**Project:** The Algorithmic Panopticon: How AI Amplifies Security Exploits in Online Communities  
**Phase:** 1 — Technical & Systems Analysis  
**Author:** Mohammad Wael  
**Date:** October 18, 2025

**1) Background**

Client-side mods (e.g., Vencord) and ephemeral messaging features create gaps between user expectations and technical reality. Phase 1 will produce a safe, reproducible technical inventory of those gaps and conceptual AI pipelines that could exploit them.

**2) Objective**

Produce a clear, ethically-compliant technical report that

(a) catalogs concrete behaviors in mod/plugin ecosystems and ephemeral messaging persistence,

(b) maps attack surface elements an AI could ingest, and

(c) produces two conceptual AI exploitation models for use in later ethical and case study analysis.

**3) Primary Research Questions**

1. Which client-side plugin features explicitly or implicitly bypass platform security/privacy controls (e.g., logging deleted messages, bypassing phishing or certificate warnings)?
2. What artifacts of "deleted" or "ephemeral" messaging persist on local systems or transit (cache, logs, backups, network residues)? How long and where?
3. What data formats, identifiers, or metadata exist that make automated ingestion and cross-user correlation feasible for an AI?
4. What constraints or obstacles prevent large-scale automated exploitation today (rate limits, platform encryption, plugin distribution channels)?
5. What are technically realistic AI pipelines (inputs → processing → outputs) that could scale these vulnerabilities into mass surveillance or social-engineering tools?

**4) Scope (what we will analyze)**

* **Platforms / Ecosystem:** Discord (primary), with references to comparable behaviors in other chat clients where relevant.
* **Plugins / Mods:** Public, open-source client-side mods (e.g., Vencord plugin directories such as MessageLogger, AlwaysTrust) — only static code review.
* **Data Types:** Message text, message IDs, timestamps, user IDs (anonymized), cached media metadata, local cache files, update manifests.
* **Methods:** Static code review, file-system observations in a controlled sandbox, documentation/API review, and literature/forensics references.
* **Deliverables (Phase 1):**
  + Annotated list of plugin features and the security controls they interact with.
  + Table of data artifacts and persistence behaviors (local & network).
  + Two conceptual AI model descriptions and architecture diagrams.
  + 2–3 page technical summary (ready for Phase 2 handoff + IRB notes).

**5) Out of Scope (explicit)**

* No dynamic execution of third-party plugins on real user accounts or community servers.
* No development of exploitative code or operational AI prototypes. (Models remain conceptual.)
* No collection or analysis of real users’ identifiable data.
* No public disclosure of private repository code or PII.

**6) Ethics & Safety Constraints (must follow)**

* Work only with public repo code, documentation, and controlled sandbox test accounts (dummy accounts you create).
* Anonymize any quoted forum content; paraphrase to avoid search re-identification.
* Keep all findings non-actionable: avoid step-by-step exploit guides; focus on threat modeling and mitigation.
* Record IRB pre-submission notes describing safeguards and data handling.