# C2 Framework: Execution Guide

## Step-by-Step: What to Run

### 1. Start the C2 Server (Controller)

You have two options depending on whether you want socket-based or web-based control.

* Option A: Socket-Based C2

Run the following command:

python C2\_Server/c2\_server.py

This is the main server that listens for agent connections and sends tasks.

* Option B: Web-Based Interface

Run the following command:

python C2\_Server/web\_interface.py

This provides a Flask-based dashboard with REST API to manage agents and tasks.

### 2. Start the Shellcode Generator (Optional)

Run the following command:

python C2\_Server/shellcode\_interface.py

Use this to generate reverse shells or encoded payloads through a browser interface.

### 3. Run the Agent on Victim Machine

Modify the IP/Port in agent/agent.py to match your C2 server, then run:

python agent/agent.py

This will initiate the connection back to your server and await commands.

## What the Other Files Do

- Files under modules/: These are imported and triggered by the agent upon receiving a specific task (e.g., keylogger, screenshot).

- priv\_esc.py, process\_injection.py, etc. are not standalone—they’re invoked by the C2 command logic.

- Files in docs/ are for documentation.

- shellcode.html is used as a front-end template by shellcode\_interface.py.

## Execution Summary

|  |  |  |
| --- | --- | --- |
| File | Purpose | Run Directly? |
| C2\_Server/c2\_server.py | Socket-based server | Yes |
| C2\_Server/web\_interface.py | Flask-based server UI | Yes |
| C2\_Server/shellcode\_interface.py | Shellcode generator UI | Yes |
| agent/agent.py | Victim payload | Yes |
| modules/\*.py | Used internally by the agent | No |
| docs/\*.md | Documentation | No |