Python Backdoor and Controller

Written and developed by Stone Preheim

**Purpose:**

**This document will define and list the functionality to be included within a backdoor and its controller both written in Python. This backdoor should be as invisible to the victim system as possible while also providing the attacker with powerful control over the victim system and its peripherals. The project will include a controller that will act as a server accepting encrypted TCP connections. Once a connection is established the controller should display a shell for the attacker to input commands into. The commands within the system should allow for file system transversal, manipulation, transfer, and other functions such as keylogging. A distribution system will be created for the backdoor as well, ideally packaged into a self-contained EXE. This project will be strictly for research and educational purposes only and will not be exposed to public spaces where it could be used for nefarious purposes.**

**Functional Specifications:**

**Green = implementation completed**

**Yellow = Partially completed**

1. **The backdoor will**
   1. **Be able to be distributed and run easily**
   2. **Hide its execution from the user**
   3. **Provide access to system resources**
      1. **Allow for file system manipulation**
      2. **Allow for transversal of file system**
      3. **Include a keylogger function**
      4. **Allow files and directories to be compressed and sent to the controller**
      5. **Allow a screenshot to be taken and uploaded to the controller**
      6. **Allow for peripheral control if possible**
   4. **Identify itself uniquely to the controller**
   5. **Encrypt its network traffic sent to the controller**
   6. **Allow controller to manually input keystrokes**
2. **The controller will**
   1. **Accept connections from backdoors on victim systems**
   2. **Provide a way to select which system to control**
      1. **(Time permitting) allow for mass botnet control**
   3. **Show a shell for inputting commands**
   4. **Send commands to selected victim system**
   5. **Receive file/video uploaded from victim systems**
   6. **Allow user to shut down backdoors remotely**
   7. **Provide a command to kill the entire network of connected systems**
   8. **Need to be hosted on a system with a static IP and port forwarded for WAN access**
3. **Documentation will be written**
   1. **For installation of both controller and backdoor**
   2. **Command list**
   3. **General documentation of included code**

**Proposed Project Timeline:**

**First Checkpoint: Functionality documentation, setting up port forwarding, and creating initial files with socket tests**

**Second Checkpoint: MVP reached, shell built with system transversal, basic commands implemented**

**Final Checkpoint: Advanced commands finished, and scripts packaged for distribution**