# 🛡️ Securing Python Modules Using Nuitka

## 📌 Objective

Protect your Python source code from reverse engineering while still being able to import it as a module in other Python scripts.

## 🔧 Installation

### 1. Install Nuitka

pip install nuitka

### 2. Install a C/C++ Compiler

On Windows:

- Install Visual Studio Build Tools: https://visualstudio.microsoft.com/visual-cpp-build-tools/  
- Select: C++ build tools and Windows 10 SDK

On Linux:

sudo apt install build-essential

## 🔐 Secure Compilation for Import

Convert `.py` to `.pyd` (Windows) or `.so` (Linux/macOS):

nuitka --module your\_module.py

This will generate:

- your\_module.pyd on Windows

- your\_module.so on Linux/macOS

## 📥 How to Import into Another Script

import your\_module  
  
your\_module.your\_function()

## 🔐 Advanced Protection Options

Enhanced compilation with more security:

nuitka --module your\_module.py \  
 --lto \  
 --plugin-enable=anti-bloat \  
 --plugin-enable=pylint-warnings

|  |  |
| --- | --- |
| Option | Purpose |
| --lto | Link Time Optimization (faster and more optimized binary) |
| --plugin-enable=anti-bloat | Removes unnecessary standard library parts |
| --plugin-enable=pylint-warnings | Strips docstrings and optimizes logic |

## 🚫 What Not to Use If You Need Import

* Avoid the following if you want to import the module because it makes exe:

nuitka --onefile your\_module.py

nuitka --standalone your\_module.py

## 🔍 Reverse Engineering Resistance

|  |  |  |
| --- | --- | --- |
| File Type | Reverse Engineering Risk | Importable? |
| .py | ❌ Very Easy | ✅ Yes |
| .pyc | ❌ Easy (with tools) | ✅ Yes |
| .exe (Nuitka) | ⚠️ Moderate-Hard | ❌ No |
| .pyd/.so (Nuitka) | ✅ Very Hard | ✅ Yes |

## ✅ Summary

- Use `nuitka --module your\_script.py` to build a secure, importable module.

- For higher protection, add flags like `--lto`, `--plugin-enable=anti-bloat`, etc.

- Never distribute raw `.py` or `.pyc` files if protection is your goal.

## 📁 Example Folder Structure

project/  
│  
├── your\_module.py # Your original code (optional: can delete after build)  
├── your\_module.pyd # Output file (protected and importable)  
└── another\_script.py # Where you import your\_module