Shane Lewis – Keylogger Project

Graphical user interface, text, application

Description automatically generatedThe purpose of this project is to understand how a keylogger works using python and walking through the script of what it does, and its impact on the person at the receiving end. A keylogger is what its name dictates; It records the keys and activities that users input and can capture anything the user types, especially and most prized, sensitive information such as logins, credit card numbers, and plenty more. Finally, I will show what can be done to prevent against keyloggers and be significantly safer and more assured that your information isn’t being captured.

This is the python script I got that is among different variations that can be created for a keylogger.

The way this python script works is this:

* We import the keyboard module from the pynput package, as this will allow us to create a listener for events occurring on the keyboard.
* The next block, ‘def on\_press(key)’ to the end of its block, defines our function ‘on\_press’ that is called when a key is pressed. The function checks whether the key being pressed is a special key such as ‘shift’ or ‘ctrl’, or if it is a character key like ‘a’ or ‘b’. If it is a character key, it prints character to console. And if special key, it prints message indicating the key that was pressed.
* The next block, ‘def on\_release(key)’ to its end block, defines function ‘on\_release’ that’s called when key is released. It prints a message indicating the key that was released. For our purpose, we included the ‘esc’ key that will start the capture and with the return false, pressing the ‘esc’ again will stop the listener.
* The last block, ‘with keyboard…’ is where we create the listener to listen for the events. This ‘Listener’ class is provided from the ‘pynput.keyboard’ module which takes the two arguments above: ‘on\_press’ and ‘on\_release’ when a key is pressed and released. Since our listener starts with the ‘with’ statement, it effectively ensures we are not using more resources than need be, only when it is being used. The ‘join()’ method that is called with the listener object is to start the event loop, which waits for keyboard inputs, and when it’s inputs are made by the user, the appropriate function that is defined is called (i.e. ‘on\_press’ or ‘on\_release’).

To try this yourself, what you need is to have Python installed. You can do this by going to the official Python website and finding the correct version of your system and the latest release. You also are going to need to turn off any anti-virus or anti-malware software as this will flag it as a Potentially Un-wanted Program (PUP), at least for me it does. You will also have to adjust the code to fit your specifications.

Graphical user interface, text, application, email

Description automatically generatedI updated and added a few additional features to see what is being recorded and so on. The revised code includes adding a log file, you can name it what you choose, to store the received inputs to, and append them to the file. This is in order to test it yourself and see its output.

There are plenty of other variations and additions that can be made to creating a keylogger to track more specific movements like mouse cursor and click count or mouse position and such, but this is a simple example of what can be used to effectively gather sensitive information and use it for unethical purposes.

After trying out the code, there appeared to be errors with undefined names and so perusing through and attempting to validate the error, it was solved. What I found was in some cases, it depends on the script or library used, and since what I was using was external, it was showing to be giving the errors.

I further looked through the code to see what could be improved upon for future errors or problems, and with utilizing ChatGPT, it pointed out that:

* Encoding issues can stem from it by not using a proper encoding format, and suggested ‘utf-8’
* Error handling for special keys: This assisted with suggesting having additional exceptions for if the file couldn’t be opened for writing to handle them
* Possibility of infinite loop and its suggestion was having a mechanism to stop the loop, in which case I already have the ‘esc’ key as the exit

Graphical user interface, text, application, email

Description automatically generatedI went on to address the issues and perform the fixes suggested with adding exception handling and the encoding to prevent future errors that could occur and stop the script. This completed version of the keylogger with extra and additional changes from the start shows improvement and fixes to basic code and to assist in its function to handle unexpected issues should they arise. In each person’s case, this code will have to be tailored to their own systems, versions, if need be, their specific needs and more. Again, more can be added, but this is a demonstration of a keylogger’s potential. Below is the results and final changes with what was needed to work more efficient and properly.

To protect yourself against keylogger attempts, it is best to keep a good rule of thumb on a number of things about your activity on the internet. Don’t download files or accept files from unknown sources. There could be trojan’s hidden within to download further malicious software and corrupt and infect your whole system. Use anti-virus/malware software to protect your system while browsing or downloading. It is not a solve-all solution, but it greatly makes a difference having it. In some cases you can combine different anti-viruses to provide layered defense. Add on top having a firewall so malicious activity won’t infect the entire network and will be blocked is a good practice to instill. The single biggest thing ordinary users can do to keep themselves safe, not just with keyloggers, but with other malware and such, is keeping your OS, your AV, Firewalls, and applications updated frequently. Developers are inherently bad at using secure coding practices and you will see frequent patches made to correct bugs and improve security over time. Their prime purpose is getting the product/code out there and improving on it later. So, keep everything updated, use safe internet habits and avoid untrusted downloads.