"How to", Practical Approach To Multi Agent Cyber Defense

Author: Dylan Renier Pérez Narváez
Advisor: Prof. Lisabel Rodriguez Espinosa, J.D.
Master in Computer Science
Graduate Project EXPO, May 2026

Abstract

This project represents a guided "how to" approach to learning multi agent defense. It explains how to move from theory to practice, and a step by step on how to get there. It includes two hands-on examples, one built in n8n and one in Python, that put these ideas into practice with real tools. The goal is to help connect research and application in order to simplify the study of the field.

Introduction

Cybersecurity no longer deals with isolated incidents. Attacks adapt, coordinate, and move across networks. Static rules fail once an experienced attacker adjusts their approach. Over 60% of enterprise breaches involve lateral movement following a credential compromise [2]. Multi agent cyber defense responds through distribution, where multiple agents operate simultaneously, each focused on a specific task.

Background

Multi agent cyber defense distributes detection across tiers [3]. Bottom-tier agents clean raw data. Mid-tier detection agents monitor network traffic and behavioral patterns independently. Top-tier agents correlate findings across the system. Communication uses information, request, proposal, and alert messages [4]. This tiered architecture is the foundation of examples in this project.

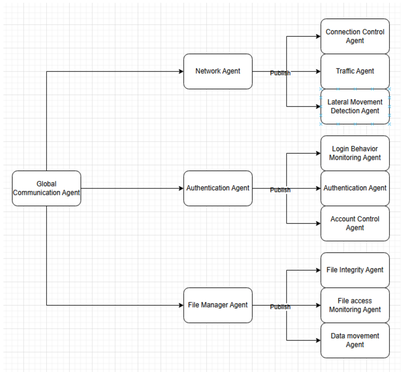

Figure 1: Cyber Defense Tiered Structure

Problem

Most research on multi agent systems assumes prior knowledge of agents and their implementation. Literature presents autonomouse agents, distributed decision making agents without really giving explanation on how to get there. This project organizes those ideas into a framework with working proof of concepts.

Methodology

Two implementations were built. The first is an n8n workflow with four Claude-powered agents (input Analysis, Network Analysis, Pattern Analysis, and the Coordinator) in a pipeline. This will show the communication pattern found in common multi agent cyber defense projects. The second is a Python script using Anthropic's tool use API where Claude decides which agents to call based on input content. This to show intelligent tool selection depending on the use case or prompt. Both were tested with four different scenarios: These were Network only prompt, pattern only prompt, combined threat prompt, and a no threat prompt.

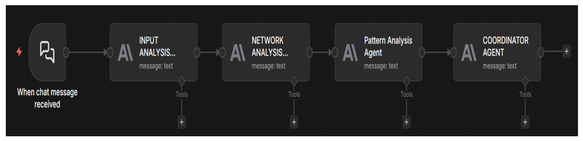

Figure 2: N8N Example Workflow

```
"name": "network_analysis",
"description": """Analyzes network-level indicators
only: IPs, ports, traffic patterns.
Does not assess behavior.""",
"input_schema": {
    "type": "object",
    "properties": {
        "raw_input": {"type": "string"}
    },
    "required": ["raw_input"]
}
```

Figure 3: Python Example Network Analysis Tool

```
SYSTEM_PROMPT = """You are the orchestrator of a
multi-agent cyber defense system.
Call network_analysis and pattern_analysis as
needed based on what the input contains.
Skip network_analysis if the input has no
network indicators. Skip pattern_analysis if
the input has no behavioral indicators.
After the analyses are done, call cordination_agent
to produce the final assessment."""
```

Figure 4: Python Example System Prompt

Results and Discussion

The n8n pipeline correctly identified threats across all four test cases with the Coordinator producing unified assessments. The Python agent demonstrated intelligent tool selection, skipping irrelevant agents based on input content. A SYN flood (DOS attack) triggered only the Network Agent. A macro attack (A behavioral attack) triggered only the Pattern Agent. Combined inputs triggered both. Normal Activity triggered neither. Both implementations validated the tiered agent architecture described in the theoretical framework.

Test	Severity	Network Threat	Pattern Threat	Agents Flagged
SYN Flood	5	5	0	Network Agent
Macro Attack	5	0	5	Pattern Agent
Both Attack	5	4 - 5	4 - 5	Both Agents
Normal Activity	0	0	0	Neither

Figure 5: N8N Workflow Agent Output Comparison

```
> User account "admin_backup" authenticated from IP 91.108.4.15 at 2
:47am, immediately ran net user /domain to enumerate all accounts, t
hen established an outbound connection to 185.220.101.45 on port 444
4

network_analysis
NETWORK THREAT LEVEL: 5

NETWORK DESCRIPTION: Multiple indicators of compromise detected in s
equence: authentication from a suspicious IP at an unusual hour (2:4
7am), immediate execution of domain enumeration command suggesting r
econnaissance activity, and establishment of outbound connection to
a known Tor exit node (185.220.101.45) on non-standard port 44444. Th
is pattern is consistent with active lateral movement and data exfil
tration.

CONFIDENCE: 5
PATTERN DESCRIPTION: The sequence demonstrates classic post-compromi
se lateral movement behavior: authentication from an unusual IP at a
n atypical hour, immediate execution of domain enumeration commands,
 and rapid establishment of an outbound connection to an external IP
 on a non-standard port. This chained activity pattern is consistent
 with credential compromise, reconnaissance, and command-and-control
 communication.

CONFIDENCE: 5

cordination_agent
OVERALL SEVERITY: Very High

THREAT SUMMARY: The "admin_backup" account has been compromised and
is actively being exploited for lateral movement and potential data
exfiltration. The attacker authenticated from a suspicious IP at an
unusual hour, immediately performed domain enumeration reconnaissanc
e, and established communication with a known Tor exit node, indicat
ing command-and-control activity. This represents an active, multi-s
tage attack with clear indicators of credential compromise and malic
ious intent.

AGENTS THAT FLAGGED THIS: Both
```

Figure 6: Python Output Both Agents Triggered

```
> Incoming traffic from IP 185.220.101.34 on ports 22, 80, 443, 8080

network_analysis
NETWORK THREAT LEVEL: 2

NETWORK DESCRIPTION: Traffic from IP 185.220.101.34 is originating f
rom a known Tor exit node, which is a common anonymization service u
sed for both legitimate privacy purposes and malicious activities. T
he multi-port probing pattern (22, 80, 443, 8080) suggests reconnais
sance scanning across common service ports. This combination indicat
es potential reconnaissance activity, though the threat level remain
s low without evidence of successful exploitation or malicious paylo
ad delivery.

CONFIDENCE: 4

cordination_agent
OVERALL SEVERITY: Low

THREAT SUMMARY: Traffic from a known Tor exit node probing multiple
common service ports (22, 80, 443, 8080) indicates reconnaissance sc
anning activity. While the multi-port pattern is consistent with act
ive enumeration, there is no evidence of successful exploitation, ma
licious payloads, or confirmed attack behavior. The activity warrant
s monitoring but does not currently pose an immediate threat.

AGENTS THAT FLAGGED THIS: Network Agent
```

Figure 7: Python Output No Agents Triggered

Conclusions

Multi agent cyber defense is not as abstract as the research makes it seem. With these examples you can go from reading research to building real projects. The n8n workflow demonstrated the communication aspect between agents while the Python agent demonstrated intelligent tool choosing apart from the communication aspect. The goal of connecting research and application was achieved in a way that lets new inexperience people in multi agent cyber defense get into the field.

Future Work

Add a reinforcement learning feedback loop so agents improve over time. Implement bidirectional communication between agents.

Acknowledgements

I would like to thank my advisor Lisabel Rodriguez Espinosa for her guidance throughout the project, she was a great help and steered my main idea for the project into what it is now. I also want to acknowledge the Polytechnic University of Puerto Rico for providing resources and trainings on how to complete the project.

References

[1] C. R. Landolt et al., "Multi-Agent RL in Cybersecurity," NATO STO-MP-IST-209, 2025.

[2] M. Imani et al., "Multi-Agent Cyber Defense with Zero-Trust Actions," IEEE MILCOM, 2025.

[3] G. G. Helmer et al., "Intelligent Agents for Intrusion Detection," Iowa State Univ., 1998.

[4] Z. Da, "Multi-Agent Communication and Collaborative Decision-Making," arXiv:2305.17141, 2023.

[5] U. R. Pol, "No-Code Intelligence: Building AI Agents with N8N," IJFMR, vol. 7, no. 6, 2025.

[6] Anthropic, "Anthropic Python SDK," 2025.

[7] H. Azmat, "Automating Business Workloads with n8n and AI Agents," 2025.

[8] Anthropic, "Claude Model Pricing," 2025.