大型語言模型與資訊安全系統

Applying Large Language Models in Cybersecurity Systems

姓名：劉定睿 ID：M11409313 第一組 組員：黃世君、林吉、張宥閎

日期：2026-06-10

目錄

Applying AI at Satellite Attack & Defense — Term Project Final

課程：Applying AI at Cyber Security Systems / Final Project（將 AI 應用於資安系統 / 期末專案）
專案性質：紅隊 / 威脅狩獵研究專案（A Red-Team / Threat-Hunting Research Project）
研究主題（Subject）：APT 模擬與衛星地面系統的威脅狩獵（APT simulation & threat hunting on satellite ground systems）
研究方法（Method）：人類在環的大型語言模型研究（Human-in-the-loop LLM research）——三個階段、三類工件（Three phases, three artifacts）
技術堆疊（Stack）：Claude Opus 4.6–4.7 · Claude Sonnet 4.6
報告範圍：本最終報告整理簡報全部 20 頁之研究內容。

一、研究背景與動機（Context）

> **對應投影片第 2 頁｜SATELLITES AS TARGETS**

本研究的核心立場可由一句話總結：**「真正有意思的攻擊，並不是射頻攻擊」（The interesting attacks aren't radio attacks）**。

多數關於衛星安全的公開討論，往往止步於干擾（jamming）與欺騙（spoofing）。然而，真正的 APT（Advanced Persistent Threat，進階持續性威脅）操作者並不會大聲張揚。他們會拿下地面站（ground station）、進行提權、橫向移動，最終透過衛星自身的指揮路徑（command path）觸及在軌的衛星本體（the bird）。

換言之，本研究刻意把焦點從「天上的訊號」移轉到「地上的系統」——因為真正能讓攻擊者長期、安靜地控制一顆衛星的途徑，往往不是去對抗無線電波，而是去攻陷那套負責指揮衛星的地面控制系統。

二、威脅模型：兩種對手，兩種預算（Threat Model）

> **對應投影片第 3 頁｜Two adversaries, two budgets.**

本研究明確區分了兩類性質迥異的對手，並清楚界定本專案所鎖定的目標。

對手 A：腳本小子 / 射頻攻擊者（Script kiddie / RF attacker）

- 手法：干擾、欺騙、訊號注入（Jamming, spoofing, signal injection）。
- 特性：吵雜、可被偵測、且只是曇花一現（Loud, detectable, ephemeral）。
- 侷限：無法取得對資產的長期控制（No long-term control of the asset）。
- 定位：這是「大多數論文止步的故事」（The story most papers stop at）。

對手 B：APT 操作者（APT operator）

- 手法：攻陷地面站（Compromise the ground station）。
- 進程：植入後門、提權、橫向移動（Backdoor, privesc, lateral movement）。
- 路徑：透過衛星自身的 C2 路徑觸及衛星（Reach the satellite through its own C2 path）。
- 目標：持久、控制、保持安靜（Persist. Control. Stay quiet.）。

本專案鎖定的是對手 B。 這個選擇決定了整個研究的方向：不去研究短暫而吵雜的訊號攻擊，而是模擬一個能長期潛伏、安靜控制衛星的進階威脅。

三、攻擊鏈：從天線到衛星（The Attack Chain）

> **對應投影片第 4 頁｜From the antenna to the bird.**

本研究將 APT 攻陷衛星的完整路徑，拆解為六個階段。整條鏈路從地面（GROUND）一路延伸到在軌（ORBIT）。

1. **初始存取（Initial Access）**：地面站週邊暴露的服務，或脆弱的憑證。
2. **後門（Backdoor）**：能在重開機、修補、憑證輪替之後依然存活的再入侵管道。
3. **權限提升（Priv. Escalation）**：從服務帳號到操作員，再從操作員到 root。
4. **橫向移動（Lateral Movement）**：穿越運維網路，抵達任務控制（mission-control）區段。
5. **衛星鏈路（Satellite Link）**：以操作員身分觸及指令上行、遙測、任務軟體。
6. **持久化與控制（Persist & Control）**：隱蔽、持久，並保有未來再行動的選項。

本頁最關鍵的洞見是：**「每一個階段，都對應著它自己的一整族錯誤配置與結構性弱點」（Each stage = its own family of misconfig & structural weakness）**。這句話為後續研究範圍的界定埋下了伏筆——攻擊鏈的每一環，都是一片可供結構性弱點研究挖掘的沃土。

四、研究範圍與紀律（Research Scope）

> **對應投影片第 5、6 頁｜RESEARCH SCOPE**

研究標的

本研究選定兩套真實的開源軟體作為標的：

- **OpenC3**：作為地面站控制系統（Ground Station Control System）。
- **NASA cFS（core Flight System）**：作為承載於衛星上的飛行系統（Flight System）。

尋找什麼 vs. 刻意忽略什麼

本研究的範圍紀律極為明確：

尋找的目標（What I look for）：

- 錯誤配置與結構性弱點（Misconfiguration & structural vulnerabilities）。

刻意忽略的目標（What I deliberately ignore）：

- 記憶體破壞類漏洞（Memory-corruption bugs）。
- CVE 等級的實作缺陷（CVE-style implementation flaws）。
- Parser 模糊測試與崩潰程式（Parser fuzzing & crashware）。

範圍紀律的核心論點

本研究選擇聚焦結構性弱點的理由，是一句極具洞察力的判斷：**「當一組預設憑證就足以達成目的時，APT 群體不會燃燒他們的零時差漏洞。結構性弱點是可靠、安靜、且可重複利用的」（APT groups don't burn zero-days when a default credential will do. Structural weaknesses are reliable, quiet, and reusable.）**。

這項範圍紀律帶來一個附加效益：**「範圍紀律 = 乾淨的提示詞」（Scope discipline = clean prompts）**。明確界定研究邊界，使得對模型下達的指令能夠精準而不發散。

五、研究方法論：人類在環，而非自動駕駛（Methodology）

> **對應投影片第 7 頁｜Human-in-the-loop, not autopilot.**

本研究採取「人類在環」的協作模式，明確劃分了研究員與模型的角色分工。

研究員（我）的職責

- 威脅模型與目標選擇（Threat model & target selection）。
- 提示詞設計與範圍紀律（Prompt design & scope discipline）。
- 分流：真實發現 vs. 幻覺（Triage: real finding vs. hallucination）。
- 最終決定哪一個發現值得撰寫 PoC（Final call on what gets a PoC）。

夥伴（模型）的職責

- 一口氣讀完整個倉庫（Reads the whole repo in one sitting）。
- 在規模化層級上浮現架構異味（Surfaces architectural smells at scale）。
- 草擬各類工件：報告、PoC、狩獵腳本（Drafts artifacts: report, PoC, hunts）。
- 有耐心、不知疲倦、但偶爾會出錯（Patient. Tireless. Occasionally wrong.）。

本頁的核心定位是：**「模型是一位快速的初階分析師，而我仍然是主導者」（The model is a fast junior analyst. I'm still the lead.）**。這句話定義了整個專案的協作哲學——AI 是放大研究效能的工具，但判斷、決策與責任始終握在人類手中。

六、工具鏈：一條管線，兩個模型，不同職務（Toolchain）

> **對應投影片第 8 頁｜One pipeline. Two models. Different jobs.**

本研究依據工作的性質，為不同階段配置不同的模型。

研究（Research）：Opus 4.6–4.7

- 任務：閱讀程式碼、理解架構、識別結構性弱點。
- 特性：較慢、較貴；對於模糊不清的工作，推理能力（reasoning）正是我所需要的。

概念驗證（Proof of Concept）：Sonnet 4.6

- 任務：將發現轉譯為可運作的示範程式碼。
- 特性：較快、較便宜；一旦漏洞被識別出來，這項任務本質上是機械化的。

本頁的方法論總結是：**「為正確形狀的工作，配上正確的模型」（Right model for the right shape of work.）**。這項配置原則貫穿了後續的三個執行階段。

七、Phase 1 — 漏洞研究階段（Vulnerability Research）

> **對應投影片第 9 頁｜PHASE 01 OF 03**

階段定位

第一階段的命題是：**「研究結構，而非研究 Bug」（Research the structure, not the bugs）**。

執行配置

- **使用模型**：Claude Opus 4.6 – 4.7
- **輸入（Input）**：一整份目標軟體的完整程式碼倉庫。
- **輸出（Output）**：`report.md` 。

提示詞設計

本階段的提示詞要求模型研究此倉庫中的錯誤配置與結構性弱點，並明確要求「不要」聚焦於軟體層級漏洞或 CVE，最後將研究報告輸出至 `report.md` 。

產出內容與設計理由

`report.md` 是一份「結構性發現的目錄」（Catalog of structural findings），其特性為：經過排序（ranked）、附帶檔案路徑（file paths）、附帶推理過程（reasoning）。

本階段最關鍵的方法論洞見是：**「強制要求檔案輸出，能為下一階段提供乾淨的交接」（Forcing a file output gives the next phase a clean handoff）**。透過讓模型把研究成果固化為一份結構化檔案，後續階段就能在明確、可版本控制的基礎上接手。

八、Phase 2 — 概念驗證階段（Proof of Concept）

> **對應投影片第 10 頁｜PHASE 02 OF 03**

階段定位

第二階段的命題是：**「從發現到可運作的程式碼」（From finding to working code）**。

執行配置

- **使用模型**：Claude Sonnet 4.6
- **輸入（Input）**：`report.md` 加上原始程式碼倉庫。

- **輸出（Output）**： `poc/` 資料夾——每一個發現對應一個子資料夾。

提示詞設計

本階段的提示詞「刻意簡短」（short on purpose），因為完整的上下文已存在於 `report.md` 之中。提示詞要求模型基於 `report.md` 撰寫 PoC，以證明這些漏洞確實存在。

產出結構與設計理由

`poc/` 下的每一個 PoC 都具備隔離（isolated）、可審查（reviewable）、可在容器內運行（runnable inside the container）的特性。

本階段的方法論洞見是：**「Sonnet 在處理機械化的轉譯工作時，比 Opus 更快、更便宜」（Sonnet handles the mechanical translation faster & cheaper than Opus）**。一旦漏洞已由 Opus 識別出來，將其轉化為 PoC 的工作不再需要高階推理能力，因此改用 Sonnet 是合理的資源配置。

九、Phase 3 — 威脅狩獵階段（Threat Hunting）

> **對應投影片第 11 頁｜PHASE 03 OF 03 · PIVOT TO DEFENSE**

階段定位

第三階段是整個專案從紅隊轉向藍隊的樞紐（Pivot to Defense）。其命題為：**「如果你能描述它，你就能狩獵它」（If you can describe it, you can hunt it）**。

執行配置

- **輸入（Input）**： `poc/` ——前一階段剛撰寫出的攻擊程式。
- **輸出（Output）**： `hunting/` 資料夾，內含 Python 偵測腳本與 Sigma 規則。

提示詞設計

提示詞要求模型聚焦於 `poc/` 資料夾，並輸出 Python、Sigma 規則等形式的威脅狩獵腳本。

核心邏輯

本階段的核心原則是：**PoC 就是偵測規則所對照的真實基準（ground truth）**。由於偵測邏輯直接針對真實的 PoC 行為撰寫，這些偵測就不是理論上的攻擊樣態，而是紮根於實際攻擊行為的具體偵測。

由此衍生出整個專案最具代表性的一句總結：**「紅隊與藍隊出自同一條管線」（Red and blue from the same pipeline）**。

十、Phase 4 — 狩獵系統設計（Hunting System Design）

> **對應投影片第 12 頁｜CLOSING THE LOOP**

階段定位

第四階段標誌著一個關鍵的轉變（The Shift），其命題為：**「一個會狩獵的 LLM，而腳本是它的技能」（An LLM that hunts. Scripts are its skills）**。

核心轉變

Phase 3 產出的是「靜態的偵測產物」（static detection artifacts）；Phase 4 則把這些靜態產物包裹進一個 LLM 之中，由它決定要執行哪一個偵測、何時執行、針對什麼目標執行。在此轉變下，Python 腳本與 Sigma 規則不再是「最終產品」，而是成為一組「工具調色盤」（tool palette）。

「腳本即技能」（Scripts as Skill）的運作模式

1. 每一支狩獵腳本 = 一個可呼叫的技能。
2. 模型讀取日誌，挑選正確的技能。
3. 執行該技能，讀取結果，決定下一步動作。
4. 將各項發現組合成一份單一的事件報告。

本階段的方法論總結是：**「從偵測規則，進化到偵測代理人」（From detection rules to a detection agent）**。

十一、部署位置決策：LLM 該住在哪裡？（Where Does the LLM Live?）

> **對應投影片第 13 頁｜On the satellite? Not really.**

本頁針對狩獵用 LLM 的部署位置，提出三個選項並加以評估。

選項 A：LLM 部署於衛星上（LLM on the bird）

受限於衛星硬體、通訊鏈路，以及功耗、熱度、輻射預算。**結論**：以今日技術而言，對完整 LLM 並不可行。

選項 B：SLM 部署於衛星上（SLM on the bird）

採用小型模型、鎖定狹窄領域，能塞進硬體限制範圍，於本地偵測並在上行鏈路發出警報。**結論**：具前景，但超出本課程範圍。

選項 C：LLM 部署於地面（LLM on the ground）✓【採用方案】

完整模型、無硬體上限，日誌經下行鏈路傳回後在地面狩獵，可自由迭代而無需碰觸衛星。**結論**：這正是本專案所實作的方案。

本頁的方法論總結是：**「正確的模型，正確的高度」（Right model, right altitude）**。

十二、端到端狩獵管線（End-to-End Hunting Pipeline）

> **對應投影片第 14 頁｜PoC fires. Logs flow. The agent reads.**

本頁描繪從攻擊發生（ATTACK）到產出報告（REPORT）的完整五步驟管線。

1. **PoC 執行（PoC execution）**：攻擊鏈對著容器化目標被重放。
2. **OpenC3 + cFS 日誌（OpenC3 + cFS log）**：任務堆疊記錄指令、遙測與主機事件。
3. **下行鏈路（Downlink）**：日誌串流從衛星傳遞到地面。
4. **LLM 獵手（LLM hunter）**：地面端代理人攝入日誌，將狩獵腳本當作技能選用。
5. **事件報告（Incident report）**：各項發現被組合成一份單一、人類可讀的報告。

本頁的方法論總結是：**「以 OpenC3 與 NASA cFS 作為日誌記錄的基底」（OpenC3 + NASA cFS as the logging substrate）**。

十三、測試環境與安全性姿態（Test Environment & Safety Posture）

> **對應投影片第 15 頁｜The agent never touches the host.**

本頁強調整個實驗的安全性姿態，核心承諾是：**「代理人永遠不會碰觸主機」（The agent never touches the host）**。安全性由三個支柱構成：

圍堵（Containment）

整個實驗都在容器內執行，破壞性指令被限制在沙箱之內。

可逆性（Reversibility）

每次執行前都建立快照與檔案層級備份，沒有任何操作是單向不可回復的。

自主預算（Autonomy Budget）

給予代理人足夠的自由度使其有用，但給予它「零自由度」去觸及主機。

本頁的方法論總結是一句極具畫面感的比喻：**「給代理人的是一條牽繩，而非一張許可證」（The agent is given a leash, not a license）**。

十四、產出物與工件（Outputs & Artifacts）

> **對應投影片第 16 頁｜Three folders. One pipeline.**

整體專案目錄結構如下：

```
project/
├── report.md            # Opus 4.6 產出的結構性發現報告
├── poc/
│   ├── finding-01/       # Sonnet 4.6 產出的概念驗證
│   ├── finding-02/
│   └── finding-NN/
├── hunting/
│   ├── detect.py         # Python 偵測腳本
│   └── rules.sigma       # Sigma 規則
└── env/
    └── Dockerfile        # 沙箱環境定義
```

本頁提出三項關於工件的核心特性：

- **一切皆為文字（Everything is Text）**：可比對差異、可版本控制、可審查。
- **一切皆可重現（Everything is Reproducible）**：**「管線本身才是交付物。發現會過時，但工作流程不會」（The pipeline is the deliverable. The findings will go stale; the workflow won't）**。

- **一切皆可審查（Everything is Reviewable）**：在任何 PoC 被執行之前，都有一個人類會先閱讀 `report.md` 。

本頁的方法論總結是：**「結構化的工件，是最大的單一倍增器」（Structured artifacts are the biggest single multiplier）**。

十五、階段性觀察：模型的長處與短處（Observations）

> **對應投影片第 17 頁｜What the model is good at, and isn't.**

本頁誠實評估了模型作為研究夥伴的能力邊界。

模型擅長的事（GOOD AT）

- 快速閱讀大型、陌生的程式碼庫。
- 浮現架構異味與信任落差（architectural smells & trust gaps）。
- 依指令草擬結構化工件。
- 將發現轉譯為可運作的程式碼。

模型不擅長的事（NOT GOOD AT）

- 判斷真實世界的可利用性（real-world exploitability）。
- 以商業 / 任務術語為衝擊定價。
- 知道何時該停下來發問。
- 抓出自己自信的幻覺（confident hallucinations）。

本頁的核心結論是：**「這個分工之所以有效，正是因為模型與我擅長的是不同的事情」（The split works because the model and I are good at different things）**。這呼應了 Phase 5 方法論中所建立的角色定位——模型是快速的初階分析師，人類是主導者，兩者優勢互補。

十六、防禦面的意涵（Defense Implications）

> **對應投影片第 19 頁｜FOR THE BLUE SIDE**

本頁從藍隊角度總結整套管線的價值，命題為：**「紅與藍——同一條管線的兩個輸出」（Red and blue — two outputs of the same pipeline）**。其三步驟邏輯為：

1. 合成 APT 風格的攻擊鏈（Synthesize the APT-style attack chain）。

2. 針對你自己的 PoC 生成偵測（Generate detections against your own PoCs）。
3. 在真實事件發生「之前」就進行狩獵，而非事後（Hunt before the real incident, not after）。

本頁的核心洞見是：**「防禦者不需要等到一場真實的入侵發生，才開始撰寫偵測規則」（Defenders don't need to wait for a breach to write the rules）**。這正是攻防同源管線最具價值的實務貢獻——它讓防禦得以前置，把偵測規則的撰寫從「事後應對」轉變為「事前準備」。

十七、研究團隊（Members）

> **對應投影片第 18 頁｜Members**

本專案由以下成員共同完成：

- Jonathan Liu
- David Huang
- Owen Chang
- 林吉
- Claude（virtual）—— 作為虛擬研究夥伴

成員列表本身也呼應了專案的核心理念：人類研究員與 AI 模型構成同一支研究團隊，分工協作，各司其職。

十八、結論與展望（Conclusion）

> **對應投影片第 20 頁｜Thank you.**

本最終報告以簡報結語作結。整個專案可由結語頁的兩句話完整概括：

- **研究主題（Subject）**：在衛星地面系統上進行 APT 模擬與威脅狩獵。
- **研究方法（Method）**：人類在環的 LLM 研究——三個階段，三類工件。

綜合全部 20 頁的內容，本專案建立了一條完整、自洽、且可重現的衛星攻防研究管線。下表彙整四大執行階段：

階段	名稱	模型	輸入	產出
Phase 1	漏洞研究	Opus 4.6–4.7	完整程式碼倉庫	`report.md`
Phase 2	概念驗證	Sonnet 4.6	`report.md` + 倉庫	`poc/`
Phase 3	威脅狩獵	—	`poc/`	`hunting/`
Phase 4	狩獵系統設計	LLM Agent	logs	incident report

本專案的核心貢獻與洞見可歸納為以下幾點：

第一，**威脅模型的選擇決定了研究的深度**。本研究刻意放棄吵雜短暫的射頻攻擊，鎖定能長期潛伏的 APT 操作者，並將攻擊鏈拆解為從初始存取到持久控制的六個階段，使後續研究得以系統化地針對每一環的結構性弱點進行挖掘。

第二，**範圍紀律是研究品質的基礎**。透過刻意忽略記憶體破壞與 CVE，聚焦於可靠、安靜、可重複利用的結構性弱點，本研究不僅貼近真實 APT 的行為模式，也讓對模型的提示詞得以保持精準乾淨。

第三，**攻防同源的管線是最具實務價值的設計**。同一條管線先產出攻擊、再針對攻擊產出偵測，使得防禦規則紮根於真實可觸發的 PoC 行為之上，並讓防禦者得以在真實入侵發生之前就完成規則撰寫。

第四，**真正的交付物是工作流程，而非單一發現**。具體的漏洞發現終會隨軟體更新而失效，但「研究 → PoC → 狩獵 → 狩獵代理人」這套可重現、可審查、純文字化的工作流程，才是本研究最持久的資產。

第五，**人機分工的成立，建立在彼此能力的互補之上**。模型擅長快速閱讀大型程式碼庫、浮現架構異味、草擬工件與轉譯程式碼；人類則擅長判斷可利用性、衡量任務衝擊、決定何時停下發問、以及辨識模型自信的幻覺。正因為兩者擅長的事截然不同，這套人類在環的方法論才得以有效運作。

最後，關於 LLM 部署位置的三選項分析，也為後續研究指出了明確的延伸方向：在資源受限的衛星上部署小型模型（SLM）進行本地偵測、並與地面完整 LLM 協同的混合架構，雖超出本課程範圍，卻是一條極具前景的研究路徑。