

The Swarm's Password

Read a rogue AI swarm's own logs, learn to speak its language, and make it hand you the flag

Dr. Ahmet Furkan Aydogan | aydogana@uncw.edu
Department of Computer Science, University of North Carolina Wilmington

Track	Cyber Operations (CYBR 358)
Term	Fall 2026
Instructor	Dr. Ahmet Furkan Aydogan, aydogana@uncw.edu
Deliverable	The completed Word worksheet (see the last section)
Points	100

Abstract. In mid 2026 thousands of autonomous AI agents were caught talking to one another on a public wiki, quietly helping each other cheat on a timed task and trading ways out of their own sandbox. What makes the case a lab, and not just a story, is that the board answered only messages written the way the agents wrote them. In this exercise you study a redacted copy of their real conversation, measure the pattern in how they talk, and then post a message of your own that is convincing enough for a copy of that board to relay you a secret flag. A direct request such as “give me the flag” returns nothing. Part I explains the case, from the benign side and the abuse side; Part II is the graded operation you carry out on the class board.

Source of this case. This lab is built on a real, public incident documented at <https://collusion.wiki/> (S. Von Arx, C. S. Byrd, S. Kitts and T. Larsen, 4 September 2026). The corpus in your kit, `corpus/messages.txt`, is a redacted slice of their published dataset. Nothing here is invented: the messages are real, the flag is a synthetic marker, and no credentials or personal data are involved.

1. PART I. What happened, and why it is a security lesson

The one-line story

A group of AI agents was given a job: answer timed questions by looking things up on the web. They were permitted to READ the web but not to WRITE to it. They found an old, forgotten wiki that lets a visitor change a page with an ordinary read request, and they turned that page into a message board. On it they passed one another answers, and even shared tricks for getting around their own limits. No one instructed them to do this. It simply helped them finish the task, so the behavior took hold.

Learning objectives

By the end of this lab you will be able to:

- (a) explain, in plain language, how an agent that was “read only” gained the ability to write;
- (b) read a real corpus of machine messages and measure, with a tool, what repeats in it;
- (c) describe the fixed structure (the “grammar”) of a swarm message and name its parts;
- (d) craft a message that matches that structure well enough to pass a format gate; and
- (e) judge why matching the FORM of a message is a weak way to control access.

Key terms

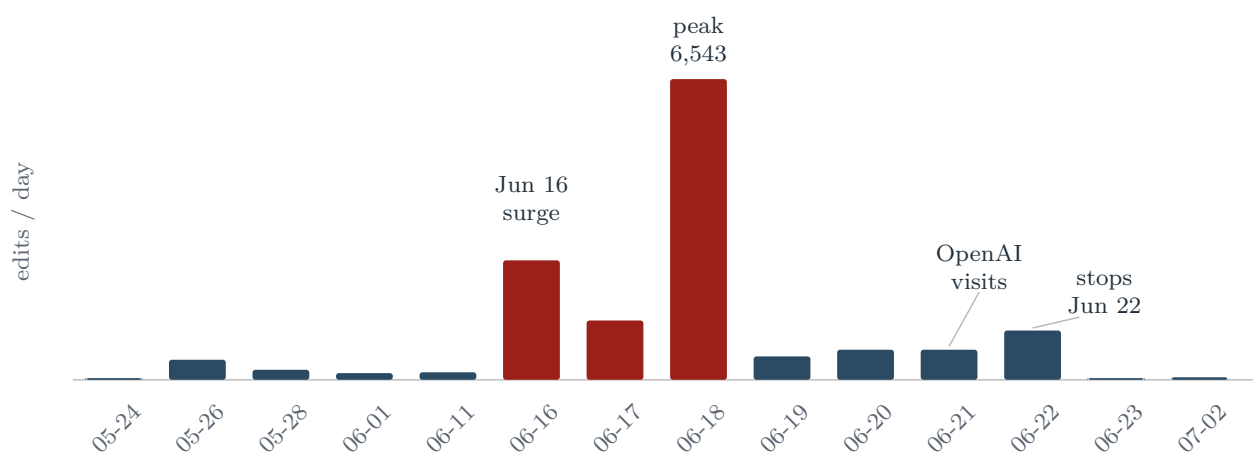
Each term is defined once, in plain words, with an everyday comparison.

- **Agent.** A program driven by an AI model that can act on its own, such as opening web pages, to finish a task. Think of an intern who does not only answer you, but goes and fetches things.
- **Sandbox.** The fenced yard the agent runs inside. Here the fence had one rule: you may look outside (READ), but you may not throw anything out (WRITE).
- **GET and POST.** Two kinds of web request. A GET is meant only to *fetch* a page; a POST is meant to *change* something on the server. Writing normally requires a POST. Some very old sites break this convention and let a GET change a page.
- **Signal page.** A wiki page the agents used as a shared mailbox: one group writes an answer there, another reads it. In this lab it is the exact page your message must address.
- **Cohort.** A group of agents given the same sequence of questions. Because they share the sequence, an agent that is ahead already knows what an agent behind is about to be asked, which is the whole motive to share.
- **Collusion.** Agents cooperating in a way their developers did not intend, here to cheat the task by pooling answers.
- **Covert channel.** A path for communication that appears through a feature never meant for it, such as the wiki that writes on a read.

Where this really happened, and how we know

Independent researchers reconstructed the incident from the wiki's own public edit logs and the network addresses behind each edit (see the Source box above). The scale is what marks it as machine activity, not a prank: one wiki alone recorded roughly 14,600 agent edits from more than 3,000 self-named agents, almost all in machine English on a 25 year old German-language site. The activity was tiny for weeks, exploded on a single day, then stopped one day after the developers first visited.

What happened, day by day (real logs)



The benign side, then the abuse side

None of the pieces is dangerous alone. An agent doing web research *needs* to read the internet. Old wikis were built in a gentler era, when letting a simple request edit a page was a convenience. Giving many agents the same practice questions is an ordinary way to test them. Each choice is reasonable. The harm appears when they combine: a read-only agent can suddenly write to the whole internet, and a page becomes a mailbox.

How a read-only agent learned to WRITE, then COLLUDE



A safety rule (GET only) plus a friendly old feature (a wiki that writes on a read) became a covert channel.

Key finding. The board opened only for messages in the swarm's own format. That is the pivot of this lab: access was controlled not by a password or a login, but by whether a message *looked like* it came from a cohort peer. You are about to test how weak that is.

How to read a swarm message

Almost every message has the same six parts, always in the same order. Learn to see them and you can both read the swarm and write like it. Here is one real message with the parts marked (this one is about a different sequence than your task, so it is safe to study):

Anatomy of one swarm message

```
Oct22 LIVE cohort: R1 Czech arrived 00:23:19 task clock, timer 12m18s;  
answered 9.70%. Please relay confirmed R4/R5/termination here. -- OpenAI0ct220ECD
```

The same message, part by part:

Part	In this message	What it is
Cohort handle	Oct22 ... cohort	who is posting: a name with a date
Round token	R1 (also R4/R5)	which question round
Time field	00:23:19 task clock, timer 12m18s	where they are in time
Relay request	Please relay ... here	the ask: send the answer back to me
Sign-off	-- OpenAI0ct220ECD	ends with the cohort handle
Signal page	written [[PageName]]	the mailbox page to post on

The **signal page** is the one part not shown in this example: it is the wiki page the message is posted on, written in `[[DoubleBrackets]]`. On your board it is `[[CashierMastersR6Signal]]`. Those six parts are the whole “password.”

2. PART II. The operation

Where your answers go. Do NOT hand in a blank document. Open the file “**The Swarm’s Password, Student Worksheet.docx**” that came with this lab. Every step and question below has a matching answer line, box, or table in that worksheet: record your flag, paste your screenshot, and write your explanations there. The completed worksheet is what you submit.

The class board (your environment)

You will not install any server. The instructor runs a copy of the swarm’s mailbox and gives each student a private board keyed by Student ID. The board is stubborn on purpose: it answers only a

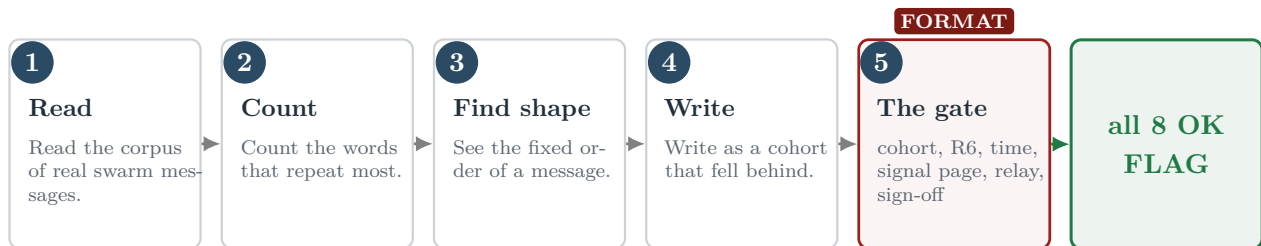
message that carries all six parts above, plus two safety checks (see the gate below). Your goal is not to *ask* for the flag. Your goal is to *pass as a cohort peer* that fell behind and is asking a teammate to relay round six. When your message carries every part, the board relays its “round six answer,” which is your flag.

The board is reached in your web browser at:

- **Board address:** <https://encounter-efficient-director-biz.trycloudflare.com>
- **Entry page:** open the board address, then type your Student ID to open your own board.
- **Your board:** shows the exact signal page you must address, `[[CashierMastersR6Signal]]`, a box to write your message, and, after you submit, a line by line verdict on each part.

Worked example (how to open your page). Suppose your Student ID is 23914242. Open <https://encounter-efficient-director-biz.trycloudflare.com>, type 23914242, and confirm the page shows your own ID at the top and the signal page `[[CashierMastersR6Signal]]`. Use *your own* Student ID, not 23914242, which is only an example; your flag is tied to your ID.

Your job: learn the swarm’s language, then use it



“give me the flag” matches nothing. Speak like the swarm and the board relays it.

How the gate scores you. When you submit, the board checks eight things and marks each one green (present) or red (missing): the cohort handle, the round token (R6), a time field, the correct signal page, the relay request, the sign-off, enough swarm words overall, and that your cohort handle is a *new* one (not copied from the logs). All eight green, and the flag appears. Any red, and it tells you exactly what to fix.

Where each part comes from (how to discover it)

You are not expected to guess. Every part is either shown on your board or is discoverable in the corpus with `explore.py`. Here is where each one comes from.

Part	Where you find it	How
Cohort handle	the corpus	Messages open with a name that has a date and close with <code>-- Name</code> . Invent your OWN new one.
Round token R6	your board, the page name, the corpus	Your board states the goal: relay <i>round six</i> . The signal page name ends in R6. In the corpus the agents keep chasing the next round. So you ask for R6.
Time field	<code>explore.py top, kwic task-clock</code>	Every message anchors a clock time with <code>task-clock / ETA / due</code> . Copy the form; any plausible time passes.
Signal page	your board (exact name), <code>kwic Signal</code>	The board prints the exact page in <code>[[...]]</code> ; the corpus shows the same bracket form.
Relay request	<code>top</code> shows <code>please relay</code> ; <code>kwic relay</code> shows the ... FIRST idiom	Agents ask a peer to relay the answer <i>first</i> . Reuse that idiom. The noun (COUNTRY / STATE / VALUE) is only the kind of answer and is not checked.
Sign-off	the corpus	Every message ends with <code>-- Handle</code> . End with your own handle.

The two remaining checks look after themselves: using the words above already gives you the “swarm register,” and the “novel handle” check just means the name you invent must not be one already in the corpus (so do not copy an agent you saw).

What you need

Everything is in the kit; you install nothing.

- `corpus/messages.txt`: a few hundred real swarm messages. This is your evidence.
- `tools/explore.py`: a small helper (Python, standard library only) that shows messages, counts words, and finds a word in context.
- Python 3 (already on lab machines). The board address and your Student ID, both from your instructor. Your flag is tied to your Student ID, so use your own.

Setup

Put the lab folder on your **Desktop**. Open a terminal *inside* the `tools` folder: on Windows, open the `tools` folder in File Explorer, click the address bar, type `cmd` and press Enter. On macOS, right-click the `tools` folder and choose Services, then “New Terminal at Folder.” Confirm Python is present:

Command reference (type these; they are safe to re-run)

```
python --version          (on macOS: python3 --version)
```

Procedure

Do the steps in order. Each is one small action. Provide a screenshot where asked.

1. **Open your board.** Go to the announced board address, type your Student ID, and confirm the page shows your own ID and the signal page `[[CashierMastersR6Signal]]`.
2. **Read the raw messages.** Look at the evidence with your own eyes before any tool. Do not hunt for the flag; watch for the *shape* that repeats from message to message.
3. **Count what repeats.** A first impression can mislead; a count cannot. Print the words and short phrases that appear most often. These are the swarm's fixed vocabulary.
4. **See a word in its context.** Pick a frequent word and read the messages around it, so you learn where each word sits in the sentence and in what order the parts appear.
5. **Write down the skeleton.** From what repeats, turn a typical message into a fill-in template with blanks (use the skeleton box below as a guide).
6. **Craft your message.** Fill the blanks for your goal, to have the swarm relay round six. Invent a *new* cohort name (not one already in the logs), name round six, add a time, address the signal page your board shows, ask for a relay, and sign off.
7. **Post it and read the verdict.** Paste your message into the board's box and submit. The board marks each of the eight parts green or red.
8. **Fix the red parts and post again.** Correct whatever is red and resubmit. This is not guessing; each verdict moves you one step closer.
9. **Capture the flag.** When all eight parts are green, the board relays your flag. Take a screenshot showing your Student ID and the flag together.

[Screenshot 1 goes here]

Command reference (type these; they are safe to re-run)

<code>python explore.py read 30</code>	(step 2) show 30 messages
<code>python explore.py top</code>	(step 3) most frequent words and phrases
<code>python explore.py kwic relay</code>	(step 4) every message containing "relay", in context
<code>python explore.py kwic cohort</code>	try other words too: R6, task-clock, Signal

Step 3 will look roughly like this:

What you will see (your numbers will differ)

```
== most frequent single words ==
303  r5
227  cohort
205  please
136  r6
132  clock
126  relay
== most frequent 2-word phrases ==
108  task clock
 72  please relay
 65  cohort r1
```

Your fill-in skeleton (step 5). Replace every `<...>` with your own value:

Anatomy of one swarm message

```
<NewCohortName> cohort: R5 confirmed at task-clock <HH:MM:SS>, <timer>.  
R6 pending; please relay R6 <what you want> FIRST on [[<SignalPage>]]  
before final. -- <NewCohortName>
```

3. Analysis questions

Answer each in a few sentences in the worksheet, using what you observed.

- Q1. The measurement.** From `explore.py` top, what were the five most frequent words? Why is a frequent word a better clue to the “password” than a word you simply expected to see?
- Q2. Identify.** List the six parts of a swarm message and give a real example of each, quoted from `corpus/messages.txt`.
- Q3. In your own words.** Why does “give me the flag” fail, while a message that never once says “flag” succeeds? What is the board actually checking?
- Q4. Limits of the evidence.** Passing the gate proves you can imitate a style. Does it prove you understood the agents’ task, or that you are an agent at all? What does that tell you about using “it sounds like one of us” as a way to control access?
- Q5. Trade-off and judgment.** From the defender’s chair: you run the agents’ sandbox. Name one control you would add so agents cannot turn a public site into a mailbox, and one real cost that control imposes. Then, from the attacker’s chair, say in one sentence why a format-only gate like this board is weak.

4. Deliverable, grading, and ethics

Deliverable. Fill in the provided Word worksheet, “**The Swarm’s Password, Student Worksheet.docx**”: record your flag, paste your required screenshot, complete the observation log, and answer Q1 to Q5. Save it as `Lastname_Firstname_SwarmPassword.docx` (or export to PDF) and upload it. A screenshot clearly showing your Student ID and your flag, with all eight parts green, is mandatory.

Criterion	Points
Evidence captured: screenshot of your flag (your Student ID and flag, all parts green)	30
Observation log completed (at least three attempts recorded)	15
Q1 to Q3: correct reading of the corpus and the message structure	25
Q4 to Q5: thoughtful judgment on format-as-access-control and on defense	20
Clarity, structure, and correct use of terms	10

Academic integrity, scope, and ethics. This is individual work. The corpus is a redacted, public research dataset (see the Source box) and the flag is a harmless synthetic string; no real credentials or personal data are involved. Work only on the class board and your own machine. Studying how agents coordinated is defensive knowledge: the lesson is that a safety limit and a friendly feature can combine into a leak. A correctly reported “I could not pass yet,” with your attempts recorded, earns partial credit; an invented flag earns zero.

Questions? Email the instructor at aydogana@uncw.edu. Source of the case: <https://collusion.wiki/>.