functions Implemented**

* **Class - Server**: Contains the following:
  + **Run Method :** Socket establishment and TLS wrapping. This also gives a “welcome message” and will present the user with the list of useable commands.
  + **Handle\_User\_Connection Method:** Handle input from the user and control sending messages and receiving messages. This includes the logic required to decode the JSON format.
  + **Remove\_Connection Method:** This removes a connection from the class instance and closes the connection to it from the server side.
  + There is also some future development code currently commented out. This is not yet implemented and may cause errors if it is run.
* **Class - Client**: Description of what the function does, its parameters, and return values.
  + **Run Method :** Socket establishment and TLS wrapping. This will connect to a server with the IP and Port given in the Class call (found at the bottom of the code.
  + **Handle\_Messages Method:** Handle messages received by the client and resolve them into the function the user requested. This function splits a string into a list and then checks its length. It will then take the second element of the list (index 1) and use that as a command argument, for example the “-la” in “ls -la”. The first element (index 0) will validate against the cmddict found in lookups.py along with the OS to select the correct function to call. This will then be executed.
  + Terminate\_client Method: This method will close the client connection on the client side and then force an exit using the ‘sys’ module.
  + There is also some future development code currently commented out. This is not yet implemented and may cause errors if it is run.

**Functions to be Implemented**

* **File Send/Receive**: A set of functions that will enable file transfers between client and server. The intent is to use the binary bytes method to read, transmit, write and save files on either side of the connection.
* **Binary/DLL hashing**: In order to confirm the integrity of the binarys/DLL’s being operated on the clients machine, the ability to run a hashing algorithm on them and then compare to known good versions will be implemented.
* **Cat/Read files**: A function enabling the user to read a file to stdout without the need to file transfer them will be implemented to enable flexibility to the user. This means a user will be able to read a file before transfer and decide if it is required.
* **NETSTAT on Linux only**. This function is already enabled for use on windows OS. This method has a working function but requires amendment to the format of the response to be viable with the JSON transfer method.d

**License**

**GNU General Public License (GPL) Version 3 - Summary**

This is a summary of the GNU General Public License (GPL) version 3, which allows you to:

* **Use** the software for any purpose.
* **Study** how the program works, and change it to make it do what you wish (access to the source code is a precondition).
* **Distribute copies** of the original program so you can help others.
* **Distribute modified versions** of the program, so you can give the community a chance to benefit from your changes. Access to the source code is a precondition for this.

However, you must keep the GPL license intact and make sure that any distributed versions (modified or unmodified) remain free software.

**Important Points:**

* You cannot impose additional restrictions on the rights granted by this License.
* There is no warranty for the software, and the authors are not liable for any damages.

For full details, refer to the complete license text at GNU.org.