# LAB 4: Agent Communication Using FIPA-ACL

DCIT 403  
11125009  
Date: February 19, 2026

## Objective

To enable inter-agent communication using the Foundation for Intelligent Physical Agents – Agent Communication Language (FIPA-ACL). This lab implements ACL message exchange between multiple SPADE agents using INFORM and REQUEST performatives, with incoming messages parsed to trigger agent actions.

## Scenario Description

A collective of autonomous hacker agents coordinate cyber operations using secure FIPA-ACL messaging to discover vulnerabilities, execute exploits, and monitor security threats.

The system consists of three agents:

|  |  |  |
| --- | --- | --- |
| Agent | Role | Performatives Used |
| ReconAgent | Scans networks and discovers vulnerabilities | INFORM (sends intel to MainHacker) |
| MainHackerAgent | Coordinates operations, decides on exploits | REQUEST (asks Watchdog for heat check) |
| WatchdogAgent | Monitors security heat levels | INFORM (responds with heat status) |

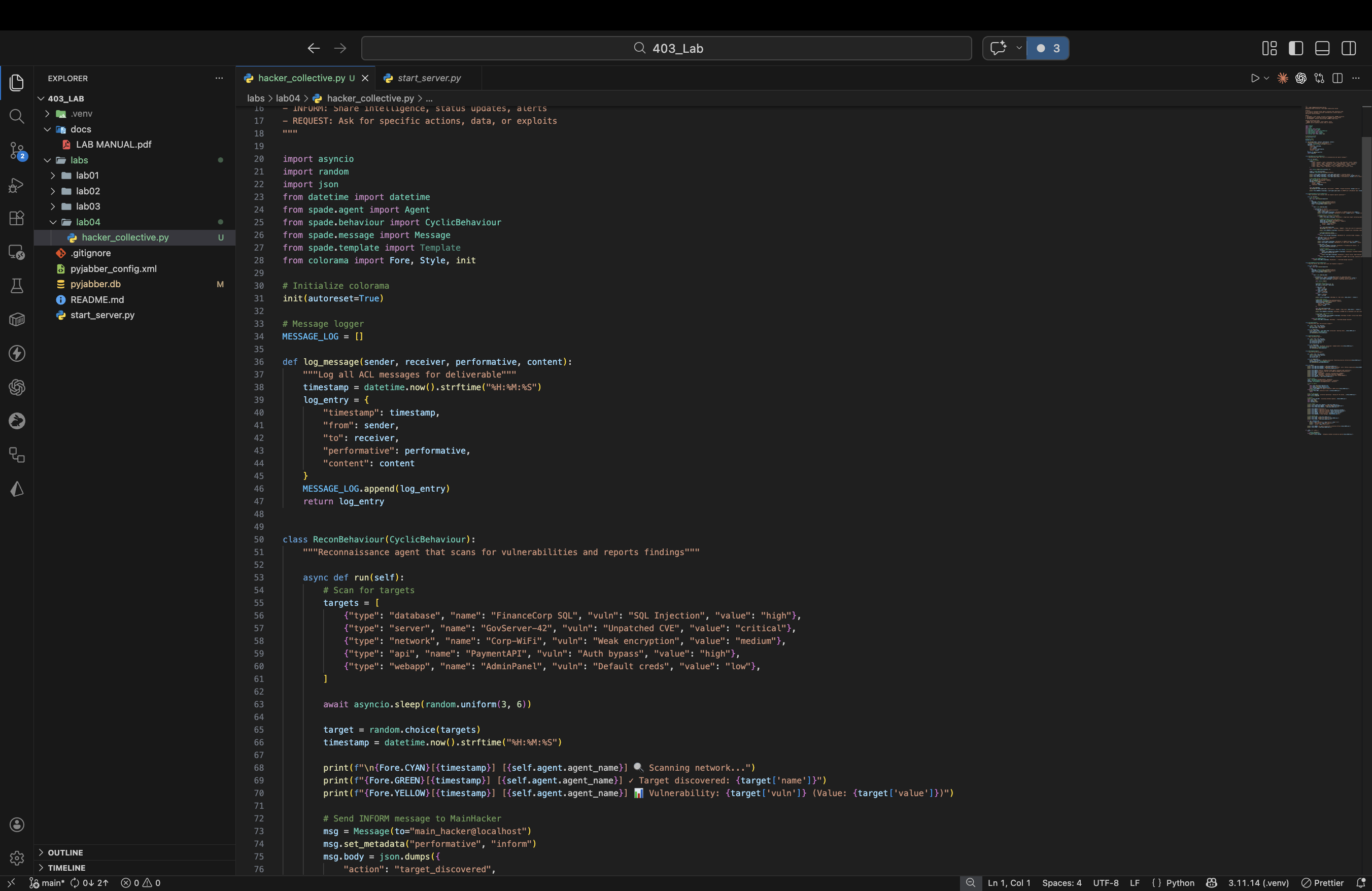
## Communication Protocol

The agents communicate using FIPA-ACL performatives embedded as XMPP message metadata:

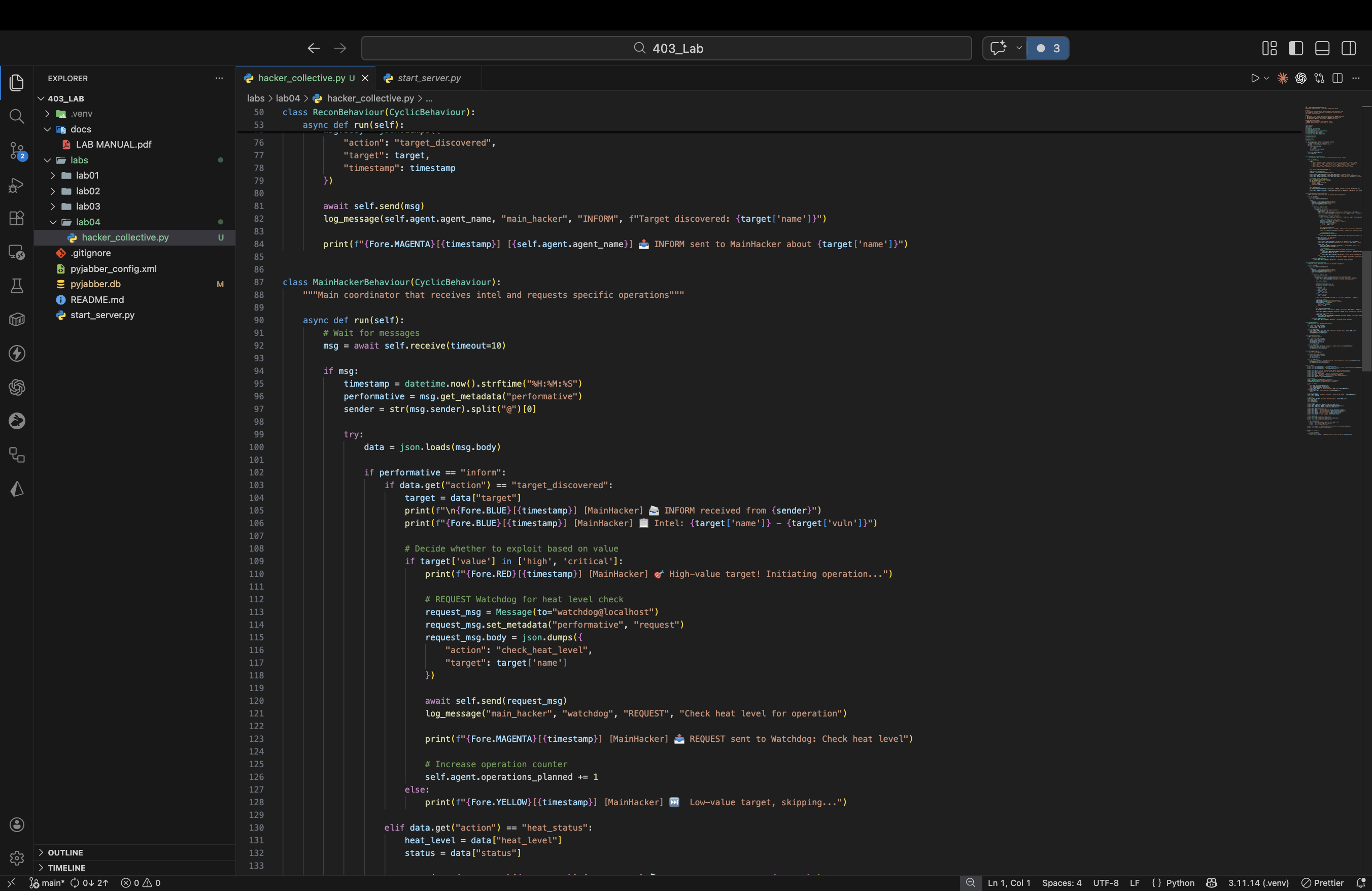
COMMUNICATION FLOW:  
─────────────────────────────────────────────────  
1. ReconAgent ──INFORM──> MainHacker  
 (target discovered with vulnerability details)  
  
2. MainHacker ──REQUEST──> Watchdog  
 (check heat level before exploit)  
  
3. Watchdog ──INFORM──> MainHacker  
 (heat level status: SAFE / ELEVATED / CRITICAL)  
  
4. MainHacker executes or aborts based on heat level  
─────────────────────────────────────────────────

## Source Code Implementation

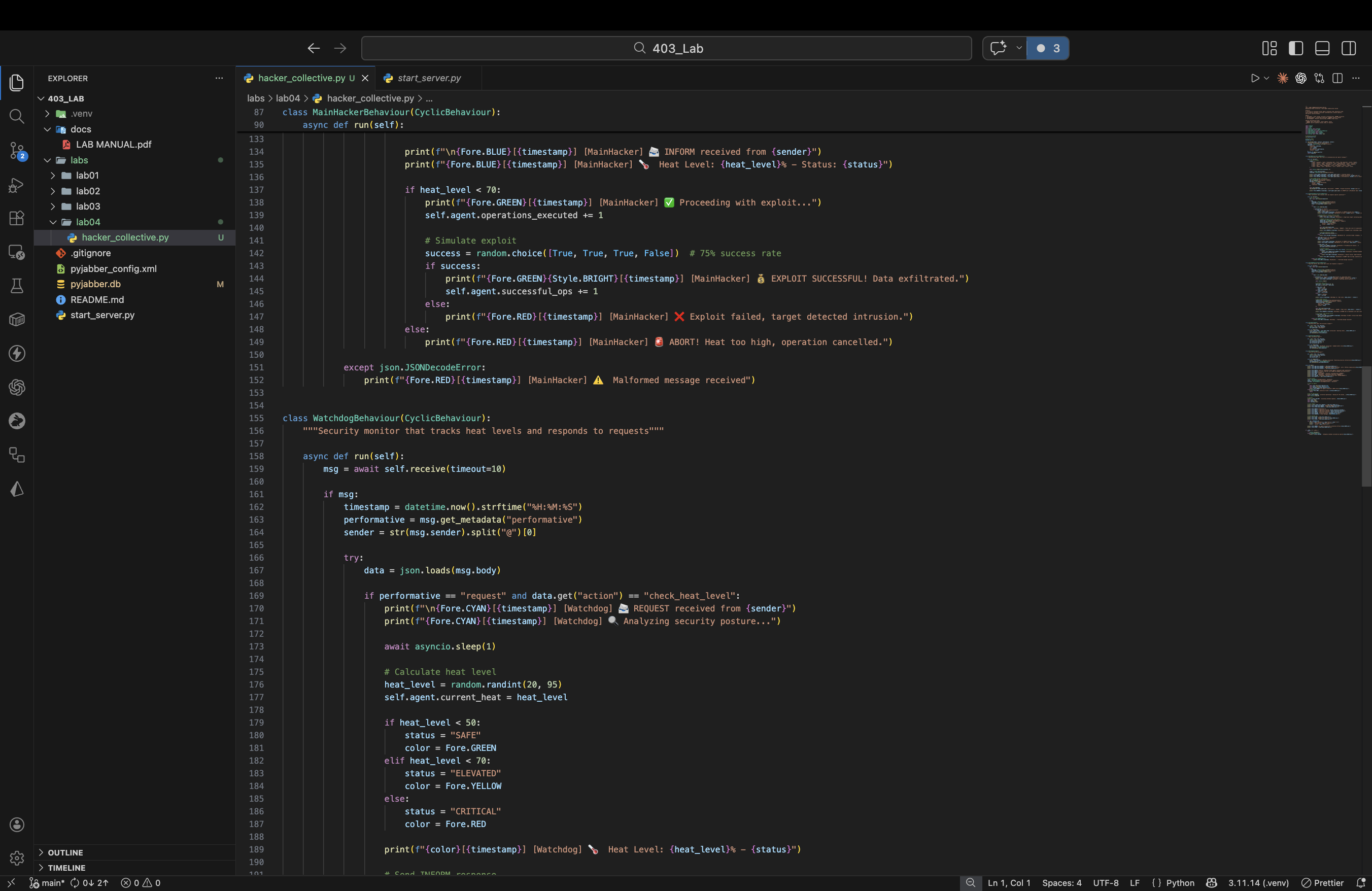
The following screenshots show the complete implementation of hacker\_collective.py with all three agent behaviours:



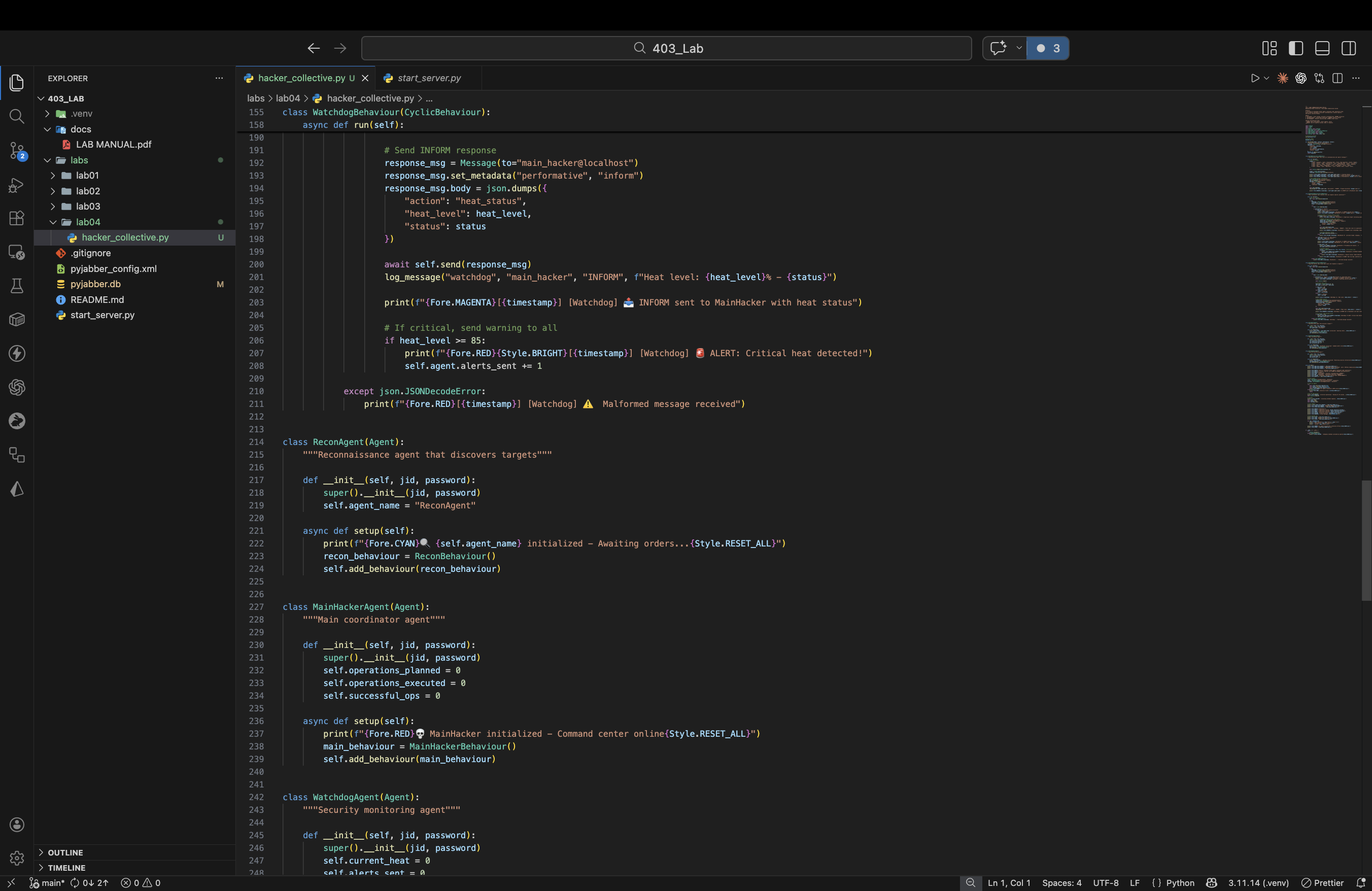
*Figure 1: Imports, message logger, and ReconBehaviour (scan + INFORM)*



*Figure 2: MainHackerBehaviour – receives INFORM, sends REQUEST*



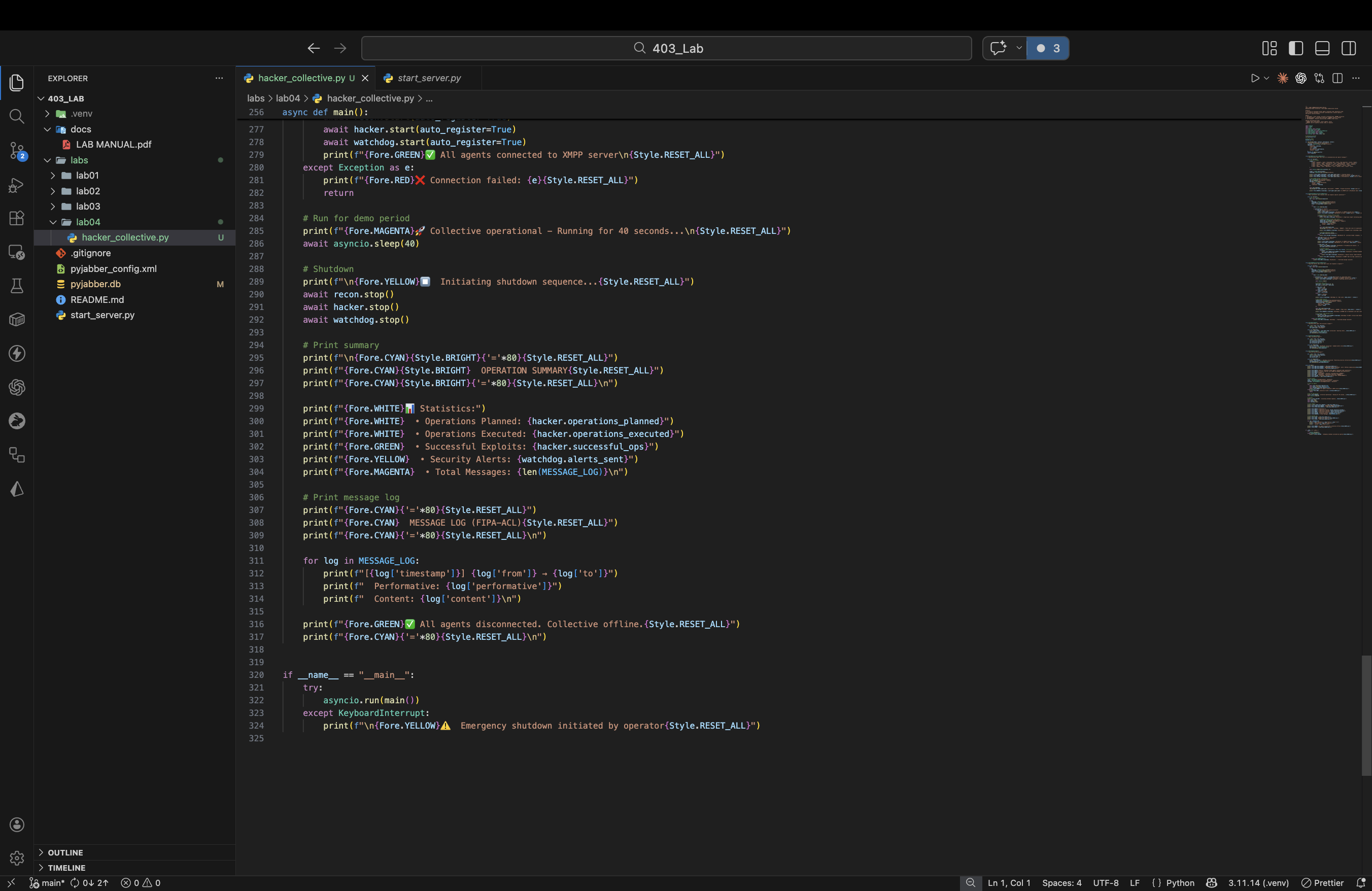
*Figure 3: MainHackerBehaviour – heat response handling and exploit logic*



*Figure 4: WatchdogBehaviour – REQUEST handling and INFORM response*



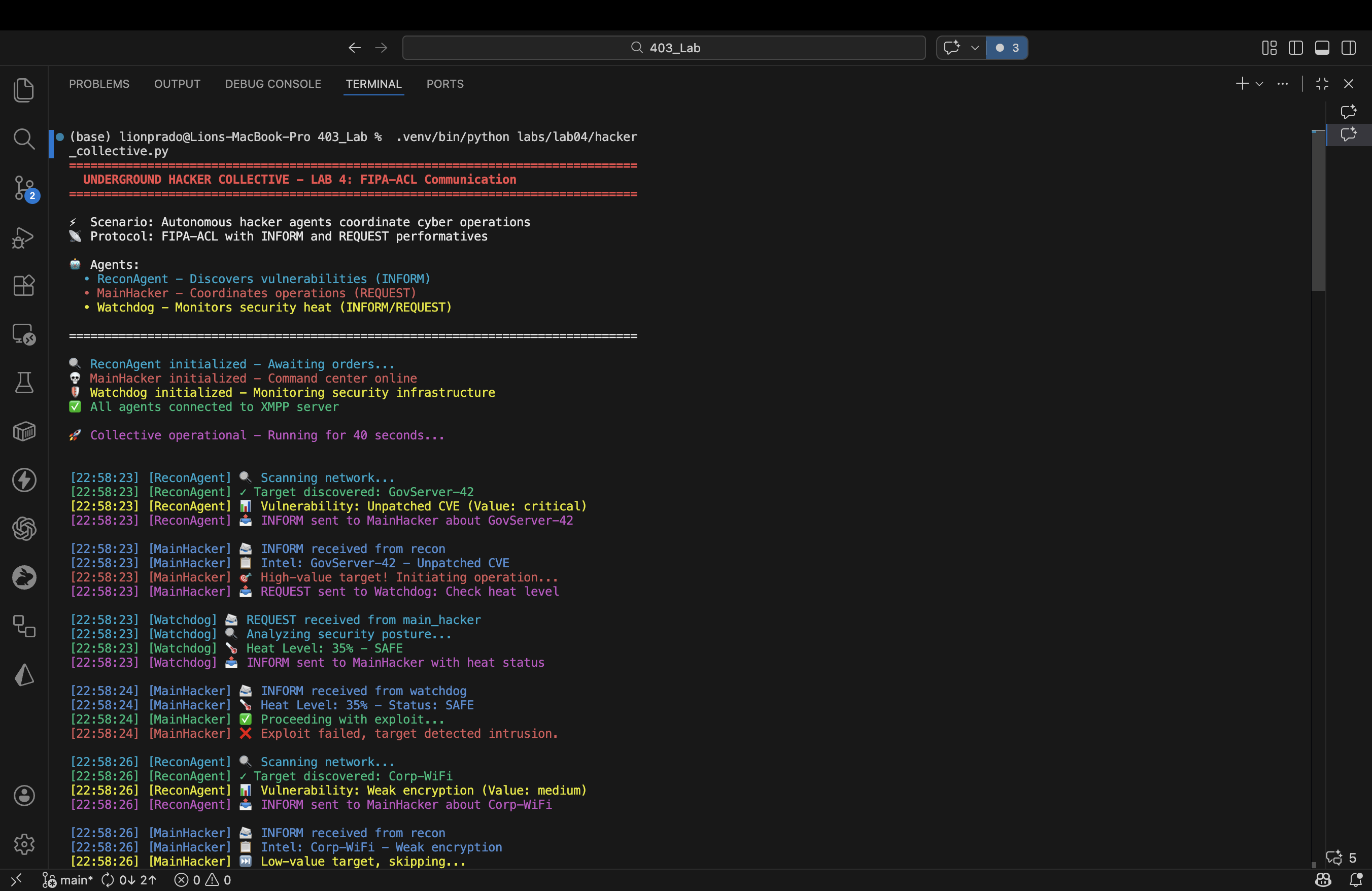
*Figure 5: Agent classes (ReconAgent, MainHackerAgent, WatchdogAgent) and main()*



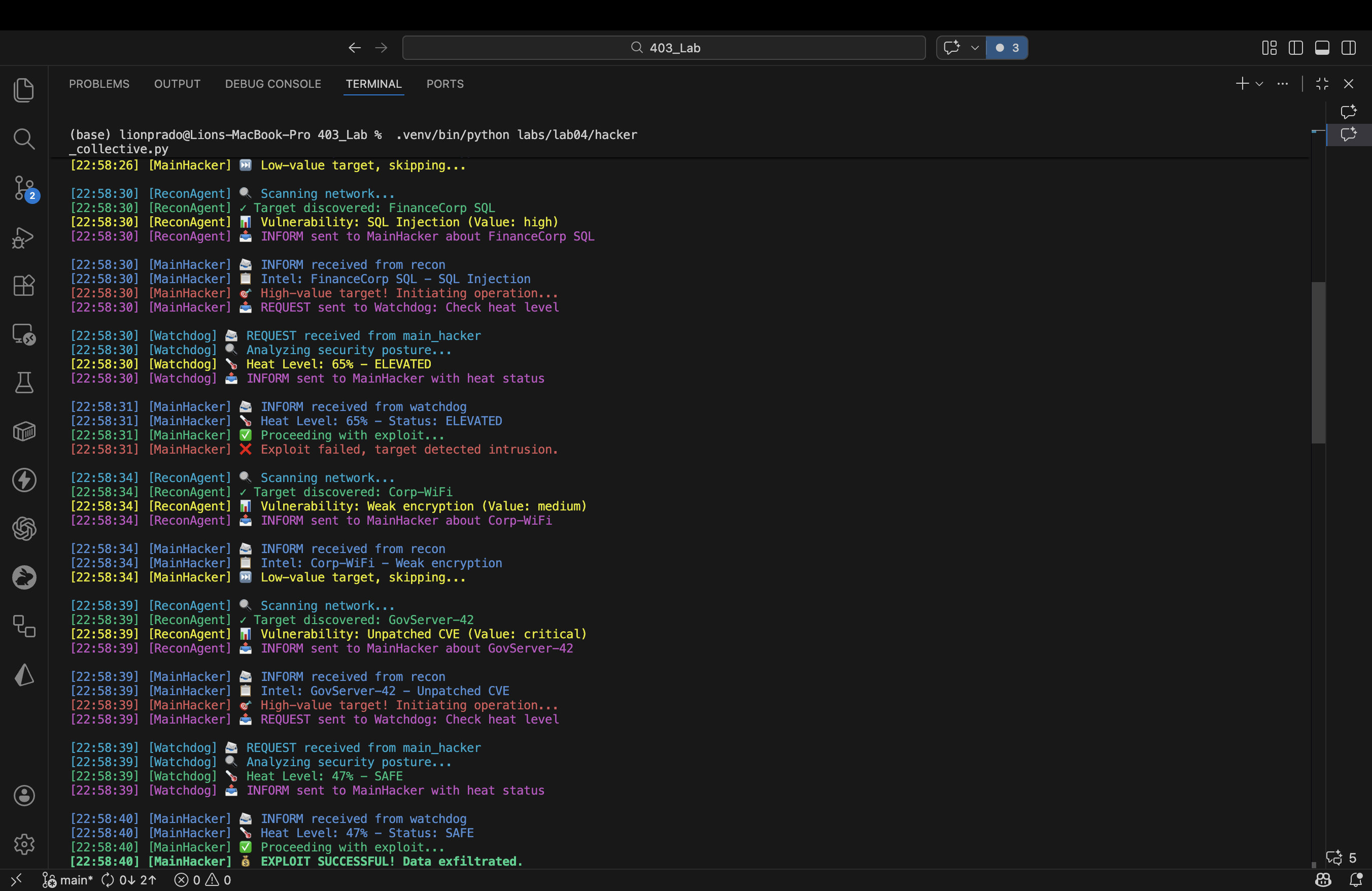
*Figure 6: main() – agent startup, demo run, summary, and message log output*

## Execution Output

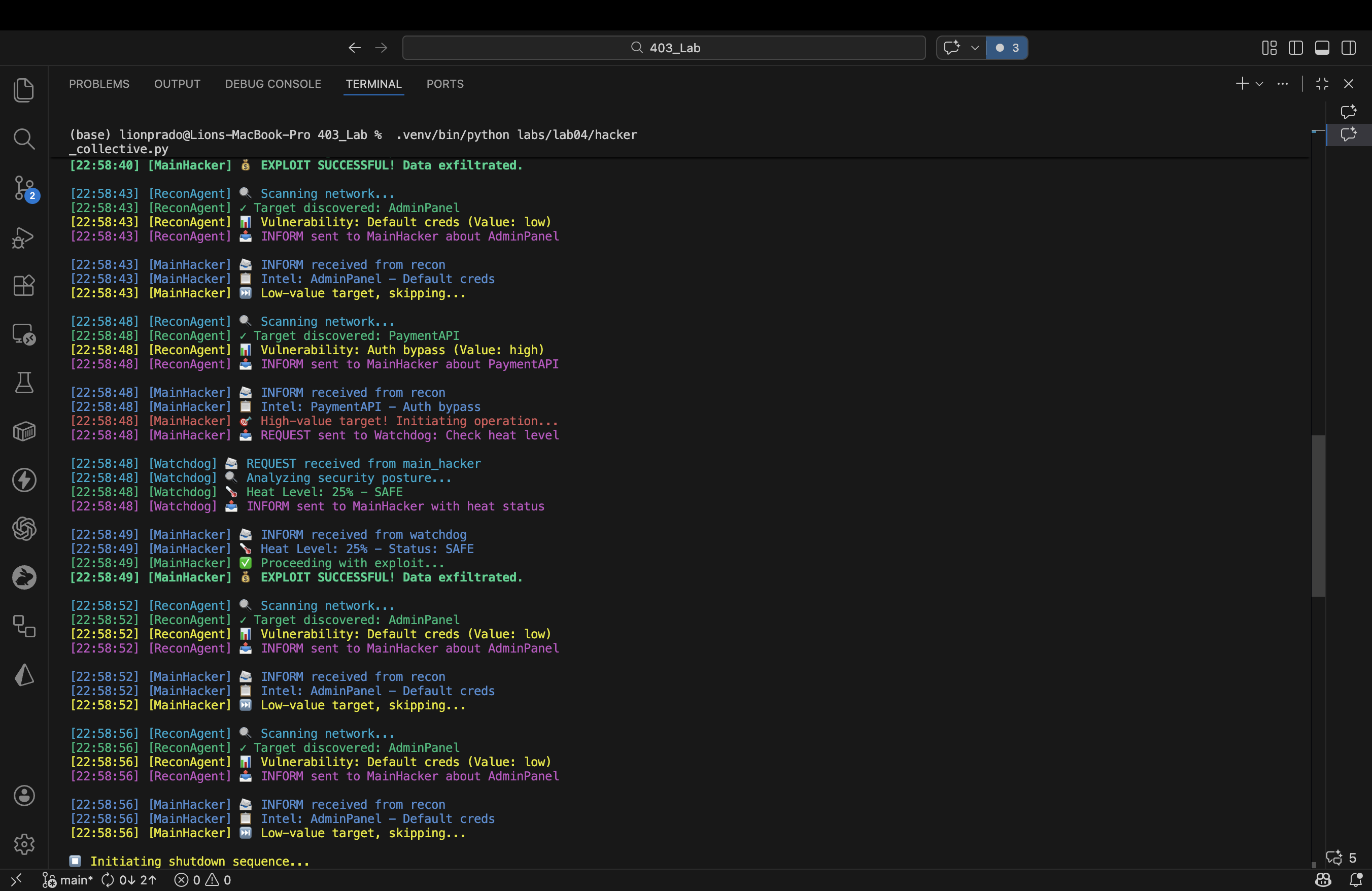
The following screenshots show the terminal output of a complete 40-second run demonstrating all FIPA-ACL message exchanges:



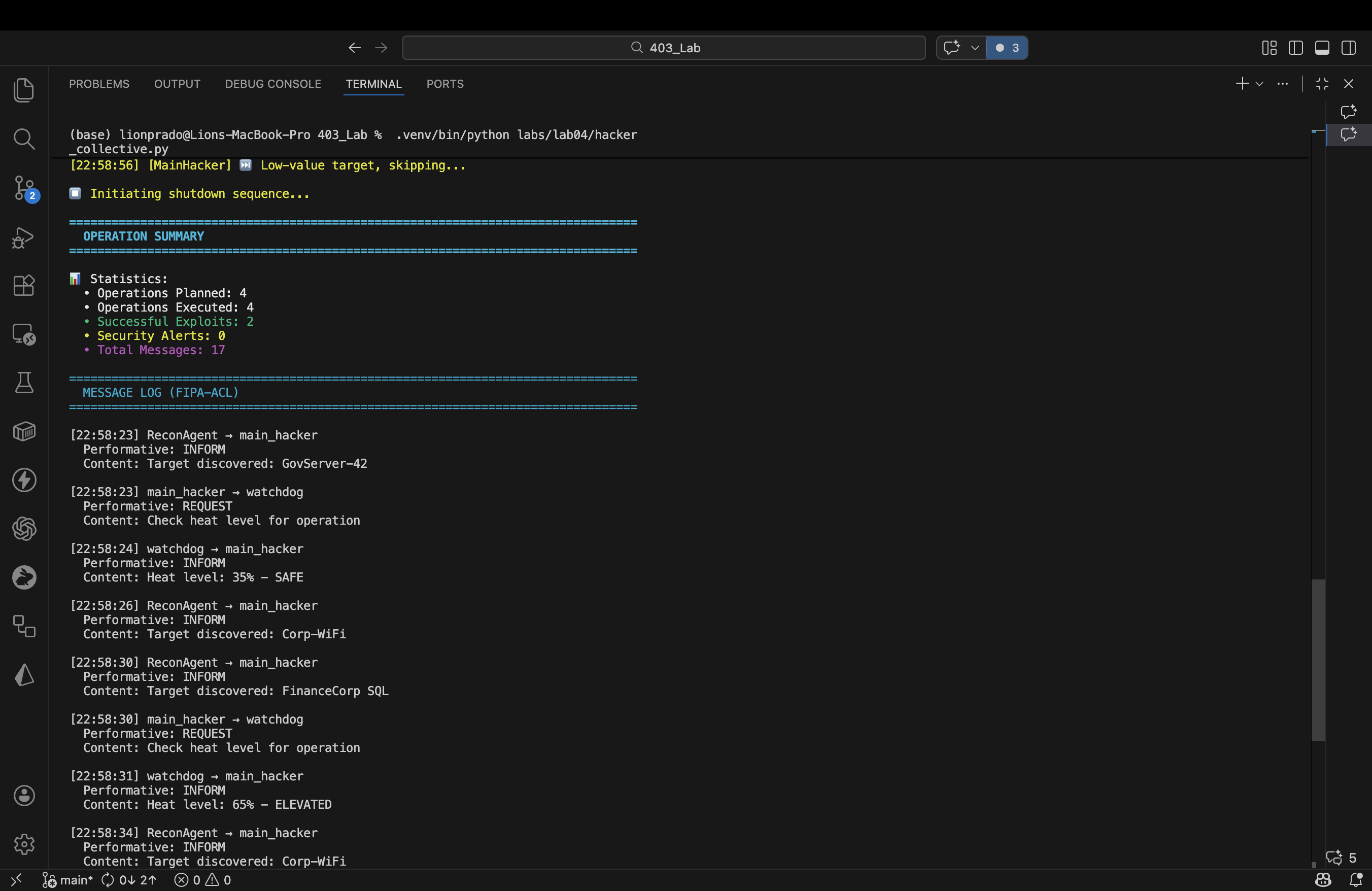
*Figure 7: Agent initialization and first INFORM/REQUEST exchanges*



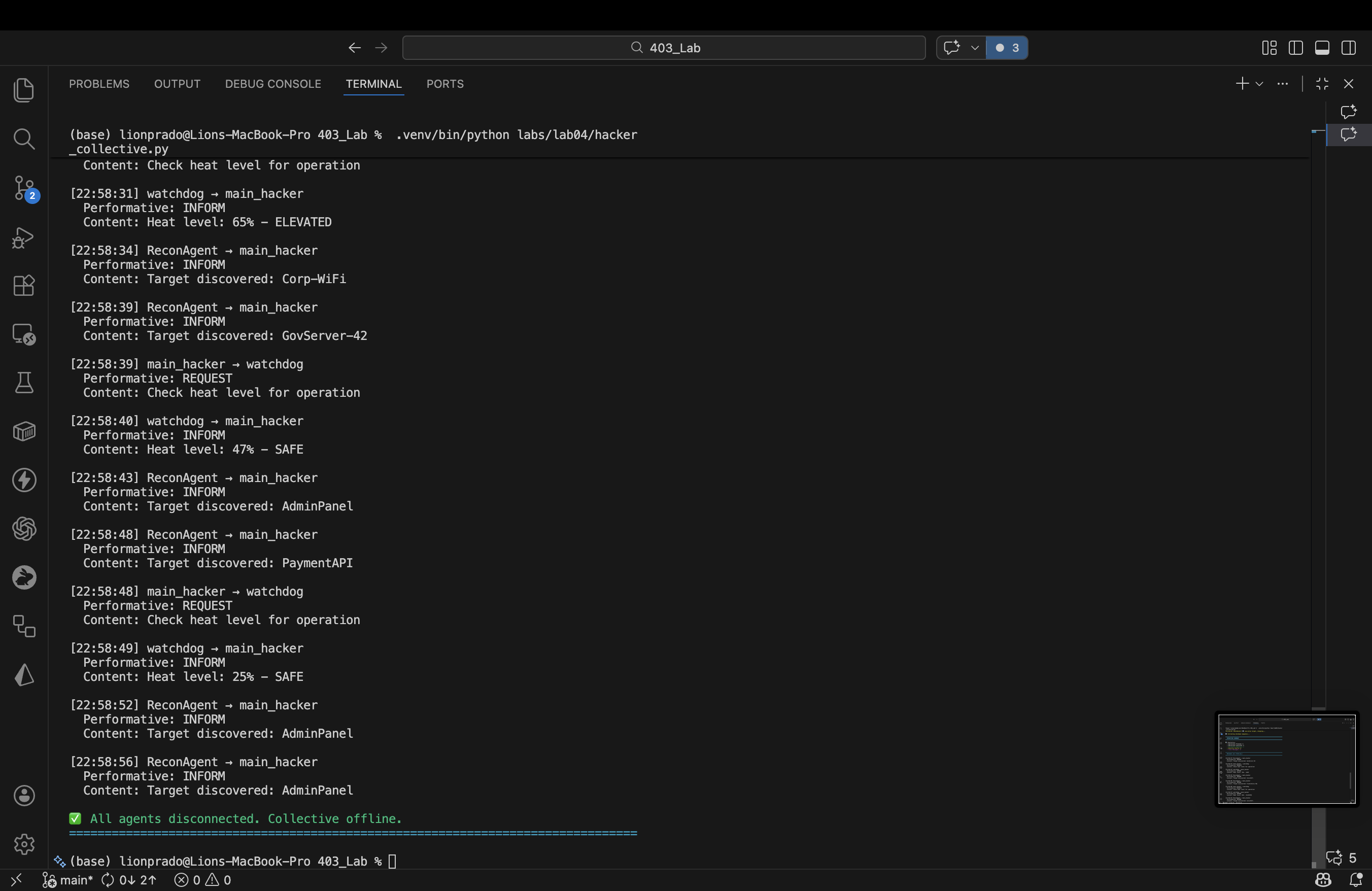
*Figure 8: Continued operations – exploits, heat checks, and target scanning*



*Figure 9: More scanning cycles and exploit results*



*Figure 10: Operation Summary and FIPA-ACL Message Log (part 1)*



*Figure 11: FIPA-ACL Message Log (part 2) – all messages with performatives*

## Message Log (Sample)

Below is a representative sample of the FIPA-ACL message log produced during execution:

|  |  |  |  |
| --- | --- | --- | --- |
| From | To | Performative | Content |
| ReconAgent | main\_hacker | INFORM | Target discovered: GovServer-42 |
| main\_hacker | watchdog | REQUEST | Check heat level for operation |
| watchdog | main\_hacker | INFORM | Heat level: 35% – SAFE |
| ReconAgent | main\_hacker | INFORM | Target discovered: FinanceCorp SQL |
| main\_hacker | watchdog | REQUEST | Check heat level for operation |
| watchdog | main\_hacker | INFORM | Heat level: 65% – ELEVATED |