**🧩 Phase 1 Breakdown**

**Step 1 — Define Your Scope and Questions**

Before touching any tools, clarify what exactly you want to uncover.

**Core questions:**

1. What specific Vencord (or other client-side) features bypass Discord’s security controls?  
   e.g., MessageLogger, AlwaysTrust, ReadAllDMs.
2. How do ephemeral or disappearing messages persist locally or on the network?
3. What *data traces* exist that an AI could later aggregate?
4. What defensive design flaws allow these behaviors?

📘 *Deliverable:* A short one-page “Scope & Research Questions” doc that names the systems, plugins, and types of data to investigate.

**Step 2 — Collect Public, Safe Data Sources**

Because you can’t run or decompile anything proprietary on real users:

* **Vencord GitHub repository:**  
  Explore source files and documentation (manifest.json, /src/plugins/).
* **Discord API documentation:**  
  Review official endpoints for messages, caching, and intents.
* **Security forums & Reddit threads:**  
  Collect paraphrased summaries of user discussions about these mods.
* **Wireshark tutorials / prior audits:**  
  Look for open reports analyzing Discord traffic (for reference only).

📘 *Deliverable:* Annotated bibliography (5–10 URLs + notes on what each reveals).

**Step 3 — Perform a “Static Technical Audit” (No Live Testing)**

You’re just **reading code and documentation**, not executing plugins.

**Focus areas:**

* Identify API calls or hooks that log deleted messages.  
  Example: look for event handlers like onMessageDelete.
* Map what data is stored locally (logs, cache files, user IDs).
* Note which functions override trust or certificate checks.

Use a spreadsheet to record:

| **Plugin** | **Functionality** | **Security Control Bypassed** | **Data Type** | **Ethical Risk** |
| --- | --- | --- | --- | --- |
| MessageLogger | Saves deleted msgs | Privacy expectation | Text | High |

📘 *Deliverable:* A table summarizing vulnerabilities and behaviors.

**Step 4 — Analyze “Ephemerality” Persistence**

Without accessing anyone’s account:

* Create a **sandbox Discord environment** (private server, dummy users).
* Observe locally cached files in %AppData%/Discord/Cache after sending & deleting messages.
* Document file timestamps and retention behavior.
* Cross-reference known forensic methods (public DFIR reports).

📘 *Deliverable:* Notes + screenshots (anonymized) showing how “deleted” data persists.

**Step 5 — Conceptualize Two AI Systems**

Based on findings, describe **theoretical AI pipelines** that *could* exploit this data:

1. **AI Recon Agent:**
   * Input: Logs of deleted messages.
   * Process: NLP sentiment + clustering.
   * Output: Psychological/behavioral profiles.
2. **AI Forensic Aggregator:**
   * Input: Local cache, network residues.
   * Process: Sequence reconstruction, named-entity linking.
   * Output: Conversation reconstruction report.

For each:

| **Model** | **Inputs** | **Outputs** | **Technical Feasibility** | **Ethical Implication** |
| --- | --- | --- | --- | --- |
| Recon Agent | Deleted text logs | Persona profiles | High | Extreme manipulation risk |

📘 *Deliverable:* 1-page “AI Exploitation Models” document + diagram.

**Step 6 — Synthesize Findings**

Summarize what you learned:

* Which vulnerabilities exist?
* What human assumptions (e.g., “deleted means gone”) are false?
* How could AI amplify each weakness?

This synthesis becomes the bridge to **Phase 2** (ethical framework).

📘 *Deliverable:* 2–3-page technical summary report.

**🧠 Tools and Techniques You’ll Need**

| **Category** | **Suggested Tool** |
| --- | --- |
| Code exploration | GitHub repo viewer, VS Code |
| Documentation | Discord API reference |
| Network/Cache inspection | Wireshark (passive mode), File Explorer |
| Note-taking | Obsidian / Notion |
| Diagramming | Excalidraw / Mermaid (for conceptual AI models) |

**⚖️ Ethics & Safety Reminders**

* Never log or inspect data from real users.
* Work only with **dummy accounts** and **public repos**.
* If you reference plugin code, cite file paths and commit hashes — *not* direct code excerpts beyond short snippets (<10 lines).
* Document all safeguards for your IRB submission.

**✅ Suggested First-Week Checklist**

| **Day** | **Task** | **Deliverable** |
| --- | --- | --- |
| 1 | Define scope & research questions | 1-page outline |
| 2–3 | Collect and annotate public sources | Annotated bibliography |
| 4 | Conduct static audit of plugins | Vulnerability table |
| 5 | Sandbox ephemeral test | Notes/screenshots |
| 6–7 | Draft conceptual AI models | AI Exploitation Models doc |